ИТОГОВЫЙ КОНТРОЛЬ

ЕГЭ

2012

Информатика и ИКТ

Контрольные тренировочные материалы с ответами и комментариями

Экзамен с «Просвещением»

ИТОГОВЫЙ КОНТРОЛЬ: ЕГЭ

Информатика и ИКТ

ЕДИНЫЙ ГОСУДАРСТВЕННЫЙ ЭКЗАМЕН

2012

Контрольные тренировочные материалы с ответами и комментариями

Москва
Санкт-Петербург
«ПРОСВЕЩЕНИЕ»
2012

УДК 002(03)
ББК 32.81я2
И 74

Проект «Итоговый контроль»
Серия «Итоговый контроль: ЕГЭ» основана в 2010 году

Руководитель проекта *М. А. Поляков*
Научный руководитель проекта к. п. н. *Г. С. Ковалёва*

Авторы: *С. М. Авдошин, Р. З. Ахметсафина, О. В. Максименкова*

Информатика и ИКТ: ЕГЭ 2012: Контрольные тренировочные материалы с ответами и комментариями (Серия «Итоговый контроль: ЕГЭ») / С. М. Авдошин, Р. З. Ахметсафина, О. В. Максименкова.— М.; СПб.: Просвещение, 2012.— 338 с.: ил.

И 74

ISBN 978-5-09-025605-6.

Данное пособие предназначено для оценки учащимися степени готовности к ЕГЭ по информатике и ИКТ, а также для выявления ими пробелов в своих знаниях. Оно поможет познакомиться с требованиями, которые предъявляются на ЕГЭ к выполнению заданий разного типа.

Содержит варианты заданий, составленных на основе спецификации и демонстрационной версии КИМов, разработанных Федеральным институтом педагогических измерений (ФИПИ).

Сопровождается краткими комментариями к ответам и ссылками на соответствующий раздел пособия «Информатика и ИКТ: ЕГЭ: Учебно-справочные материалы».

Данное пособие может использоваться как для самостоятельной подготовки к ЕГЭ, так и для работы в классе.

УДК 002(03)
ББК 32.81я2

© С. М. Авдошин, Р. З. Ахметсафина, О. В. Максименкова, 2012
© Издательство «Просвещение», 2012
© Художественное оформление. Издательство «Просвещение», 2012
Все права защищены

ISBN 978-5-09-025605-6

Предисловие

Уважаемые читатели! Вы держите в руках второй сборник контрольных тренировочных материалов для подготовки к Единому государственному экзамену (ЕГЭ) по информатике и ИКТ, первый вышел годом ранее. Сборник предназначен как для учеников 10–11-х классов средних школ, так и для их учителей.

Настоящее пособие призвано помочь систематизировать изученный материал, а также выявить и устранить пробелы в знаниях.

В настоящее время в России изданы учебники разных авторов по информатике и ИКТ для старших классов средних общеобразовательных и специализированных школ. Последовательность изложения материала и его объём в разных учебниках зачастую существенно различаются. Соответственно различаются и учебные программы по информатике и ИКТ в разных школах. В одних учебниках делается упор на моделирование, в других — на программирование, в третьих – на информационные технологии и т.п. Поэтому научиться решать задания ЕГЭ, имея только один учебник по информатике и ИКТ, очень сложно.

В сборнике приводятся подробные решения одного примерного варианта ЕГЭ по информатике и пять вариантов для самостоятельного решения с ответами и комментариями.

Все задания снабжены ссылками на теоретический материал и примеры в учебно-справочных материалах для подготовки к ЕГЭ[1].

Для подготовки к экзамену и решения некоторых заданий рекомендуем использовать свободно распространяемые системы программирования, разработанные Научно-исследовательским институтом системных исследований Российской академии наук (НИИСИ РАН):

- КуМир (http://www.niisi.ru/kumir-ege.htm) – среда программирования на школьном алгоритмическом языке (школьном АЯ);
- FreePascal (http://www.niisi.ru/fp.htm) – среда программирования на языке Паскаль.

---

[1] Информатика и ИКТ: ЕГЭ: Учебно-справочные материалы для 11 класса (Серия «Итоговый контроль: ЕГЭ») / С. М. Авдошин, Р. З. Ахметсафина, О. В. Максименкова. – М.; СПб.: Просвещение, 2012.

Авторы пособия работают в Национальном исследовательском университете «Высшая школа экономики» на отделении программной инженерии. Все авторы имеют опыт разработки олимпиадных и тестовых заданий по информатике для выпускников школ и студентов. В сборнике предложены оригинальные решения некоторых заданий ЕГЭ.

Мы благодарим за помощь в составлении и проверке контрольных тренировочных материалов наших студентов: Лидию Антонову, Людмилу Галкину, Елену Кальсину, Елену Козлову, Дениса Кильдишева, Александру Малахову, Ксению Сергееву, Даниила Ширгазина.

Авторы выражают глубокую признательность Галине Васильевне Этиной, заместителю директора по информационным технологиям лицея № 126 Санкт-Петербурга, за большую помощь в работе над сборником, доброжелательную и конструктивную критику.

Надеемся, что материал сборника окажется полезным для вас и поможет успешно сдать Единый государственный экзамен по информатике и информационно-коммуникационным технологиям.

Ваши вопросы, замечания, пожелания, решения заданий вы можете присылать на адреса авторов, которые вы найдёте на сайте Национального исследовательского университета «Высшая школа экономики» www.hse.ru

Авторы также приглашают всех заинтересованных читателей в группы в социальных сетях:

http://www.facebook.com/groups/gia.ege/ и

http://vkontakte.ru/club30019188

Успехов и удачи!

Инструкция по выполнению работы

Экзаменационная работа состоит из трёх частей, содержащих 32 задания.

Часть 1 содержит 14 заданий с выбором ответа. К каждому заданию даётся 4 варианта ответа, из которых только один правильный.

Часть 2 состоит из 14 заданий с кратким ответом (к этим заданиям вы должны самостоятельно сформулировать и записать ответ).

Часть 3 состоит из 4 заданий. Для выполнения заданий этой части вам необходимо написать развёрнутый ответ в произвольной форме.

На выполнение частей 1 и 2 рекомендуется отводить 1,5 часа (90 минут). На выполнение заданий части 3 — 2,5 часа (150 минут).

В заданиях используются следующие *соглашения*:

1. Обозначения для логических связок (операций):
   - *отрицание* (инверсия, логическое НЕ) обозначается $\neg$ (например, $\neg A$);
   - *конъюнкция* (логическое умножение, логическое И) обозначается $\wedge$ (например, $A \wedge B$) либо & (например, A & B);
   - *дизъюнкция* (логическое сложение, логическое ИЛИ) обозначается $\vee$ (например, $A \vee B$);
   - *следование* (импликация) обозначается $\rightarrow$ (например, $A \rightarrow B$);
   - *строгая дизъюнкция* (исключающее ИЛИ, сложение по модулю 2) обозначается $\oplus$ (например, $A \oplus B$). Результатом операции строгой дизъюнкции является ИСТИНА, если значения A и B различны;
   - *эквиваленция* (равносильность, тождество) обозначается $\equiv$ либо $\leftrightarrow$ (например, $A \equiv B$, $A \leftrightarrow B$). Выражение $A \equiv B$ истинно тогда и только тогда, когда значения A и B совпадают (либо оба истинны, либо оба ложны);
   - символ 1 используется для обозначения истины (истинного высказывания); символ 0 — для обозначения лжи (ложного высказывания).

2. Обозначения для множеств:
   - объединение множеств обозначается $\cup$ (например, $A \cup B$);
   - пересечение множеств обозначается $\cap$ (например, $A \cap B$);
   - мощность множества обозначается $|\ \ |$ (например, $|A|$).
3. Два логических выражения, содержащих переменные, называются равносильными (эквивалентными), если значения этих выражений совпадают при любых значениях переменных. Так, выражения $A \to B$ и $(\neg A) \vee B$ равносильны, а $A \vee B$ и $A \wedge B$ — нет (значения выражений разные, например, при $A = 1$, $B = 0$).
4. Приоритеты логических операций в порядке убывания:
   1) инверсия (отрицание);
   2) конъюнкция (логическое умножение);
   3) дизъюнкция (логическое сложение), строгая дизъюнкция (исключающее ИЛИ);
   4) импликация (следование), эквиваленция (равносильность, тождество).

Таким образом, $\neg A \wedge B \vee C \wedge D$ означает то же, что и $((\neg A) \wedge B) \vee (C \wedge D)$.

Возможна запись $A \wedge B \wedge C$ вместо $(A \wedge B) \wedge C$.

То же относится и к дизъюнкции: возможна запись $A \vee B \vee C$ вместо $(A \vee B) \vee C$.

Контрольные тренировочные задания

Вариант 1 (*с решениями*)

Часть 1

При выполнении заданий части 1 в бланке ответов № 1 под номером выполняемого вами задания (А1—А14) поставьте знак «×» в клеточке, номер которой соответствует номеру выбранного вами ответа.

А1 Сколько значащих нулей в двоичной записи числа 1156?

1) 7 2) 8 3) 9 4) 6

Решение. Переведём число 1156 в двоичную систему счисления. Для того чтобы сделать это быстро, выделим максимальные степени двойки из числа, затем из оставшейся части числа и т.д.

$1156 - 1024 = 132;$

$132 - 128 = 4.$

Итак, $1156 = 1024 + 128 + 4 = 2^{10} + 2^7 + 2^2$.

Десятый, седьмой и второй разряды записи двоичного числа заполним единицами, остальные — нулями, получим: $1156_{10} = 10010000100_2$. (Нумерация разрядов в записи целого числа начинается с нуля и увеличивается справа налево.) Количество значащих нулей равно 8.

Заметим, что не обязательно записывать число в двоичной системе счисления, чтобы определить число значащих нулей. Выделив степени двойки, можно определить число разрядов и число единиц в двоичной записи числа. Этого достаточно, чтобы вычислить количество значащих нулей. Так, в данном примере общее количество разрядов двоичной записи числа равно 11 ($1024 = 2^{10} < 1156 < 2^{11}$), количество единиц равно 3. Количество значащих нулей равно $11 - 3 = 8$.

Ответ: 2.

См. учебно-справочные материалы: Перевод числа из одной позиционной системы счисления в другую, с. 44.

А2 Между населёнными пунктами A, B, C, D, E, F построены дороги, протяжённость которых приведена в таблице.

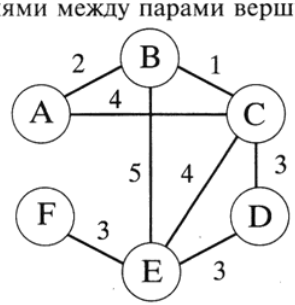
(Отсутствие числа в таблице означает, что прямой дороги между пунктами нет.)

	A	B	C	D	E	F
A		2	4			
B	2		1		5	
C	4	1		3	4	
D			3		3	
E		5	4	3		3
F					3	

Определите длину кратчайшего пути между пунктами A и F при условии, что передвигаться можно только по построенным дорогам.

1) 9 2) 10 3) 11 4) 12

Решение. Для решения задачи представим населённые пункты и расстояния между ними в виде графа. Вершины графа соответствуют населённым пунктам, они обозначены кружками, в которых записаны их обозначения. Дороги между ними обозначены линиями (рёбрами графа). Пометим рёбра числами-расстояниями между парами вершин. Получим:



Для решения воспользуемся алгоритмом Дейкстры поиска кратчайшего расстояния от одной вершины графа до других. Определим понятия:

- метка вершины X — расстояние (длина пути) от вершины A до вершины X. В решении длину пути будем обозначать AX;

- посещённая вершина — вершина, для которой определено минимальное расстояние от вершины A;
- соседи вершины X — все вершины, в которые ведут рёбра из вершины X:

Алгоритм работает пошагово — на каждом шаге «посещается» одна вершина и делается попытка уменьшить значения меток. Посещённые вершины на следующих шагах алгоритма не рассматриваются. Алгоритм Дейкстры останавливается, когда все вершины помечены как посещённые.

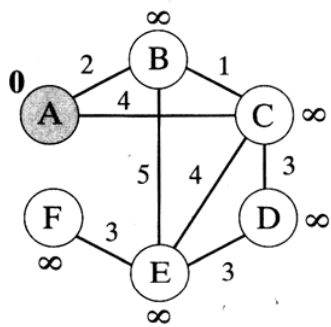
Шаг 0. Начальной вершине A присвоим метку 0, всем остальным вершинам — метку ∞. Все вершины пока непосещённые.

Шаг 1. Из непосещённых вершин выбираем вершину V с минимальной меткой. Для каждого соседа вершины V, кроме отмеченных как посещённые, определим новую длину пути, равную сумме значений текущей метки V и длины ребра, соединяющего V с этим соседом. Если полученное значение длины меньше значения метки соседа, заменим значение метки полученным значением длины. Рассмотрев всех соседей, пометим вершину V как посещённую.

Шаг 2. Если вершина F помечена как посещённая, завершим работу алгоритма, иначе — выполним Шаг 1.

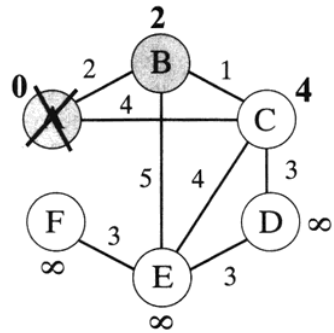
Выполним алгоритм. Будем записывать метки вершин и помечать посещённые вершины зачёркиванием.

Шаг 0. Пометим вершины. Метки запишем рядом с вершинами.

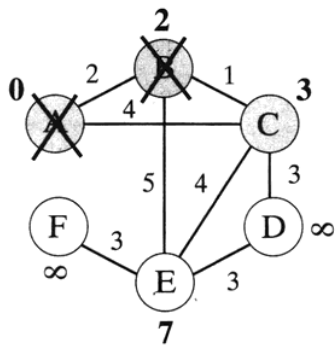


Шаг 1-1. Выберем вершину с минимальной меткой — это вершина A (она выделена цветом). Определим метки соседей, сложив метку вершины A с расстоянием до соседней вершины. Если полученное значение меньше метки соседней вершины, запишем его как метку соседней вершины:

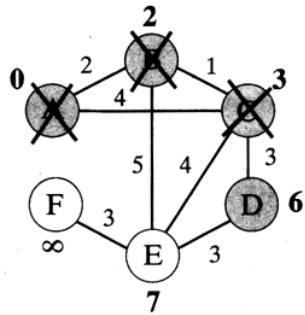
AB = 0 + 2 = 2; 2 < ∞;
AC = 0 + 4 = 4; 4 < ∞.
Пометим вершину A как посещённую (зачёркиванием).



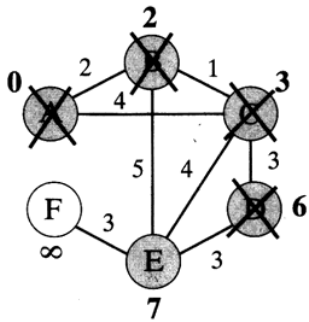
Шаг 1-2. Из непосещённых вершин выберем вершину с минимальной меткой — это вершина B (выделена цветом). Определим метки её соседей (посещённую вершину A не рассматриваем).
AC = 2 + 1 = 3; 3 < 4;
AE = 2 + 5 = 7; 7 < ∞.
Вершине C присвоим метку 3, вершине E — метку 7. Вершину B пометим как посещённую.



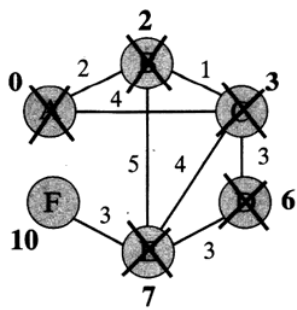
Шаг 1-3. Из непосещённых вершин выберем вершину с минимальной меткой — это вершина C. Определим метки её соседей (посещённые вершины A и B не рассматриваем).
AD = 3 + 3 = 6; 6 < ∞;
AE = 3 + 4 = 7; 7 = 7.
Вершине D присвоим метку 6, метка вершины E не изменится. Вершину C пометим как посещённую.



Шаг 1-4. Из непосещённых вершин выберем вершину D с минимальной меткой. Определим метку её соседа E, посещённые вершины не рассматриваем.
AE = 6 + 3 = 9; 9 > 7.
Метка вершины E не изменится. Вершину D пометим как посещённую.



Шаг 1-5. Из непосещённых вершин E и F выберем вершину E с минимальной меткой. Определим метку её соседа F.
AF = 7 + 3 = 10; 10 < ∞.
Метка вершины F равна 10, это и есть кратчайшее расстояние от вершины A до вершины F. Вершину E пометим как посещённую.



Вершина F не имеет непосещённых соседей, пометим её как посещённую, выполнение алгоритма завершается.

В результате метка каждой вершины соответствует минимальному расстоянию от A до этой вершины.

Ответ: 2.

См. учебно-справочные материалы: Элементы теории графов, с. 124.

A3 Дан фрагмент таблицы истинности выражения **F**:

X	**Y**	**Z**	**F**
0	0	1	**0**
0	1	1	**0**
1	0	1	**1**

Каким может быть выражение **F**?

1) $\neg X \vee \neg Y \vee Z$
2) $X \wedge \neg Y \wedge Z$
3) $X \vee \neg Y \vee Z$
4) $\neg X \wedge Y \wedge \neg Z$

Решение. Если бы таблица истинности была задана полностью, т.е. на всех наборах переменных X, Y и Z, можно было бы решить это задание построением КНФ или ДНФ[2].

Самый простой способ решения этого задания — подстановка значений переменных из строк таблицы истинности в предложенные в ответах выражения и сравнение результата вычисления выражения с соответствующим значением F в заданном фрагменте таблицы истинности. Выражение, значения которого для всех трёх наборов аргументов будут совпадать со значениями, указанными в таблице, и есть правильный ответ.

[2] Конъюнктивная нормальная форма или дизъюнктивная нормальная форма.

X	Y	Z	F	¬X > ¬Y > Z	X < ¬Y < Z	X > ¬Y > Z	¬X < Y < ¬Z
0	0	1	**0**	1	0	1	0
0	1	1	**0**	1	0	1	0
1	0	1	**1**	1	1	1	0

Значения выражения варианта ответа № 2 совпадают с **F** на всех заданных наборах.

Ответ: 2.

См. учебно-справочные материалы: Нормальные формы логических функций, с. 97.

A4 Для групповых операций с файлами используются **маски имён файлов.** Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы:

символ «?» (вопросительный знак) означает ровно один произвольный символ;

символ «*» (звёздочка) означает любую последовательность символов произвольной длины, в том числе символ «*» может задавать и пустую последовательность.

Определите, по какой из масок будет выбрана указанная группа файлов:

```
cceeeec.c
ebcb.res
ecbe.res
eeeecc.exe
```

1) `*c*.??**`
2) `?c*.*?`
3) `*c*.*`
4) `*cc*.*`

Решение. Определим закономерность в именах файлов этой группы. Во-первых, обратим внимание на расширения этих файлов. Один из них (`cceeeec.c`) имеет расширение, со-

стоящее только из одного символа. Таким образом, ответ 1 неверный, так как он подразумевает наличие двух и более символов в расширении. Рассмотрим ответ 2. Он подразумевает, что на втором месте в имени файла стоит символ «c». Но это неверно для двух файлов: `ebcb.res` и `eeeecc.exe`. Рассмотрим ответ 4. Он тоже неверен в силу того, что только в двух файлах из четырёх встречается двойной символ «c». Таким образом, верным является ответ 3: символ «*» допускает любое количество символов как перед, так и после символа «c» в имени файла, а также в расширении.

Можно составить таблицу, в которой запишем, удовлетворяет ли имя файла маске. Если хотя бы одно имя файла не соответствует маске, остальные имена для этой маски проверять не надо.

	1) *c*.??**	2) ?c*.*?	3) *c*.*	4) *cc*.*
cceeeec.c	–	+	+	+
ebcb.res		–	+	–
ecbe.res			+	
eeeecc.exe			+	

Ответ: 3.

См. учебно-справочные материалы: Поиск файлов, с. 36.

A5 Автомат получает на вход два трёхзначных числа. По этим числам строится новое число по следующим правилам:

1. Записывается результат сложения значений старших разрядов двух заданных чисел.
2. К нему дописывается результат сложения значений средних разрядов этих чисел по такому правилу: если он меньше первой суммы, то второе полученное число приписывается к первому слева, иначе — справа.
3. Итоговое число получают приписыванием справа к полученному после второго шага числу суммы значений младших разрядов исходных чисел.

Пример. Исходные трёхзначные числа: 138, 212. Поразрядные суммы: 3, 4, 10. Результат: 3410.

Определите, какое из предложенных чисел может быть результатом работы автомата.

1) 3507 2) 13412 3) 91216 4) 13197

Решение. Решим задачу с помощью рассуждений. Итоговое число состоит из трёх частей, каждая из которых представляет собой сумму двух однозначных чисел.

- Сумма цифр старших разрядов принадлежит отрезку [2, 18]. Минимальные цифры в старших разрядах трёхзначного числа равны 1, минимальная сумма старших разрядов равна 2. Максимальные цифры в разрядах равны 9, максимальная сумма равна 18.
- Суммы цифр остальных разрядов принадлежат отрезку [0, 18].

По заданию можно определить, что вторая часть числа должна быть больше первой. Рассмотрим каждое из предложенных чисел:

1) 3507 — варианты представления числа: 35 0 7, 3 50 7 и 3 5 07. В первых двух вариантах представления значения 35 и 50 больше 18. В третьем варианте десятичная запись числа начинается с 0 (07). Ответ неверный.
2) 13412 — разобьём число на части, учитывая, что каждая часть числа должна быть меньше 19: 13, 4, 12. 13 > 4, ответ неверный.
3) 91216 — можно разбить число на 9, 12 и 16. 9 < 12. Следовательно, вариант ответа может быть верным.
4) 13197 — каким бы образом мы не разбили это число на три части, одна из них всегда будет больше 18, значит, вариант ответа неверный.

Ответ: 3.

См. учебно-справочные материалы: Основы алгоритмизации, с. 161.

А6 База данных гостиницы состоит из трёх связанных таблиц. Ниже приведены фрагменты этих таблиц.

Таблица «Бронирование гостиничных номеров»

ID_ брони	ID_постояльца	ID_номера	Дата заезда	Дата выезда	Количество суток
11117	D01	131	24.01.11	16.02.11	24

Продолжение таблицы

ID_ брони	ID_по-стояльца	ID_но-мера	Дата заезда	Дата выезда	Количество суток
11125	D13	114	19.03.11	30.03.11	11
11141	D34	131	02.06.11	30.06.11	28
11152	D05	111	12.04.11	24.04.11	12
11158	D34	211	09.07.11	05.09.11	58
11169	D45	476	17.04.11	25.04.11	9
11174	D67	211	28.09.11	18.10.11	20
11178	D13	425	15.11.11	30.11.11	15

Таблица «Постояльцы»

ID_постояльца	Фамилия	Пол	Номер паспорта
D01	Васильев	М	1280567
D05	Смирнова	Ж	3560246
D13	Иванова	Ж	1730567
D34	Фёдоров	М	1956230
D45	Ильин	М	1854634
D67	Андреева	Ж	9672549

Таблица «Гостиничные номера»

ID_номера	Телефон	Количество комнат	Категория	Стоимость в сутки
111	7575-111	2	люкс	2500,00
114	7575-114	1		1700,00
131	7575-131	2	люкс	2700,00
211	7575-211	1		2000,00
425	7575-425	3	люкс	4000,00
476	7575-476	1		1500,00

По данным в представленных фрагментах таблиц определите, какую выручку получит гостиница за номера категории «люкс», заказанные клиентами женского пола.

1) 30 000 2) 60 000 3) 90 000 4) 148 700

Решение. Выясним по таблице «Гостиничные номера» **ID_номера** категории «люкс».

ID_номера	Телефон	Количество комнат	Категория	Стоимость в сутки
111	7575-111	2	люкс	2500,00
131	7575-131	2	люкс	2700,00
425	7575-425	3	люкс	4000,00

В таблице «Постояльцы» находим **ID_постояльца** женского пола:

ID_постояльца	Фамилия	Пол	Номер паспорта
D05	Смирнова	ж	3560246
D13	Иванова	ж	1730567
D67	Андреева	ж	9672549

В таблице «Бронирование гостиничных номеров» находим строки, в которых **ID_номера** и **ID_постояльца** равны полученным.

ID_брони	ID_постояльца	ID_номера	Дата заезда	Дата выезда	Количество суток
11152	D05	111	12.04.11	24.04.11	12
11178	D13	425	15.11.11	30.11.11	15

Считаем выручку, которую получит гостиница:

$$12 \cdot 2500 + 15 \cdot 4000 = 90\ 000.$$

Ответ: 3.

См. учебно-справочные материалы: Реляционная база данных, с. 244.

A7 В динамической (электронной) таблице приведены значения количества студентов, изучающих иностранные языки на разных курсах, и количество отличников из них. Укажите изучаемый иностранный язык, для которого общий процент отличников максимальный.

Курс	Количество студентов	Язык			
		английский	французский	немецкий	китайский
1	всего	160	140	130	50
	из них отличников	25	15	30	3
2	всего	120	120	120	30
	из них отличников	24	12	12	7
3	всего	130	110	150	20
	из них отличников	12	13	20	5
4	всего	100	70	125	10
	из них отличников	10	15	17	4
Итого	**всего**	**510**	**440**	**525**	**110**
	из них отличников	**71**	**55**	**79**	**19**

1) английский
2) французский
3) немецкий
4) китайский

Решение. Следует оценить отношение отличников к общему количеству студентов, изучающих соответствующий язык. Получим:
английский: 71 / 510 ≈ 0,14;
французский: 55 / 440 ≈ 0,125;
немецкий: 79 / 525 ≈ 0,15;
китайский: 19 / 110 ≈ 0,17.

Частное от деления довольно сложно вычислить без калькулятора. Предлагаем оценить отношение следующим образом:

1) английский: $71 \cdot 7 = 497$; $71 \cdot 8 = 568$; следовательно, $71 \cdot 7 < 510 < 71 \cdot 8$, и $1 / 8 < 71 / 510 < 1 / 7$;
2) французский: $55 / 440 = 1 / 8$;
3) немецкий: $1 / 7 < 79 / 525 < 1 / 6$;
4) китайский: $1 / 6 < 19 / 110 < 1 / 5$.

Наиболее успешно студенты изучают китайский язык.

Ответ: 4.

A8 Производится одноканальная монозвукозапись с частотой дискретизации 16 кГц и 24-битным разрешением. Запись длится 1 минуту, её результаты записываются в файл, сжатие данных не производится. Какое из приведённых ниже чисел наиболее близко к размеру полученного файла, выраженному в мегабайтах?

1) 0,2 2) 2 3) 3 4) 4

Решение. Измерение звука производится 16 000 раз в секунду или $16000 \cdot 60$ раз в минуту. Для сохранения результата каждого измерения требуется 24 бит = 3 байт. Всего $16\,000 \cdot 60 \cdot 3$ байт. Вычислим размер файла в мегабайтах:
$16\,000 \cdot 60 \cdot 3 / 2^{20} = 2^7 \cdot 125 \cdot 2^2 \cdot 15 \cdot 3 / 2^{20} = 125 \cdot 15 \cdot 3 / 2^{11} \approx$
$\approx 6000 / 2048 \approx 3$.

Ответ: 3.

См. учебно-справочные материалы: Кодирование звуковой информации, с. 70.

A9 Для кодирования некоторой последовательности, состоящей из букв **А**, **Б**, **В**, **Г** и **Д**, решили использовать неравномерный двоичный код, позволяющий однозначно декодировать двоичную последовательность, появляющуюся на приёмной стороне канала связи. Использовали код: А – 1, Б – 000, В – 001, Г – 011. Укажите, каким кодовым словом должна быть закодирована буква **Д**.

Длина этого кодового слова должна быть наименьшей из всех возможных. Код должен удовлетворять свойству однозначного декодирования.

1) 00 2) 0101 3) 11 4) 010

Решение. В задании используется префиксный код — код со словами переменной длины, в котором ни одно кодовое слово не является началом другого кодового слова. Он позволя-

ет однозначно декодировать сообщение без специального разделения кодов символов.

Проверим, являются ли предложенные ответы началом заданных кодовых слов или, наоборот, заданные коды — началом кодов, предложенных в ответах.

1) 00 — является началом кода символа Б (000).
2) 0101 — не является началом ни одного из заданных кодов символов, ни один из заданных кодов не является началом этого кода.
3) 11 — код символа А (1) является началом этого кода.
4) 010 — не является началом ни одного из заданных кодов символов, ни один из заданных кодов не является началом этого кода.

Таким образом, кодом буквы **Д** может быть 0101 и 010. По условию задания требуется выбрать кодовое слово минимальной длины, следовательно, верным ответом является код 010.

Ответ: 4.

См. учебно-справочные материалы: Префиксные коды, с. 8.

A10 Какое из приведённых имён удовлетворяет логическому условию:

¬ (первая буква гласная) →
→ (последняя буква согласная ∧ в слове больше 5 букв)?

1) ПОЛИНА 2) ГЛЕБ 3) СЕРГЕЙ 4) НИКИТА

Решение. Введём обозначения высказываний:

А — первая буква гласная,
В — последняя буква согласная,
С — в слове больше 5 букв.

Логическое условие можно записать в виде выражения $\neg A \to (B \wedge C) = A \vee B \wedge C$.

Проще всего решить данную задачу с помощью построения таблицы истинности:

	A	**B**	**C**	**B ∧ C**	**A ∨ B ∧ C**
ПОЛИНА	0	0	1	0	0
ГЛЕБ	0	1	0	0	0
СЕРГЕЙ	0	1	1	1	1
НИКИТА	0	0	1	0	0

Правильный ответ – СЕРГЕЙ.

Ответ: 3.

См. учебно-справочные материалы: Основные законы алгебры логики, с. 92.

A11 При сдаче экзамена в университете в память компьютера заносится индивидуальный буквенно-цифровой код студента и его оценка. Код состоит из 12 символов. В качестве символов используют 26 букв английского алфавита и десятичные цифры в любом порядке. При этом используют посимвольное кодирование и все символы кодируются одинаковым и минимально возможным количеством битов. Оценка — число от 0 до 10 — также кодируется с использованием минимально возможного и одинакового количества битов. Код студента вместе с его оценкой записывается минимально возможным и целым числом байтов. Определите объём файла, содержащего такую информацию, после сдачи экзамена 48 студентами.

1) 624 байт 2) 480 байт 3) 384 байт 4) 528 байт

Решение. Количество возможных символов в индивидуальном коде студента 26 букв + 10 цифр = 36. Определим минимально необходимое количество битов для кодирования 36 различных символов. При мощности алфавита, равной двум (для кодирования числа используются цифры 0 и 1), количество разрядов (бит) для кодирования одного из 36 различных символов определяется по формуле:

$$\lceil I \rceil = \lceil \log_2 N \rceil$$

где I — количество битов для кодирования одного символа, а N — количество возможных различных символов. Запись $\lceil X \rceil$ означает, что X округляется до ближайшего большего целого. Для кодирования одного символа нам потребуется $\lceil \log_2 36 \rceil = 6$ бит. На 12 символов кода необходимо $6 \cdot 12 = 72$ бит. Для кодирования оценки необходимо $\lceil \log_2 11 \rceil = 4$ бит. Итого на каждого студента $72 + 4 = 76$ бит, или 10 байт. Информация о 48 студентах займёт $48 \cdot 10 = 480$ байт.

Ответ: 2.

См. учебно-справочные материалы: Измерение информации, с. 11.

A12 В программе используется одномерный целочисленный массив **A** с индексами от 0 до 10 и целочисленные переменные **tmp** и **i**. Ниже представлен фрагмент программы, записанный на разных языках программирования, в котором значения элементов сначала задаются, а затем меняются.

Бейсик	Паскаль

FOR i=0 TO 10
  A(i)=i
NEXT i
FOR i=0 TO 9 STEP 3
  TMP = A(i)
  A(i) = A(i+1)
  A(i+1) = TMP
NEXT i
	for i:=0 to 10 do
	  A[i]:=i;
	for i:=0 to 9 do
	  if (i mod 3)=0
	  then
	  begin
	  tmp := A[i];
	  A[i]:=A[i+1];
	  A[i+1]:=tmp;
	  end;

Си	**Алгоритмический язык**

for (i=0;i<=10;i++)
  A[i]=i;
for (i=0;i<10;i+=3)
{
  tmp= A[i];
  A[i]=A[i+1];
  A[i+1]=tmp;
}
	нц для i от 0 до 10
	  A[i]:=i
	кц
	нц для i от 0 до 9 шаг 3
	  tmp := A[i]
	  A[i]:=A[i+1]
	  A[i+1]:=tmp
	кц

Чему будут равны элементы этого массива после выполнения фрагмента программы?

1) 3 1 2 6 4 5 9 7 8 0 10
2) 1 0 3 2 5 4 7 6 9 8 10
3) 1 0 2 4 3 5 7 6 8 10 9
4) 0 1 2 3 4 5 6 7 8 9 10

Решение. В задаче используются два оператора цикла с параметром. В первом цикле

Бейсик	Паскаль

FOR i=0 TO 10
  A(i)=i
NEXT i
	for i:=0 to 10 do
	  A[i]:=i;

Си	Алгоритмический язык

`for (i=0;i<=10;i++)`
`  A[i]=i;`
	`нц для i от 0 до 10`
	`  A[i]:=i`
	`кц`

массив А заполняется числами от 0 до 10:

i	0	1	2	3	4	5	6	7	8	9	10
A[i]	0	1	2	3	4	5	6	7	8	9	10

Во втором цикле

Бейсик	Паскаль

`FOR i=0 TO 9 STEP 3`
`  TMP = A(i)`
`  A(i) = A(i+1)`
`  A(i+1) = TMP`
`NEXT i`
	`for i:=0 to 9 do`
	`  if ( i mod 3 )=0`
	`  then`
	`  begin`
	`  tmp := A[i];`
	`  A[i]:=A[i+1];`
	`  A[i+1]:=tmp;`
	`  end;`

Си	**Алгоритмический язык**

`for (i=0;i<10;i+=3)`
`{`
`  tmp= A[i];`
`  A[i]=A[i+1];`
`  A[i+1]=tmp;`
`}`
	`нц для i от 0 до 9 шаг 3`
	`  tmp := A[i]`
	`  A[i]:=A[i+1]`
	`  A[i+1]:=tmp`
	`кц`

параметр цикла `i` изменяется с шагом 3 и принимает значения 0, 3, 6, 9. В теле цикла меняются местами значения переменных `A[i]` и `A[i+1]`, т. е. значения A[0] и A[1], A[3] и A[4], A[6] и A[7], A[9] и A[10]:

	A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]	A[10]
исх	0	1	2	3	4	5	6	7	8	9	10
i=0	1	0	2	3	4	5	6	7	8	9	10
i=3	1	0	2	4	3	5	6	7	8	9	10
i=6	1	0	2	4	3	5	7	6	8	9	10
i=9	1	0	2	4	3	5	7	6	8	10	9

Значения A[2] = 2, A[5] = 5, A[8] = 8 остались без изменений. В результате получим:

i	0	1	2	3	4	5	6	7	8	9	10
A[i]	1	0	2	4	3	5	7	6	8	10	9

Ответ: 3.

См. учебно-справочные материалы: Одномерные массивы, с. 218.

A13 Система команд исполнителя РОБОТ, «живущего» в прямоугольном лабиринте на клетчатой плоскости:

вверх	**вниз**	**влево**	**вправо**

При выполнении любой из этих команд РОБОТ перемещается на одну клетку соответственно: вверх ↑, вниз ↓, влево ←, вправо →.

Четыре команды проверяют истинность условия отсутствия стены у каждой стороны той клетки, где находится РОБОТ:

сверху свободно	**снизу свободно**	**слева свободно**	**справа свободно**

Цикл ПОКА <*условие*> *команда* выполняется, пока условие истинно, иначе происходит переход на следующую строку.

Если РОБОТ начнёт движение в сторону стены, то он разрушится и программа прервётся.

Сколько клеток лабиринта соответствует требованию, что, выполнив предложенную программу, РОБОТ уцелеет и остановится в той же клетке, с которой он начал движение?

НАЧАЛО

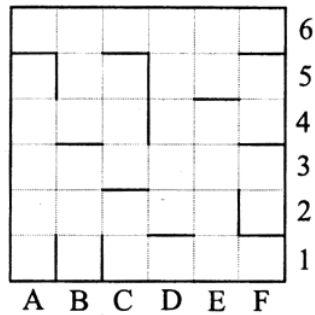
ПОКА <**слева свободно**> **вверх**

ПОКА <**сверху свободно**> **вправо**

ПОКА <**справа свободно**> **вниз**

ПОКА <**снизу свободно**> **влево**

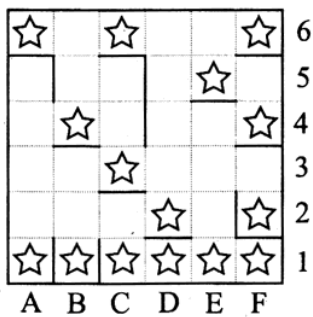
КОНЕЦ



1) 0 2) 1 3) 2 4) 4

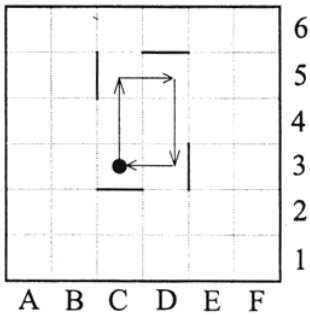
Решение. Очевидным, но трудоёмким способом решения данной задачи является проверка всех клеток поля. Сократим перебор.

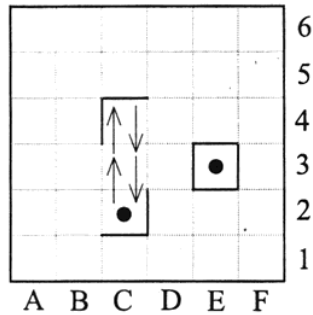
Заметим, что РОБОТ заканчивает выполнение алгоритма в клетке, ограниченной стеной снизу, следовательно, надо проверить только клетки, имеющие стенку снизу. Отметим эти клетки:



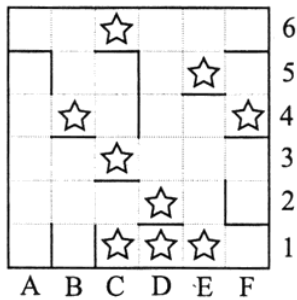
На рисунке отмечено максимальное количество клеток, которые потребуется проверять. Продолжим рассуждения, чтобы сократить число проверяемых клеток.

Для того чтобы РОБОТ успешно вернулся в начальную клетку, требуется выполнение алгоритма, задающего траекторию движения РОБОТА. В общем случае траекторией является прямоугольник, движение начинается вверх и идёт по часовой стрелке. Возможные виды траекторий показаны на рисунках ниже.



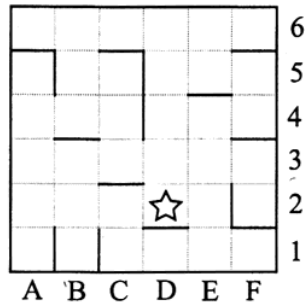


Отметим, что при выполнении заданного алгоритма РОБОТ всегда разрушается, если клетка ограничена тремя стенами. Удалим отметки с таких клеток:



Для оставшихся отмеченных клеток проверим выполнение алгоритма:

B4	Вверх B5 — вправо C5 — влево B5 – разрушился при движении влево
C1	Вправо D1 — разрушился при движении вниз
C3	Вверх C4, C5 — разрушился при движении вверх
C6	Разрушился при движении вверх
D1	Разрушился при движении вверх
D2	Вверх D3, D4 — вправо E4 — вниз E3, E2 — влево D2
E1	Вверх E2, E3, E4, разрушился при движении вверх
E5	Вверх E6, разрушился при движении вверх
F4	Вверх F5, разрушился при движении вверх



Отмеченной осталась одна клетка.

Ответ: 2.

См. учебно-справочные материалы: Формальный исполнитель РОБОТ, с. 179.

A14 Определите, какое число будет напечатано в результате работы следующей программы (для вашего удобства программа представлена на четырёх языках):

Бейсик	Паскаль

`Module A14`
`  Sub Main()`
`  Dim d, a, b, t, M, R As Double`
`  a = -2 : b = 3`
`  d = 0.1`
`  t = a: M = a: R = F(a)`
`  While t <= b`
`  If F(t) < R Then`
`  M = t`
`  R = F(t)`
`  End If`
`  t = t + d`
`  End While`
`  Console.Write(M)`
`End Sub`
`  Function F(ByVal x As Double)`
`As Double`
`  Return (x−1) * (x−3)`
`  End Function`
`End Module`
	`Program A14;`
	`Uses crt;`
	`Var d,a,b,t,M,R :real;`
	`Function`
	`F(x : real):real;`
	`  begin`
	`  F:=(x-1)*(x-3);`
	`  end;`
	`BEGIN`
	`a:=-2; b:=3;`
	`d:=0.1;`
	`t:=a; M:=a; R:=F(a);`
	`while t<=b do`
	`  begin`
	`  if (F(t)<R) then`
	`  begin`
	`  M:=t;`
	`  R:=F(t);`
	`  end;`
	`  t:=t+d;`
	`  end;`
	`write(M);`
	`END.`

Си	Алгоритмический язык

`#include <stdio.h>`
`double F(double x)`
`{`
`  return (x-1)*(x-3);`
`}`
`void main()`
`{`
`  double d, a, b, t, M, R;`
`  a = -2; b = 3;`
`  d = 0.1;`
`  t = a; M = a; R = F(a);`
`  while (t<=b) {`
`  if ( F(t)<R ) {`
`  M = t; R = F(t);`
`  }`
`  t = t + d;`
`  }`
`  printf("%.0f", M);`
`}`
	`алг A14`
	`нач`
	`  вещ d, a, b, t, M, R`
	`  a:= -2; b:= 3`
	`  d:= 0.1`
	`  t:=a; M:=a; R:=F(a)`
	`  нц пока t<=b`
	`  если F(t)< R`
	`  то`
	`  M:= t; R:= F(t)`
	`  все`
	`  t:= t + d`
	`  кц`
	`  вывод M`
	`кон`
	`алг вещ F(вещ x)`
	`нач`
	`  знач := (x-1)*(x-3)`
	`кон`

1) –1 2) –2 3) 2 4) 3

Решение. Для решения этого задания нерационально использовать трассировку программы, так как тело цикла выполняется $(b - a) / d + 1 = 51$ раз.

Следует понять, что именно вычисляется в программе. Это поможет определить результат её работы.

Переменные a, b, d используются для организации цикла.

В программе используется функция пользователя $F(x)$, которая возвращает значение выражения $(x-1)*(x-3)$. Значение аргумента в теле функции не изменяется[3].

В цикле вычисляется минимальное значение функции $F(t)$ на интервале $[a, b]$ с шагом d. В переменной R сохраняется значение функции $F(t)$, претендующее на минимальное. В переменной M сохраняется значение аргумента t, при котором функция принимает минимальное значение.

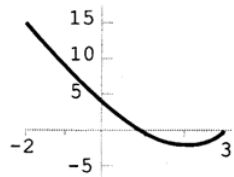
Таким образом, в программе определяется значение аргумента функции F, при котором функция принимает минимальное значение, на отрезке $[-2;3]$.

[3] Параметры функции передаются в неё по значению.

Запишем функцию:

$F(x) = (x - 1)*(x - 3) = x^2 - 4*x + 3$.

Получили уравнение параболы ($px^2 + qx + k$), ветви которой направлены вверх ($p > 0$), а вершина находится в точке $x_в = -q / 2p = 4 / 2 = 2$, $x_в \in [-2;3]$.



В вершине параболы её значение минимально. Таким образом, на печать будет выведено число 2.

Ответ: 3.

См. учебно-справочные материалы: Подпрограммы, определяемые разработчиком, с. 232.

Часть 2

Ответом к заданиям части 2 (B1–B14) является число, последовательность букв или цифр. Впишите ответы сначала в текст работы, а затем перенесите их в бланк ответов № 1 справа от номера соответствующего задания, начиная с первой клеточки, без пробелов, запятых и других дополнительных символов. Каждую букву или цифру пишите в отдельной клеточке в соответствии с приведёнными в бланке образцами.

B1 Автоматическое устройство осуществило перекодировку информационного сообщения на русском языке длиной 75 символов, первоначально записанного в 16-битной кодировке Unicode, в 8-битный код КОИ-8. На сколько битов уменьшилась длина сообщения? *В ответе запишите только число.*

Решение. Сообщение состоит из 75 символов. Первоначально оно имело объём 75 · 2 байт, после перекодирования — 75 · 1 байт. Следовательно, объём сообщения уменьшился на

$75 \cdot 2 - 75 = 75$ байт = 600 бит.

Ответ: 600.

См. учебно-справочные материалы: Кодирование текстовой информации, с. 60.

B2 У исполнителя Вычислитель две команды, которым присвоены номера:

```
1. приписать 3 справа
2. вычесть 1
```

Первая из них приписывает к числу на экране справа цифру 3, вторая вычитает из него 1.

Запишите порядок команд в алгоритме получения из числа 7 числа 532, содержащем не более 5 команд, указывая только номера команд (например, 21222 — это алгоритм:

```
2. вычесть 1
1. приписать 3 справа
2. вычесть 1
2. вычесть 1
2. вычесть 1
```

который преобразует число 4 в число 30).

Если таких алгоритмов более одного, запишите любой из них.

Решение. Построим дерево выполнения команд. В результате выполнения команды 1 получим $10 \cdot x + 3$, где x — число, к которому применяется команда.

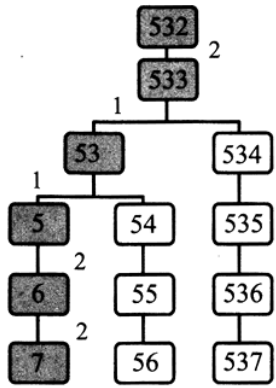
Команду 1 можно применить ко всем числам. Но обратную к ней команду «вычесть 3 и разделить на 10» можно применить только к числам, в которых последняя цифра 3. Таких чисел в десять раз меньше, и обратная команда может применяться в 10 раз реже. Команда 2 и обратная ей могут применяться ко всем числам. Вывод: количество ветвей дерева, построенного от результата 532 к исходному числу 7 с использованием команд, обратных заданным, меньше, чем при построении дерева от заданного числа 7 к результату 532.

Обратные команды:

```
1. вычесть 3 и разделить на 10
2. прибавить 1
```

Из каждого узла дерева могут выходить две ветви: левая соответствует выполнению обратной команды 1, правая —

выполнению обратной команды 2. Если команда 1 не может быть выполнена, из узла выходит одна ветвь, соответствующая команде 2.



Запишем последовательность команд получения из числа 7 числа 532 (от листа к корню дерева).

Ответ: 22112.

См. учебно-справочные материалы: Формальный исполнитель Вычислитель, с. 185.

B3 Определите, что будет напечатано в результате работы следующего фрагмента программы:

Бейсик	Паскаль

`Dim a, b`
`  a = 75`
`  b = 10 – a Mod 2`
`  While a – 4 * b > 2`
`  If a > 30 Then`
`  a = a – 30`
`  Else : a = a + b + 10`
`  End If`
`  End While`
`  Console.Write(a)`
	`var a,b: integer;`
	`BEGIN`
	`  a:=75;`
	`  b:=10 – a mod 2;`
	`while a-4*b>2 do`
	`  begin`
	`  if a>30 then`
	`  a:=a-30`
	`  else`
	`  a:=a+b+10;`
	`  end;`
	`  write(a);`
	`END.`

Си	Алгоритмический язык

`{`
`  int a,b;`
`  a = 75;`
`  b = 10 – a % 2;`
`  while (a-4*b)>2`
`  if(a>30)`
`  a = a – 30;`
`  else`
`  a = a + b + 10;`
`  printf("%d", a);`
`}`
	`нач`
	`  цел a,b`
	`  a:=75`
	`  b:=10-mod(a,2)`
	`  нц пока a-4*b>2`
	`  если a>30 то`
	`  a:=a-30`
	`  иначе`
	`  a:=a+b+10`
	`  все`
	`  кц`
	`  вывод a`
	`кон`

Решение. В алгоритме используется циклическая структура с предусловием, в теле цикла выполняется структура ветвления — полная форма команды «если». Составим трассировочную таблицу первого вида:

№	Команда или условие (логическое выражение)	Вычисление правой части команды присваивания или условия (логического выражения)	a	b
1	a := 75	75	75	
2	b := 10 – mod(a, 2)	10 – 1 = 9		9
3	a – 4*b > 2	75 – 4*9 = 75 – 36=39; (39 > 2) = да		
4	a > 30	(75 > 30) = да		
5	a := a – 30	75 – 30 = 45	45	
6	a – 4*b > 2	45 – 4*9 = 9; (9 > 2) = да		
7	a > 30	(45>30) = да		
8	a := a – 30	45 – 30 = 15	15	
9	a – 4*b > 2	15 – 4*9 = 15 – 36 = –21; (–21 > 2) = нет		
Результат:			15	9

Ответ: 15.

См. учебно-справочные материалы: Алгоритмы, заданные на алгоритмическом языке, с. 193.

B4 Для кодирования сообщений решено использовать последовательности разной длины из двух знаков: € и ₣. Сколько различных символов можно закодировать, используя подобный код длиной не менее пяти и не более десяти знаков (€ и ₣)?

Решение. Мощность кодового алфавита равна количеству знаков, $M = 2$. Если длина кода символа равна I, то можно закодировать $N = M^I$ различных символов.

Способ 1. Если длина кода может быть от 5 до 10, общее количество различных символов определяется как сумма:

$$s = \sum_{i=5}^{10} 2^i.$$

Или $2^5 + 2^6 + 2^7 + 2^8 + 2^9 + 2^{10} = 32 + 64 + 128 + 256 + 512 + 1024 = 2016$.

Способ 2. Заметим, что последовательность количества символов при $I = 5, ..., 10$ образует геометрическую прогрессию с первым членом $b_1 = 2^5$ и знаменателем прогрессии $q = 2$. Сумма n членов геометрической прогрессии при $q \neq 1$, равна:

$$s = b_i \frac{1-q^n}{1-q}.$$

Для решения задачи найдём сумму первых $10 - 5 + 1 = 6$ членов геометрической прогрессии:

$$s = 2^5 \frac{1-2^6}{1-2} = 2^5 \times (2^6 - 1) = 32 \times 63 = 2016.$$

Другой подход к решению — найти сумму первых 10 членов геометрической прогрессии с первым элементом $b_1 = 2$, затем вычесть из неё сумму первых четырёх членов геометрической прогрессии с первым элементом $b_1 = 2$:

$$s = 2\frac{1-2^{10}}{1-2} - 2\frac{1-2^4}{1-2} = 2(1023 - 15) = 2016.$$

Ответ: 2016.

См. учебно-справочные материалы: Кодирование информации, с. 7; Измерение информации, с. 11; Позиционные системы счисления, с. 40.

B5 Дан фрагмент электронной таблицы:

	A	B	C	D
1	1	1	1	
2	=A1+$B$1	=B1+2	=C1+$D$1	=D1+A1
3				

Диапазон ячеек A2:D2 скопировали в диапазон A3:D3.

Какое число должно быть записано в ячейке D1, чтобы построенная по значениям диапазона ячеек A3:D3 диаграмма соответствовала рисунку?

Решение. Выполним копирование ячеек, учитывая, что абсолютные части ссылок в формулах не меняются, а относительные меняются при копировании:

	A	B	C	D
1	1	1	1	
2	=A1+$B$1	=B1+2	=C1+$D$1	=D1+A1
3	=A2+$B$1	=B2+2	=C2+$D$1	=D2+A2

Проведём возможные вычисления в ячейках таблицы, а также преобразуем формулы, получим:

	A	B	C	D
1	1	1	1	
2	2	3	=C1+$D$1	=D1+A1
3	3	5	=C2+$D$1	=D2+A2

На диаграмме первый сектор меньше трёх других, которые имеют одинаковые значения. Из таблицы следует, что первый сектор соответствует ячейке A3, равной 3, а три оставшихся сектора — значениям ячеек B3 = C3 = D3 = 5. В ячейке C3 должно находиться значение 5, следовательно,

C2+$D$1 = 5.

Подставим формулу из ячейки C2, получим:

1+$D$1+$D$1 = 5;

$D$1 = 2.

В ячейке D1 должно быть записано число 2. Проверим значение в D3:

D3 = D2+A2 = D1+A1+A2 = 5.

Ответ: 2.

B6 Определите значение переменной *c* после выполнения следующего фрагмента программы, в котором *a*, *b* и *c* — переменные вещественного (действительного) типа.

Бейсик	Паскаль

`a = 74`
`b = 36`
`IF b > a / 2 – 1 THEN`
`  c = b + a / 4`
`  ELSE`
`  c = b + a`
`ENDIF`
	`a:= 74;`
	`b:= 36;`
	`if b > a / 2 – 1`
	`then`
	`  c := b + a / 4`
	`else`
	`  c:= b + a;`

Си	**Алгоритмический**

`a = 74;`
`b = 36;`
`if (b > a / 2 – 1)`
`  c = b + a / 4;`
`else`
`  c = b + a;`
	`a := 74`
	`b := 36`
	`если b > a / 2 – 1`
	` то c := b + a / 4`
	` иначе c := b + a`
	`все`

Решение. Построим трассировочную таблицу первого вида:

№	Команда или условие (логическое выражение)	Вычисление правой части команды присваивания или условия (логического выражения)	a	b	c
1	a = 74	74	74		
2	b = 36	36		36	
3	b > a / 2 – 1	(36 >36) = нет			
4	c = b + a	36 + 74 = 110			110
Результат			74	36	110

Ответ: 110.

См. учебно-справочные материалы: Алгоритмы, заданные на алгоритмическом языке, с. 193.

B7 Ниже приведены тексты одной и той же программы, записанные на четырёх языках программирования. Что будет напечатано в результате выполнения этой программы?

Алгоритмический язык

```
алг Задача
нач
      цел L=5; целтаб R[1:L]; цел N, p
      R[1]:=1;R[2]:=2;R[3]:=3;R[4]:=0;R[5]:=2
      ArrProcessing_1(L,R,p)
      если p=0 то
            вывод "Переполнение"; стоп
      все
      N:=MultCounter(L,R)
      вывод N
кон
алг ArrProcessing_1(цел L, аргрез целтаб R[1:L],
рез цел p)
нач
      цел i,t
      p:=0
      нц для i от 2 до L
            t:=R[i]+R[i-1]*p
            R[i]:=mod(t,2)
            p:=div(t,2)
      кц
кон
алг цел MultCounter(цел L, аргрез целтаб R[1:L])
нач
      цел N, i, T
      N:=0
      T:=1
      нц для i от 1 до L
            N:=N+T*R[i]
            T:=2*T*(i)
      кц
      знач:=N
кон
```

Бейсик

```
Module Task
    Sub Main()
        Dim L As Integer = 5
        Dim R(5) As Integer
        Dim N, p As Integer
        R.SetValue(1, 1)
        R.SetValue(2, 2)
        R.SetValue(3, 3)
        R.SetValue(0, 4)
        R.SetValue(2, 5)
        ArrProcessing_1(L, R, p)
        If p = 0 Then
            Console.Write("Переполнение")
            Stop
        End If
        N = MultCounter(L, R)
        Console.Write(N)
    End Sub
    Sub ArrProcessing_1(ByVal L As Integer, ByRef
R As Array, ByRef p As Integer)
        Dim i, t As Integer
        p = 0
        For i = 2 To L
          t = R.GetValue(i)+R.GetValue(i - 1) * p
            R.SetValue(t Mod 2, i)
            p = t \ 2
        Next
    End Sub
    Function MultCounter(ByVal L As Integer, ByRef
R As Array) As Integer
        Dim N, i, T As Integer
        N = 0
        T = 1
        For i = 1 To L
            N = N + T * R.GetValue(i)
            T = 2 * T * i
        Next
        Return N
    End Function
End Module
```

Паскаль

```
Program Task;
Uses crt;
const L = 5;
type atype = array [1..L] of integer;
Var R: atype; N,p: integer;
Procedure ArrProcessing_1(L: integer; var R:
atype; var p:integer);
var i,t: integer;
begin
     p:=0;
     for i:=2 to L do
     begin
          t:=R[i]+R[i-1]*p;
          R[i]:=t mod 2;
          p:=t div 2;
     end;
end;
Function MultCounter(L: integer; var R:
atype):integer;
var N,i,T: integer;
begin
     N:=0;
     T:=1;
     for i:=1 to L do
     begin
          N:=N+T*R[i];
          T:=2*T*i;
     end;
     MultCounter:=N;
end;
BEGIN
     clrscr;
     R[1]:=1;R[2]:=2;R[3]:=3;R[4]:=0;R[5]:=2;
     ArrProcessing_1(L,R,p);
     if p=0 then
     begin
          write('Переполнение');
          halt;
     end;
     N:=MultCounter(L,R);
     write(N);
END.
```

Си

```
void ArrProcessing_1(int L, int* R, int* p)
{
     int i, t;
     *p = 0;
     for(i=1; i<L; i++)
     {
              t = R[i] + R[i-1]**p;
              R[i] = t % 2;
              *p = t / 2;
     }
 }
 int MultCounter(int L, int* R)
 {
     int N, i, T;
     N = 0;
     T = 1;
     for (i=0; i<L; i++)
     {
         N = N + T*R[i];
         T=2*T*(i+1);
     }
     return N;
}

void main()
{
    int L = 5;
    int* R = (int*) calloc(L, sizeof(int));
    int N, p;
   R[0]=1; R[1] = 2; R[2] = 3; R[3] = 0; R[4] = 2;
    ArrProcessing_1(L, R, &p);
    if (p==0)
    {
            printf("Переполнение");
            free(R);
            return;
    }
    N = MultCounter(L,R);
    printf("%d\n", N);
    free(R);
}
```

Решение. Перед тем как приступить к вычислениям, проанализируем структуру программы и сформулируем некоторые наблюдения. Это поможет сэкономить время во время экзамена.

1) В заданиях, требующих определить, что будет выведено на экран, в первую очередь необходимо выявить все команды вывода на экран. В данной задаче такая команда одна: `printf("%d\n", N)`.
   В результате работы данной команды на экран выведется значение переменной `N`, его и потребуется отследить.
2) Заметим, что в основной программе осуществляются вызовы вспомогательных функций[4] `ArrProcessing_1()` и `MultCounter()`, следовательно, прямая трассировка программы усложняется.
3) Напомним, что при описании функции в её заголовке в скобках записываются так называемые формальные параметры, при этом обязательно указывается их тип. В теле функции они используются так же, как переменные.

При вызове функции на выполнение вместо формальных параметров необходимо подставить фактические. Ими могут быть константы, выражения, имена переменных соответствующих типов.

Важно понимать, что формальные параметры функции могут быть двух видов в зависимости от способа передачи их в функцию при её вызове и от того, «видит» ли вызывающая функцию программа изменения, произошедшие с параметрами в теле функции.

Первый тип.

Для передачи параметров по ссылке:

В Паскале применяются параметры-переменные (Паскаль), перед их описанием обязательно служебное слово `var`.

В Бейсике после имени параметра обязательно используется служебное слово `ByRef`.

В Си используется тип указатель.

При вызове функции вместо этих параметров можно подставлять только переменные соответствующего типа. Изменения значений этих переменных будут «видны» вызывающей программе.

---

[4] Некоторые языки программирования, например Pascal и Basic, относят к функциям подпрограммы, возвращающей значение, а к процедурам — подпрограммы, не возвращающей значения. При решении задач в данном пособии будем использовать более общую терминологию, предполагая, что процедура есть частный случай функции.

Второй тип.

Для передачи параметров по значению:

В Паскале применяются параметры-значения, перед ними не ставится слово `var`.

В Бейсике после имени параметра обязательно используется служебное слово `ByVal`.

В Си используются параметры простых типов и типов пользователя, но не указатели.

При вызове функции на выполнение вместо этих параметров можно подставлять

- константы
- выражения
- переменные

соответствующих типов.

Изменения фактических параметров «не видны» вызывающей функцию программе.

Таким образом:

- при передаче параметров по значению функция работает с копиями переданных ей данных, в вызывающей программе их значения не изменяются,
- при передаче параметров по ссылке функция работает (и, возможно, изменяет) непосредственно с переданными ей данными, в вызывающей программе их значения также меняются.

В нашей задаче по ссылке в функцию `ArrProcessing_1()` передаются массив и переменная `p`, в функцию `MultCounter()` — массив.

```
…
void ArrProcessing_1(int L, int* R, int* p)
…
ArrProcessing_1(L, R, &p);
```

Параметр передаётся по значению Параметры передаются по ссылке

4) Проанализируем действия, выполняющиеся в функциях. Функция `ArrProcessing_1()` возвращает значение типа `void` и занимается поэлементной обработкой массива, переданного в качестве параметра, по некоторому правилу. Именно изменённый по этому правилу массив R и является аргументом функции `MultCounter()`, которая вычисляет и возвращает целочисленное значение в переменную `N`. Это значение и будет выведено на экран.

5) Переменная `p` передаётся из вызывающей функции в `ArrProcessing_1()` по ссылке, следовательно, по завершении работы функции в `p` будет значение, вычисленное в функции.

Таким образом, решение задачи состоит из следующих шагов:

Шаг 1. Определить исходные значения элементов массива R.

Шаг 2. Определить значение переменной `p` и правило изменения элементов массива и их значений после выполнения функции `ArrProcessing_1()`.

Шаг 3. Определить значение, возвращаемое функцией `MultCounter()`.

Выполним решение по шагам.

Шаг 1. Исходные значения элементов массива R:

R[0]	R[1]	R[2]	R[3]	R[4]
1	2	3	0	2

Шаг 2. Трассировка вызова функции: `ArrProcessing_1 (L, R, &p)`:

i	t	R[0]	R[1]	R[2]	R[3]	R[4]	p
		1	2	3	0	2	0
1	R[1]+ +R[0]**p= 2+1*0=2		t%2= =2%2= =0				t/2= =2/2= =1
2	R[2]+ +R[1]**p= 3+0*1=3			t%2= =1%2= =1			t/2= =1/2= =0
3	R[3]+ +R[2]**p= 0+1*0=0				t%2= =1%2= =1		t/2= =0/2= =0
4	R[4]+ +R[3]**p= 2+1*0=2					t%2= =2%2= =0	t/2= =2/2= =1
Итог:		**1**	**0**	**1**	**1**	**0**	**1**

Шаг 3. Трассировка вызова функции: `MultCounter (L, R)`:

i	N	T
	0	1
0	N+T*R[0]=0+1*1=1	2*T*(i+1)=2*1*(0+1)=2
1	N+T*R[1]=1+2*0=1	2*T*(i+1)=2*2*(1+1)=8
2	N+T*R[2]=1+8*1=9	2*T*(i+1)=2*8(2+1)=48
3	N+T*R[3]=9+48*1=57	2*T*(i+1)=2*48(3+1)=384
4	N+T*R[4]=57+384*0=57	2*T*(i+1)=2*384(4+1)=3840
Итог:	**57**	**3840**

Из последней таблицы видно, что в переменную N вызывающей программы попадёт значение 57, именно оно будет выведено на экран.

Ответ: 57.

См. учебно-справочные материалы: Подпрограммы, определяемые разработчиком, с. 232.

B8 Запись числа 30_{10} в системе счисления с основанием N оканчивается на 0 и содержит 4 цифры. Чему равно основание этой системы счисления N?

Решение. Запишем число в N-ичной системе счисления $XYZ0_N$, где X, Y, Z — цифры, которые могут принимать значения от 0 до N−1.

Развёрнутая запись числа имеет вид:

$$X \cdot N^3 + Y \cdot N^2 + Z \cdot N = 30.$$

Вынесем сомножитель N за скобки и разложим 30 на простые множители:

$$N \cdot (X \cdot N^2 + Y \cdot N + Z) = 2 \cdot 3 \cdot 5.$$

N — целое число. Из последнего равенства видно, что N является сомножителем в разложении числа 30 на простые множители. Таким образом, N может быть равно 2, 3, 5, 6, 10, 15.

Сразу исключим из рассмотрения 10 и 15, так как запись числа в N-ичной системе счисления не совпадает с записью

в десятичной по условию, а запись в 15-ричной системе счисления должна выглядеть меньше 30.

Далее задачу можно решать двумя способами.

1) Последовательно перевести 30 в двоичную, троичную, пятеричную, шестеричную системы счисления, пока не будет получена четырёхзначная запись, оканчивающаяюся на ноль.
2) Составить неравенство $(XYZ0_N)_{min} \leqslant 30 \leqslant (XYZ0_N)_{max}$. Напомним, что если в системе счисления с основанием N используются K цифр для записи числа, то
   - минимальное K-значное число равно N^{K-1}. В его записи старшая цифра — единица, остальные цифры — нули, например 1000.
   - максимальное K-значное число равно $N^K - 1$ и состоит из K старших цифр системы счисления, например, 9999_{10}, 1111_2, 5555_6. Значение старшей цифры равно $(N - 1)$.

Запись минимального четырёхзначного числа в любой системе счисления имеет вид 1000_N, его значение равно N^3.

Для определения максимального значения четырёхзначного числа $XYZ0_N$ из максимально возможного четырёхзначного числа $(N^4 - 1)$ вычтем максимально возможную последнюю цифру $(N - 1)$, так как младший разряд по условию задачи равен 0:

$$(N^4 - 1) - (N - 1) = N^4 - N.$$

Таким образом, можно записать неравенство:

$$N^3 \leqslant 30 \leqslant N^4 - N.$$

Проверим выполнение неравенства для возможных значений N (2, 3, 5, 6):

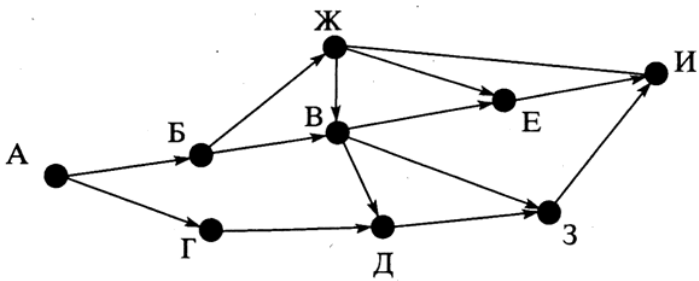
N	$N^3 \leqslant 30$	$30 \leqslant N^4 - N$
2	$8 \leqslant 30$ да	$30 \leqslant 16 - 2$ нет
3	$27 \leqslant 30$ да	$30 \leqslant 81 - 3$ да
5	$125 \leqslant 30$ нет	–
6	–	–

Неравенства выполняются для N = 3, запись числа в троичной системе счисления имеет вид 1010_3.

Ответ: 3.

См. учебно-справочные материалы: Позиционные системы счисления, с. 40.

В9 На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город И?



Решение. Введём функцию

$$F(Q)=\begin{cases} K, & Q \neq X \\ 1, & Q = X \end{cases}$$

где Q — название населённого пункта, K — количество различных путей, ведущих из населённого пункта Q в населённый пункт X.

Будем называть соседями города, в которые можно попасть непосредственно из города Q.

Для вычисления количества путей из А в И определим соседей А, т.е. в какие города ведут дороги из А. Количество путей из А в И будет складываться из количества путей до И из соседей А. Далее последовательно будем определять количество путей из очередного рассматриваемого города как сумму путей его соседей.

Вычисления завершаются, когда в правой части F(А) останутся лишь слагаемые F(И).

При выполнении преобразований будем записывать слагаемые в алфавитном порядке обозначений городов[5].

[5] Преобразования обычно выполняются в порядке топологической сортировки вершин, здесь алгоритм топологической сортировки не описан и не применяется.

Из А выходят две дороги: в Б и в Г, следовательно, количество путей из А в И — это сумма количества путей из Б в И и из Г в И:
$F(А) = F(Б) + F(Г)$.

Из Б можно попасть в Ж и в В, из Г можно попасть только в Д. Подставим
$F(Б) = F(В) + F(Ж)$ и $F(Г) = F(Д)$
в правую часть $F(А)$, получим:
$F(А) = F(В) + F(Д) + F(Ж)$.

Далее определим слагаемые правой части:
$F(Д) = F(З)$;
$F(В) = F(Е) + F(З) + F(Д) = F(Е) + 2 \cdot F(З)$;
$F(Ж) = F(В) + F(Е) + F(И) = 2 \cdot F(Е) + 2 \cdot F(З) + F(И)$;
подставим их в правую часть $F(А)$ и приведём подобные:
$F(А) = F(Е) + 2 \cdot F(З) + F(З) + 2 \cdot F(Е) + 2 \cdot F(З) + F(И) =$
$= 3 \cdot F(Е) + 5 \cdot F(З) + F(И)$.

Из города З, как и из города Е, есть только одна дорога до И, следовательно, можно записать: $F(З) = F(И)$ и $F(Е) = F(И)$, подставить в правую часть выражения. Получим результат:

$F(А) = 5 \cdot F(И) + 3 \cdot F(И) + F(И) = 9 \cdot F(И) = 9$.

Из города А в город И можно попасть девятью различными путями.

Ответ: 9.

B10 У Васи есть доступ в Интернет по высокоскоростному одностороннему радиоканалу, обеспечивающему скорость получения им информации 2^{22} бит/с. У Миши нет скоростного доступа в Интернет, но есть возможность получать информацию от Васи по низкоскоростному телефонному каналу со средней скоростью 2^{15} бит/с. Миша договорился с Васей, что тот будет скачивать для него данные объёмом 12 Мбайт по высокоскоростному каналу и ретранслировать их Мише по низкоскоростному каналу.

Компьютер Васи может начать ретрансляцию данных не раньше, чем им будет получен 1 Мбайт этих данных. Каков минимально возможный промежуток времени (в секундах) с момента начала скачивания Васей данных до полного их получения Мишей?

В ответе укажите только число, слово «секунд» или букву «с» добавлять не нужно.

Решение. Объём полученной (переданной) информации вычисляется по формуле:

V [бит] = v [бит/с]· t [с],

где v – скорость, а t – время передачи информации.

Для нахождения общего времени передачи данных от Васи к Мише составим уравнение. Пусть

t_1 – время, которое необходимо компьютеру Васи, чтобы скачать информацию объёмом V_1 = 1 Мб со скоростью $v_1 = 2^{22}$ бит/с и начать ретрансляцию Мише;

t_2 – время, которое необходимо компьютеру Миши, чтобы скачать информацию объёмом V_2 =12 Мб со скоростью $v_2 = 2^{15}$ бит/с с компьютера Васи. Тогда

$$t_{общ} = t_1 + t_2 = \frac{V_1}{v_1} + \frac{V_2}{v_2}.$$

Выразим все заданные в задаче объёмы информации в битах:

1 Мб = 2^{10} Кб = 2^{10+10} байт = 2^{20+3} бит = 2^{23} бит;

12 Мб = 12 · 2^{10} Кб = 12 · 2^{10+10} байт = 3 · 2^{20+3+2} бит = 3 · 2^{25} бит.

Вычислим время:

$t_1 = 2^{23}$ бит / 2^{22} бит/с = 2 с;

t_2 = 3 · 2^{25} бит / 2^{15} бит/с = 3 · 2^{10} с = 3 · 1024 с = 3072 с;

$t_{общ}$ = 3072 с + 2 с = 3074 с.

Ответ: 3074.

См. учебно-справочные материалы: Передача информации, с. 20.

B11 В терминологии сетей TCP/IP **маской сети** называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес. Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске.

По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 59.166.121.209

Маска: 255.255.248.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса сети и запишите в нужном порядке соответствующие им буквы, без использования точек.

A	B	C	D	E	F	G	H
0	54	209	59	109	166	120	248

Пример.

Пусть искомый IP-адрес сети — 192.168.120.0 и дана таблица

A	B	C	D	E	F	G	H
255	*11*	*166*	*120*	*127*	*0*	*192*	*168*

В этом случае правильный ответ будет записан в виде GHDF.

Решение. 1) Если значение байта маски сети $255=11111111_2$, то соответствующий ему байт адреса сети совпадает с соответствующим байтом IP-адреса.

2) Если значение байта маски сети равно нулю, то соответствующий байт адреса сети равен нулю.

3) Если значение байта маски сети лежит в диапазоне от 1 до 254, следует записать значение соответствующих байтов маски и IP-адреса в двоичной системе счисления и выполнить поразрядную конъюнкцию. Полученный результат есть соответствующий байт адреса сети.

Первые два байта IP-адреса сети совпадают с первыми двумя байтами IP-адреса узла, так как первые два байта маски равны 255.

Последний байт IP-адреса сети равен 0, так как последний байт маски равен 0.

Определим третий байт адреса сети:

$121 = 01111001_2$
$248 = 11111000_2$

Поразрядная конъюнкция выполняется в соответствии с таблицей истинности:

```
  01111001
& 11111000
  --------
  01111000
```

Переведём полученное число, записанное в двоичной системе счисления, в десятичную систему:
$01111000_2=120$.

Таким образом, IP-адрес сети: 59.166.120.0

Ответ: DFGA.

См. учебно-справочные материалы: Подсети. Маски подсетей, с. 270.

B12 В языке запросов поискового сервера для обозначения логической операции «ИЛИ» используется символ «|», для логической операции «И» — символ «&», для логической операции «НЕ» — символ «-».

В таблице приведены запросы и количество найденных по ним страниц некоторого сегмента сети Интернет.

Запрос	Найдено страниц (в тысячах)
Бетховен	1500
Классика & Оркестр	800
Классика & Бетховен	400
Оркестр & Бетховен	200
Классика & Оркестр & Бетховен	100

Какое количество страниц (в тысячах) будет найдено по запросу

(-*Классика & Оркестр & Бетховен*) |
(*Классика & -Оркестр & Бетховен*) |
(*Бетховен & -*(*Классика* | *Оркестр*))?

Считается, что все запросы выполнялись практически одновременно, так что набор страниц, содержащих все искомые слова, не изменялся за время выполнения запросов.

Решение. Введём обозначения: Б — множество страниц, полученных по запросу *Бетховен*, К — множество страниц, полученных по запросу *Классика*, О — множество страниц, полученных по запросу *Оркестр*.

|X| — мощность множества X, например, |Б| — это количество страниц, найденных по запросу *Бетховен*.

Перепишем запрос с использованием введённых обозначений:

(-*К & О & Б*) | (*К & -О & Б*) | (*Б & -*(*К* | *О*)).

Для нахождения результата требуемого запроса используем круги Эйлера (диаграммы Эйлера–Венна), каждый из которых будет соответствовать множеству Б, К и О. Найдём искомое множество поэтапно.

К О Б	К О Б	К О Б	К О Б
(-К & О & Б)	*(К & -О & Б)*	*Б & -(К | О)*	Результат запроса

Вычтем из количества страниц, найденных по запросу *Бетховен*, количество страниц по запросу *Классика & Оркестр & Бетховен*.

К О Б	К О Б
Б	*К & О & Б*

Количество страниц, найденных по запросу в задании, определяется по формуле:

$|Б| - |К\ \&\ О\ \&\ Б| = 1500 - 100 = 1400.$

Ответ: 1400.

См. учебно-справочные материалы: Элементы теории множеств, с. 132.

B13 У исполнителя Вычислитель две команды:

```
1. прибавь 3
2. вычти 2
```

Первая из них увеличивает число на экране на 3, вторая уменьшает его на 2 (отрицательные числа допускаются). Программа для Вычислителя — это последовательность команд. Сколько различных чисел можно получить из числа 1 с помощью программы, которая содержит ровно 5 команд?

Решение. В общем случае в дереве решений, содержащем 5 уровней, может быть $2^5 = 32$ листа. Построение такого дерева требует времени и внимания.

Поскольку обе команды исполнителя аддитивные (прибавляют положительное или отрицательное число), последовательность выполнения команд не имеет значения. Разные программы могут выдавать один и тот же результат. Например, в результате выполнения трёх разных программ:

Программа 1	Программа 2	Программа 3
прибавь 3	**прибавь 3**	**вычти 2**
вычти 2	**вычти 2**	**вычти 2**
вычти 2	**прибавь 3**	**прибавь 3**
вычти 2	**вычти 2**	**прибавь 3**
прибавь 3	**вычти 2**	**вычти 2**

будет получено число $1 + 2 \cdot 3 + 3 \cdot (-2) = 1$.

Таким образом, важно лишь количество команд 1 и 2 в программе.

Пусть x — количество команд 1 в программе, $0 \leqslant x \leqslant 5$, тогда количество команд 2 равно $(5 - x)$. В результате выполнения какой-либо программы из 5 команд будет получено число:

$k = 1 + x \cdot 3 + (5 - x) \cdot (-2) = 5 \cdot x - 9$.

x может принимать 6 разных значений (0, 1, 2, 3, 4, 5), следовательно, в результате выполнения программ, состоящих из 5 команд, будет получено 6 разных чисел.

Ответ: 6.

B14 Сколько различных решений имеет система уравнений

$$(a \wedge \neg b) \vee (\neg a \wedge b) \vee (c \wedge d) \vee (\neg c \wedge \neg d) = 1$$
$$(c \wedge \neg d) \vee (\neg c \wedge d) \vee (e \wedge f) \vee (\neg e \wedge \neg f) = 1$$
$$(e \wedge \neg f) \vee (\neg e \wedge f) \vee (g \wedge h) \vee (\neg g \wedge \neg h) = 1$$
$$(g \wedge \neg h) \vee (\neg g \wedge h) \vee (i \wedge j) \vee (\neg i \wedge \neg j) = 1$$

где $a, b, c, d, e, f, g, h, i, j$ — логические переменные? В ответе запишите количество различных наборов значений $a, b, c, d, e, f, g, h, i, j$, при которых выполняется данная система равенств.

Решение. 1. При решении будем использовать более удобные обозначения логических операций. Система уравнений будет иметь вид:

$$a\bar{b} \vee \bar{a}b \vee cd \vee \bar{c}\,\bar{d} = 1$$
$$c\bar{d} \vee \bar{c}d \vee ef \vee \bar{e}\,\bar{f} = 1$$
$$e\bar{f} \vee \bar{e}f \vee gh \vee \bar{g}\,\bar{h} = 1$$
$$g\bar{h} \vee \bar{g}h \vee ij \vee \bar{i}\,\bar{j} = 1$$

2. Так как все четыре логических выражения должны быть одновременно истинны, можно объединить их в одно операцией коньюнкции и приравнять единице (истине):

$$(a\bar{b} \vee \bar{a}b \vee cd \vee \bar{c}\,\bar{d})(c\bar{d} \vee \bar{c}d \vee ef \vee \bar{e}\,\bar{f}) \times$$
$$\times\ (e\bar{f} \vee \bar{e}f \vee gh \vee \bar{g}\,\bar{h})(g\bar{h} \vee \bar{g}h \vee ij \vee \bar{i}\,\bar{j}) = 1$$

Таблица истинности выражения, в которое входят 10 логических переменных, состоит из $2^{10} = 1024$ строк. Будем искать другой подход к решению.

3. Заметим, что в выражениях используются операции исключающего ИЛИ

$$x \oplus y = x\bar{y} \vee \bar{x}y$$

и эквиваленции

$$x \equiv y = xy \vee \bar{x}\,\bar{y},$$

которые связаны соотношением

$$x \equiv y = \overline{x \oplus y}$$

в соответствии с таблицей истинности

x	y	$x \oplus y$	$x \equiv y$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

4. Введём обозначения:

$$K = a \oplus b \qquad \bar{K} = a \equiv b$$
$$L = c \oplus d \qquad \bar{L} = c \equiv d$$
$$M = e \oplus f \qquad \bar{M} = e \equiv f$$
$$N = g \oplus h \qquad \bar{N} = g \equiv h \qquad (*)$$
$$O = i \oplus j \qquad \bar{O} = i \equiv j$$

Заметим, что переменные K, L, M, N, O взаимно независимы.

5. Система уравнений примет вид:

$$K \vee \bar{L} = 1$$

$$L \vee \bar{M} = 1$$

$$M \vee \bar{N} = 1$$

$$N \vee \bar{O} = 1$$

Объединим уравнения операцией конъюнкции, получим:

$$(K \vee \bar{L})(L \vee \bar{M})(M \vee \bar{N})(N \vee \bar{O}) = 1 \qquad (**)$$

6. Полученное уравнение можно решить двумя способами: построением таблицы истинности и приведением к СДНФ.

6.1. *Способ 1.* Построим таблицу истинности, она будет состоять из $2^5 = 32$ строк. Строки — решения уравнения (**) выделим цветом.

№ набора	K	L	M	N	O	K∨¬L	L∨¬M	M∨¬N	N∨¬O	Результат
0	0	0	0	0	0	1	1	1	1	1
1	0	0	0	0	1	1	1	1	0	0
2	0	0	0	1	0	1	1	0	1	0
3	0	0	0	1	1	1	1	0	1	0
4	0	0	1	0	0	1	0	1	1	0
5	0	0	1	0	1	1	0	1	0	0
6	0	0	1	1	0	1	0	1	1	0
7	0	0	1	1	1	1	0	1	1	0
8	0	1	0	0	0	0	1	1	1	0
9	0	1	0	0	1	0	1	1	0	0
10	0	1	0	1	0	0	1	0	1	0
11	0	1	0	1	1	0	1	0	1	0
12	0	1	1	0	0	0	1	1	1	0
13	0	1	1	0	1	0	1	1	0	0
14	0	1	1	1	0	0	1	1	1	0

Продолжение таблицы

№ набора	*K*	*L*	*M*	*N*	*O*	$K \vee \neg L$	$L \vee \neg M$	$M \vee \neg N$	$N \vee \neg O$	Результат
15	0	1	1	1	1	0	1	1	1	0
16	1	0	0	0	0	1	1	1	1	1
17	1	0	0	0	1	1	1	1	0	0
18	1	0	0	1	0	1	1	0	1	0
19	1	0	0	1	1	1	1	0	1	0
20	1	0	1	0	0	1	0	1	1	0
21	1	0	1	0	1	1	0	1	0	0
22	1	0	1	1	0	1	0	1	1	0
23	1	0	1	1	1	1	0	1	1	0
24	1	1	0	0	0	1	1	1	1	1
25	1	1	0	0	1	1	1	1	0	0
26	1	1	0	1	0	1	1	0	1	0
27	1	1	0	1	1	1	1	0	1	0
28	1	1	1	0	0	1	1	1	1	1
29	1	1	1	0	1	1	1	1	0	0
30	1	1	1	1	0	1	1	1	1	1
31	1	1	1	1	1	1	1	1	1	1

Получили шесть решений уравнения (**) для введённых обозначений (*).

6.2. *Способ 2.* Приведём выражение (**) к СДНФ. Известно, что в СДНФ количество слагаемых равно количеству строк таблицы истинности, в которых функция принимает значение «истина». Раскрывая скобки и используя закон непротиворечия ($A \wedge \neg A = 0$), получим:

$$(KL \vee K\bar{M} \vee \cancel{\bar{L}L} \vee \bar{L}\bar{M})(MN \vee M\bar{O} \vee \cancel{\bar{N}N} \vee \bar{N}\bar{O}) = 1$$

$$KLMN \vee KLM\bar{O} \vee KL\bar{N}\bar{O} \vee \cancel{K\bar{M}MN} \vee \cancel{K\bar{M}M\bar{O}} \vee K\bar{M}\bar{N}\bar{O} \vee$$
$$\vee \cancel{\bar{L}\bar{M}MN} \vee \cancel{\bar{L}\bar{M}M\bar{O}} \vee \bar{L}\bar{M}\bar{N}\bar{O} = 1$$

или

$$KLMN \vee KLM\bar{O} \vee KL\bar{N}\bar{O} \vee K\bar{M}\bar{N}O \vee \bar{L}\bar{M}\bar{N}\bar{O} = 1.$$

Получили ДНФ. Перейдём к СДНФ. Для этого неполные конъюнкции следует домножить на выражения вида $(A \vee \neg A = 1)$, где A — отсутствующая в неполной конъюнкции переменная[6]:

$$KLMN(O \vee \bar{O}) \vee KLM\bar{O}(N \vee \bar{N}) \vee KL\bar{N}\bar{O}(M \vee \bar{M}) \vee$$
$$\vee K\bar{M}\bar{N}O(L \vee \bar{L}) \vee \bar{L}\bar{M}\bar{N}\bar{O}(K \vee \bar{K}) = 1.$$

Раскроем скобки, при этом переменные в элементарных конъюнкциях будем записывать в лексикографическом порядке, и используем закон идемпотентности (тавтологии, $A \vee A = A$), получим:

$$KLMNO \vee KLMN\bar{O} \vee \cancel{KLMN\bar{O}} \vee KLM\bar{N}\bar{O} \vee \cancel{KLM\bar{N}\bar{O}} \vee$$
$$\vee KL\bar{M}\bar{N}\bar{O} \vee \cancel{KL\bar{M}\bar{N}\bar{O}} \vee K\bar{L}\bar{M}\bar{N}\bar{O} \vee \cancel{K\bar{L}\bar{M}\bar{N}\bar{O}} \vee \bar{K}\bar{L}\bar{M}\bar{N}\bar{O} = 1$$

или

$$KLMNO \vee KLMN\bar{O} \vee KLM\bar{N}\bar{O} \vee KL\bar{M}\bar{N}\bar{O} \vee K\bar{L}\bar{M}\bar{N}\bar{O} \vee$$
$$\vee \bar{K}\bar{L}\bar{M}\bar{N}\bar{O} = 1.$$

СДНФ содержит шесть слагаемых.

7. Все переменные таблицы истинности (и сомножители каждого слагаемого в СДНФ) взаимно независимы. Каждая из пяти переменных (*) в соответствии с таблицей истинности операций исключающего ИЛИ и эквиваленции принимает значение «истина» при двух наборах переменных.

   Таким образом, каждый набор в таблице истинности и каждое слагаемое в СДНФ будут принимать значение «истина» на $2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 2^5$ наборах значений исходных переменных a, b, c, d, e, f, g, h, i, j.

   Тогда общее количество наборов будет равно $2^5 \cdot 6 = 192$, так как в таблице истинности шесть наборов удовлетворяют условию задачи и СДНФ содержит шесть слагаемых.

Ответ: 192.

См. учебно-справочные материалы: Построение СДНФ логической функции, с. 102.

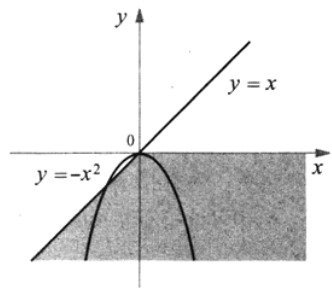
Не забудьте перенести все ответы в бланк ответов № 1.

[6] Полная конъюнкция содержит в качестве сомножителей все логические переменные или их отрицания.

Часть 3

Для записи ответов к заданиям части 3 (С1–С4) используйте бланк ответов № 2. Запишите сначала номер задания (С1, С2 и т.д.), а затем полное решение. Ответы записывайте чётко и разборчиво.

С1 Требовалось написать программу, при выполнении которой с клавиатуры считываются координаты точки на плоскости (x, y — действительные числа) и определяется принадлежность этой точки заданной закрашенной области (включая границы), неограниченной снизу. Программист торопился и написал программу неправильно.



Паскаль	Бейсик	Си

var x, y: real;
begin
readln(x, y);
if y <= x then
if y <= 0 then
if y <= -x*x
then
write
('принадлежит')
else
write
('не принадлежит')
end.
	INPUT x,y
	IF y <= x THEN
	IF y <= 0 THEN
	IF y <= -x*x THEN
	PRINT
	"принадлежит"
	ELSE
	PRINT
	"не принадлежит"
	ENDIF
	ENDIF
	ENDIF
	END
		void main(void)
		{float x,y;
		scanf
		("%f%f",&x,&y);
		if (y <= x)
		if (y <= 0)
		if (y <= -x*x)
		printf
		("принадлежит");
		else
		printf
		("не принадлежит");
		}

Последовательно выполните следующее:

1) Приведите пример таких чисел x, y, при которых программа неверно решает поставленную задачу. Объясните, почему для указанных чисел программа неверно решает поставленную задачу.
2) Укажите, как нужно доработать программу, чтобы не было случаев её неправильной работы. (Это можно сделать не-

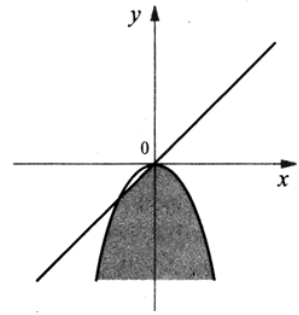
сколькими способами, достаточно указать любой способ доработки исходной программы.)

Решение. 1) Приведённым в программе трём ограничениям удовлетворяют точки плоскости, для которых истинно:

($y <= x$) И ($y <= 0$) И ($y <= -x^2$).

Очевидно, что обозначенная область не совпадает с приведённой в условии задачи.

Таким образом, программа неверно решает поставленную задачу для точек из заштрихованной области, например, при $x = 2$, $y = -1$.



2) При невыполнении первого или второго условия программа не выдаёт сообщений из-за отсутствия в первых двух условных операторах ветви ELSE.

Отметим также, что ограничение, связанное с параболой, требуется накладывать только на область $x <= 0$ и $y <= 0$. Для области $x > 0$ и $y <= 0$ никаких дополнительных ограничений не требуется.

3) Возможная доработка программы — заменить все операторы ветвления программы на предложенные:

Паскаль

```
if (y<=0) and (x>=0) or (x<=0) and (y<=x) and (y<=-x*x) then
  writeln('принадлежит')
  else
  writeln('не принадлежит');
```

Бейсик

```
IF y<=0 AND x>=0 OR x<=0 AND y<=x AND y<=-x*x THEN
  PRINT "принадлежит"
  ELSE
  PRINT "не принадлежит"
ENDIF
```

Си

```
if (((y<=0)&&(x>=0))||  ((x<=0)&&(y<=x)&&(y<=-x*x)))
  printf("принадлежит");
  else
  printf("не принадлежит");
```

C2 Дан массив из 50 целых чисел. В массиве хранятся баллы, набранные командами на олимпиаде по программированию. Один элемент массива соответствует количеству баллов, набранных одной командой, и имеет значение от 0 до 100 включительно. Необходимо разработать программу, которая по данным массива вычисляет максимальное количество баллов среди команд, показавших зачётный результат, и количество команд с максимальным зачётным количеством очков. Зачётное количество очков 55. Гарантируется, что хотя бы одна команда набрала зачётное количество очков.

Фрагмент программы с объявлением переменных и заполнением массива приведён ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из описанных переменных.

Паскаль	Бейсик

`const n = 50;`
`var  a:array[1..n] of integer;`
`  i,max,k:integer;`
`begin`
`  for i:=1 to n do`
`  readln(a[i]);`
`  .....`
`end.`
	`N = 50`
	`DIM A(N) AS INTEGER`
	`DIM I, MAX, K AS INTEGER`
	`FOR I = 1 TO N`
	`  INPUT A(I)`
	`NEXT I`
	`.....`
	
	`END`

Си	**Естественный язык**

`#include <stdio.h>`
`#define N  50`
`void main()`
`{`
`  int a[N],i,max,k;`
`  for(i=0; i<N; i++)`
`  scanf("%d",&a[i]);`
`  .....`
`}`
	Объявляем массив A из 50 элементов. Объявляем целочисленные переменные I, MAX, K. В цикле от 1 до 50 вводим элементы массива A с 1-го по 50-й.
	.....

В качестве ответа вам необходимо привести фрагмент программы (или описание алгоритма на естественном языке), кото-

рый должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например Borland Pascal 7.0) или в виде блок-схемы. В этом случае вы должны использовать переменные, аналогичные переменным, используемым в алгоритме, записанном на естественном языке, с учётом синтаксиса и особенностей используемого вами языка программирования.

Решение. Пусть a_i — количество баллов, набранных i-й командой на олимпиаде, а max — максимальное количество набранных баллов. Опишем алгоритм нахождения количества команд, набравших максимальные баллы:

1. Положим, max = 55 (минимальное зачётное количество очков);
2. Положим, $k = 0$ (пока ни одну команду мы не проверили);
3. Положим, $i = 1$;
4. Если a_i = max, увеличиваем значение k на единицу: $k = k + 1$;
5. Если $a_i > 55$, заменяем значения max и k: max = a_i, $k = 1$;
6. Если $i \leqslant n$, увеличиваем значение i на единицу: $i = i + 1$ и переходим к шагу 4;
7. Если $i > n$, завершаем повторения.
8. Печатаем значение k.

Фрагмент программы с решением:

Паскаль

```
  max:=55;
  k:=0;
  for i:=1 to n do
  begin
   if a[i] = max then k:=k+1;
   if (a[i] > max) then
        begin
             max:=a[i];
             k:=1;
        end
   end;
writeln(max,' ',k);
```

Бейсик

```
MAX = 55
K = 0
FOR I = 1 TO N
  IF A(I) = MAX THEN K = K + 1
  IF A(I) > MAX THEN
    MAX = A(I)
    K = 1
  END IF
NEXT I
PRINT MAX
PRINT K
```

Си

```
  max=55;
  k=0;
  for(i=0; i<N; i++)
  {
    if(a[i] == max) k++;
    if(a[i] > max)
    {
       max=a[i];
       k=1;
    }
  }
  printf("%d %d",max, k);
```

Естественный язык

Записываем в переменную MAX начальное значение, равное 55, и в переменную K начальное значение, равное 0. В цикле от первого до последнего элемента выполняем два сравнения. Если значение текущего элемента совпадает со значением MAX, увеличиваем значение переменной K на единицу. Если значение текущего элемента массива превышает значение MAX, заменяем значение MAX на значение текущего элемента массива и устанавливаем значение переменной K, равным единице. После завершения цикла выводим значения переменных MAX и K.

C3 У исполнителя Калькулятор две команды, которым присвоены номера:

```
1. умножь на 2
2. прибавь 5
```

Первая команда умножает число на 2, вторая увеличивает его на 5.

Программа для Калькулятора — это последовательность команд.

Сколько есть программ, которые число 3 преобразуют в число 47?

Решение. В задаче нет ограничения на количество команд в программе. Отметим, что обе команды увеличивают число. Если в результате выполнения какой-либо программы будет получено число больше 47, дальнейшее выполнение команд не приведёт к необходимому результату.

Задание можно решать разными способами. Рассмотрим два из них: построение дерева решения и вычисление результата по рекуррентной формуле[7].

Способ 1. Построение дерева выполнения команд. Напомним, что подобное дерево используется при решении задачи В2, правила его построения подробно рассмотрены в учебно-справочных материалах (тема 7.6.2, с.185 — 189).

В отличие от задачи В2, где мы искали самую короткую программу получения одного числа из другого, в этом задании необходимо найти количество вершин, содержащих результат. Если строить прямое дерево (получение числа 47 из числа 3), оно будет полным, так как из каждой вершины можно выполнить обе команды. Если использовать обратные команды

```
1. раздели на 2
2. вычти 5
```

дерево будет меньшего размера, так как не все числа кратны двум и команда 1 не может быть выполнена для нечётных чисел (предполагается, что все числа — целые).

Будем строить дерево решений получения числа 3 из числа 47 с использованием обратных команд. Корнем дерева является вершина со значением требуемого результата 47, так как команды только увеличивают значение числа. Задача — найти количество вершин со значением 3. К каждой из них ведёт свой путь, т. е. своя программа. Так как неизвестно ко-

[7] Рекуррентной называется формула, выражающая n-й член числовой последовательности через несколько предыдущих её членов. Например, вычисление чисел Фибоначчи задаётся формулами: $a_0 = 0$, $a_1 = 1$, $a_{n+2} = a_{n+1} + a_n$ ($n \geqslant 0$).

личество команд в программах (количество уровней дерева), будем строить дерево в глубину.

Будем считать, что левая ветвь дерева соответствует команде 1, правая ветвь — команде 2. Если из вершины может выйти только одна ветвь (команда 2), направим её вниз. Листьями дерева будут вершины, значения которых равно или меньше 3, они обозначены прямоугольниками. Выделим цветом листья, значение которых равно 3.

Сначала из корня дерева построим левые ветви до листьев, затем вернёмся выше на ближайший уровень, из которого ещё не построена правая ветвь, и продолжим построение.

Проиллюстрируем построение дерева. Первая программа состоит из последовательности команд 212111 и получает из числа 47 число

$$((47 - 5) / 2 - 5) / 2 / 2 / 2 = 2<3$$

(см. рис. 1). Этот лист не выделен на рисунке цветом, так как не получено требуемое число 3.

Возвращаемся вверх до первой вершины, из которой ещё не была выполнена вторая команда, — это вершина 4, выполняем команду 2 (вычти 5). Получим лист со значением $-1 < 3$.

Ещё раз возвращаемся вверх, на этот раз до листа 8, выполним команду 2, получим число 3 — лист дерева с нужным результатом. Этот лист выделим цветом (рис. 2).

Затем выполним команду 2 из листа 16, получим 11, продолжим построение до получения листьев 3 (выделен цветом) и 1 (рис. 3).

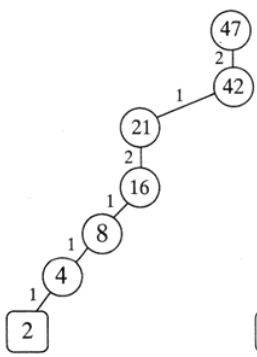


Рис. 1

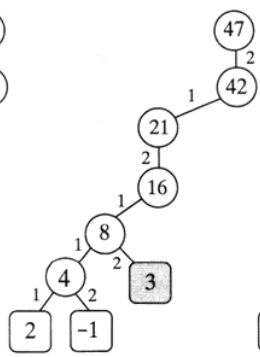


Рис. 2

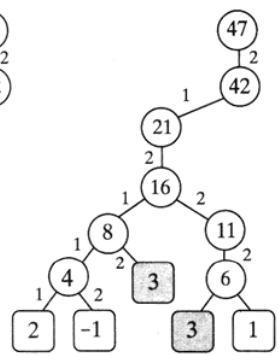


Рис. 3

Следующая команда выполняется от вершины 42. При построении получим последовательность вершин 37, 32, 16. Обратим внимание на то, что вершина 16 нам уже встречалась (обе вершины выделены жирным контуром), из неё можно получить два листа с интересующим нас результатом — числом 3. Поэтому можно не продолжать построение от второй вершины 16, а сразу записать, что будет получено два листа (рис. 4).

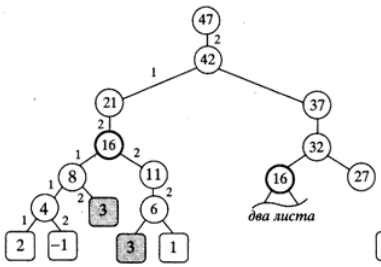


Рис. 4

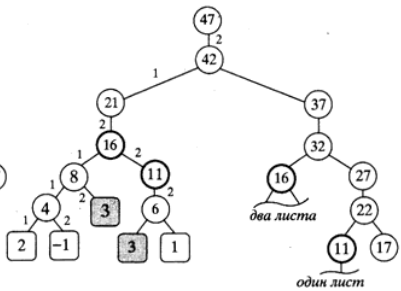


Рис. 5

Аналогично не продолжаем построения из вершин 11 (рис. 5) и 6 (рис. 6), которые уже встречались.

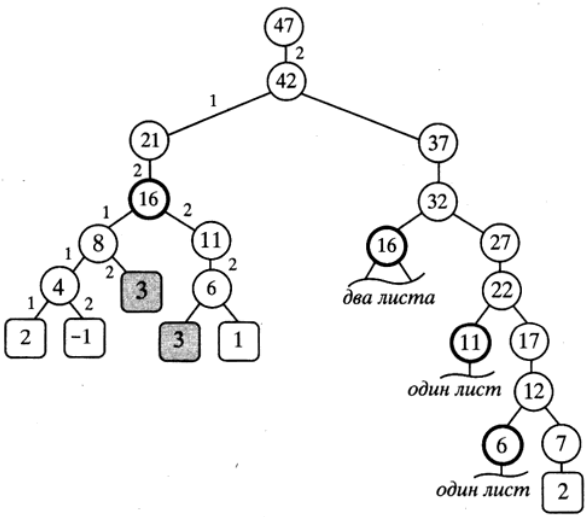


Рис. 6

Для получения результата следует подсчитать количество листьев со значением 3: $2 + 2 + 1 + 1 = 6$.

Способ 2. Запишем рекуррентные соотношения. Введём функцию $F(n)$, значением которой является количество программ Калькулятора, преобразующих заданное число 3 к аргументу функции n. Следует определить, каким образом можно вычислить значение функции по её предыдущим значениям, т. е. составить рекуррентные соотношения. Кроме того, для рекуррентных соотношений необходимо задать некоторые начальные значения функции.

Шаг 1. Определим начальные значения функции $F(n)$. После выполнения первой команды программы получим либо $3 \cdot 2 = 6$, либо $3 + 5 = 8$. Ни 6, ни 8 невозможно получить другой программой (последовательностью команд). Тогда

$F(6) = F(8) = 1$.

Обе команды увеличивают обрабатываемое число, следовательно, невозможно получить из числа 3 числа меньше 6 и число 7. Можно записать:

$F(n) = 0$, при $n < 6$, $n = 7$.

Шаг 2. Рекуррентная формула для нечётных аргументов. Нечётные числа $n = 2k + 1$ можно получить только после выполнения команды «прибавь 5». Эта команда применяется к числу $(n - 5)$: $2k + 1 - 5 = 2k - 4$. Для нечётных аргументов функции можно записать:

$F(2k + 1) = F(2k - 4)$, при $k \geqslant 4$.

Шаг 3. Рекуррентная формула для чётных аргументов. Чётные числа $n = 2k$ можно получить после выполнения любой из двух команд. При этом команда «умножь на 2» применяется к предшествующему числу k, а команда «прибавь 5» — к предшествующему числу $2k - 5$. Для чётных чисел можно записать:

$F(2k) = F(2k - 5) + F(k)$, при $k > 4$.

Шаг 4. Вычисления будем проводить по рекуррентным формулам:

$F(2k + 1) = F(2k - 4)$, при $k \geqslant 4$;
$F(2k) = F(2k - 5) + F(k)$, при $k > 4$;
$F(6) = 1$; $F(8) = 1$;
$F(k) = 0$, при $k < 6$, $k = 7$.

Для записанных соотношений:
$F(9) = F(4) = 0$,
$F(10) = F(5) + F(5) = 0$,
$F(11) = F(6) = 1$, и т. д.

Шаг 5. Вычислим по рекуррентным формулам количество программ получения числа 47 из числа 3:

$$F(47) = F(42) = F(37) + F(21) =$$
$$= F(32) + F(21) = F(27) + F(16) + F(16) =$$
$$= F(22) + 2 \cdot F(16) = F(17) + 2 \cdot F(16) + F(11) =$$
$$= 2 \cdot F(16) + F(12) + F(11) =$$
$$= F(12) + 3 \cdot F(11)\ 2 \cdot F(8) =$$
$$= 2 + F(7) + 4 \cdot F(6) = 2 + 4 = 6.$$

Ответ: 6.

C4 В международном конкурсе певцов принимают участие не более 14 представителей разных стран, по одному из каждой страны. Победитель определяется при помощи Интернет-голосования телезрителей и радиослушателей, которые указывают название страны наиболее понравившегося им певца. Вам предлагается написать эффективную, в том числе по используемой памяти, программу, которая будет статистически обрабатывать приходящие на сервер голоса, чтобы определить победителя. Следует учитывать, что количество голосов в списке может быть очень велико.

Перед текстом программы кратко опишите используемый вами алгоритм решения задачи.

На вход программе в первой строке подаётся количество пришедших голосов **N**. В каждой из последующих **N** строк записано название страны в виде текстовой строки. Длина строки не превосходит 100 символов, название может содержать буквы, пробелы, знаки препинания.

Пример входных данных:

```
5
Кот-д'Ивуар
Канада
Кот-д'Ивуар
Уругвай
Российская Федерация
```

Программа должна вывести названия всех стран, встречающихся в списке, в порядке убывания (не возрастания) количества голосов, отданных за ту или иную страну, с указанием количества отданных за неё голосов. При этом каждая страна должна быть выведена ровно один раз, вне зависимости от того, сколько за неё было подано голосов.

Пример выходных данных для приведённого выше примера входных данных:

```
Кот-д'Ивуар 2
Канада 1
Уругвай 1
Российская Федерация 1
```

Решение. В первую очередь необходимо определить основные типы данных, которые будут использоваться в программе.

Данные о голосовании за участников конкурса могут храниться в виде таблицы. Например

Номер элемента	Данные	
1	Кот-д'Ивуар	2
2	Канада	1
3	Уругвай	1
4	Российская Федерация	1

Таблица в программах представляется типом данных «массив». Здесь должен использоваться одномерный массив, состоящий из 14 элементов, по количеству стран-участниц.

Элементы массива — группа из двух значений разных типов: строкового для названия страны и целочисленного для подсчёта количества голосов. Такие данные в программах описываются структурированным типом «запись». Здесь запись `cnt` состоит из двух полей: строкового (`name`) длиной 100 символов и целого (`count`).

Описание алгоритма:

1. Программа считывает в переменную `N` количество голосов, которые ей предстоит обработать.
2. В переменную `M` записывается 0 — текущее количество стран, за которые отданы голоса, в массиве `country`.

3. В цикле от 1 до N обрабатываются строки входных данных.
   3.1. В переменную строкового типа s считывается очередная строка входных данных.
   3.2. Во внутреннем цикле просматриваются элементы массива до тех пор, пока не встретится такая же строка в поле name и переменная цикла j меньше или равна M.
      3.2.1. Если обнаружена такая же строка в поле name, то значение поля count соответствующего элемента массива увеличивается на 1 и выполнение внутреннего цикла прерывается.
      3.2.2. Если переменная цикла j стала больше M, значение M увеличивается на 1, в поле name M-го элемента массива записывается строка, в поле count записывается 1.
4. Проводится сортировка полученных данных в порядке убывания значения поля count по алгоритму сортировки «пузырьком».
5. Данные выводятся на экран. Решение — программа на языке Турбо Паскаль 7.0.

```
{ запись о стране - участнике конкурса и набранных
голосах }
type cnt=record
       name:string[100];        { название страны }
       count:integer;           { число голосов }
      end;
var
   N,i,j,M:integer;
   country:array[1..14]of cnt;  { массив элементов
типа cnt }
  t:cnt;                { вспомогательная переменная }
   s:string[100];          { вспомогательная строка }
begin
     for i:=1 to 14 do begin
          country[i].name:='';
          country[i].count:=0;
     end;
     readln(N);            { считываем число голосов }
     M:=0;
      { цикл чтения строк с исходными данными }
     for i:=1 to N do begin
         readln(s);
         j:=1;
```

```
{ в цикле ищем страну в уже существующих элементах
массива}
         while((j<=M)and(country[j].name<>s))do
            inc(j,1);
         if(j>M)then begin
         {если страны нет в массиве - добавляем её}
            inc(M,1);
            country[M].name:=s;
            country[M].count:=1;
         end
         else
{ иначе увеличиваем текущее количество голосов }
            inc(country[j].count,1);
     end;                          {конец цикла for}
{ сортируем записи в порядке убывания }
     for i:=1 to M do
         for j:=1 to M-i do
           if(country[j+1].count>country[j].count) then
begin
                t:=country[j];
                country[j]:=country[j+1];
                country[j+1]:=t;
             end;
      { выводим записи на экран}
     for i:=1 to M do
      writeln(country[i].name,' ',country[i].count);
end.
```

Вариант 2

Часть 1

При выполнении заданий части 1 в бланке ответов № 1 под номером выполняемого вами задания (А1–А14) поставьте знак «×» в клеточке, номер которой соответствует номеру выбранного вами ответа.

А1 Сколько значащих нулей в двоичной записи числа 1537?

1) 7 2) 8 3) 9 4) 6

См. учебно-справочные материалы: Перевод числа из одной позиционной системы счисления в другую, с. 44.

А2 Между населёнными пунктами A, B, C, D, E, F построены дороги, протяжённость которых приведена в таблице. (Отсутствие числа в таблице означает, что прямой дороги между пунктами нет.)

	A	B	C	D	E	F
A		2		6	17	
B	2		4	3		18
C		4			7	
D	6	3			10	
E	17		7	10		6
F		18			6	

Определите длину кратчайшего пути между пунктами A и F (при условии, что передвигаться можно только по построенным дорогам).

1) 17 2) 18 3) 19 4) 20

См. учебно-справочные материалы: Элементы теории графов, с. 124.

A3 Дан фрагмент таблицы истинности выражения **F**:

X	Y	Z	F
0	1	0	**1**
1	0	0	**1**
1	1	1	**0**

Каким может быть **F**?

1) $\neg X \vee \neg Y \vee \neg Z$
2) $\neg X \wedge \neg Y \wedge \neg Z$
3) $X \vee Y \vee \neg Z$
4) $\neg X \wedge Y \wedge Z$

См. учебно-справочные материалы: Логические выражения, с. 89. Построение таблиц истинности, с. 90.

A4 Для групповых операций с файлами используются **маски имён файлов**. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы:

— символ «?» (вопросительный знак) означает ровно один произвольный символ;

— символ «*» (звёздочка) означает любую последовательность символов произвольной длины, в том числе символ «*» может задавать и пустую последовательность.

Определите, по какой из масок будет выбрана указанная группа файлов:

```
aabbbaa.pas
dbbaa.pas
aaaaa.cs
daabda.cs
```

1) `?aa*.****?`
2) `*ba.??*`
3) `*a*.????`
4)`*a*.?*?`

См. учебно-справочные материалы: Поиск файлов, с. 36.

A5 Автомат получает на вход два трёхзначных числа. По этим числам строится новое число по следующим правилам:

1. Записывается результат сложения значений старших разрядов заданных чисел.

2. К нему дописывается результат сложения значений средних разрядов этих чисел по такому правилу: если он меньше первой суммы, то второе полученное число приписывается к первому слева, иначе — справа.

3. Итоговое число получают приписыванием справа к полученному после второго шага числу суммы значений младших разрядов исходных чисел.

Пример. Исходные трёхзначные числа: 328, 207. Поразрядные суммы: 5, 2, 15. Результат: 2515.

Определите, какое из предложенных чисел может быть результатом такой операции.

1) 15117 2) 19213 3) 181717 4) 11146

См. учебно-справочные материалы: Основы алгоритмизации, с. 161.

А6 В фрагменте базы данных представлены сведения о родственных отношениях. Определите на основании приведённых данных фамилию и инициалы бабушки Ивановой А.И.

Таблица 1

ID	Фамилия_И.О.	Пол
10	Логинов Н.А.	М
13	Черных И.А.	Ж
23	Иванова А.И.	Ж
42	Петренко А.И.	Ж
71	Иванов Т.М.	М
82	Черных А.Н.	М
85	Петренко И.Т.	М
96	Петренко Н.Н.	Ж
95	Логинова Т.Н.	Ж
	...	

Таблица 2

ID_Родителя	ID_Ребёнка
23	71
13	23
85	23
82	13
95	13
85	42
82	10
95	10
...	...

1) Иванов Т.М.
2) Черных И.А.
3) Логинова Т.Н.
4) Петренко Н.Н.

См. учебно-справочные материалы: Реляционная база данных, с. 244.

A7 В динамической (электронной) таблице приведены количества автомобилей, побывавших в сервис-центре, и количества иномарок из них. Укажите автомеханика, у которого доля обслуженных иномарок по отношению к общему количеству обслуженных им автомобилей, максимальна.

Механик	Количество а/м	Вид работы				Итого
		смена масла	диагностика	полировка	установка акустики	
Иванов	всего	20	10	5	7	42
	из них иномарок	10	2	1	5	18

Продолжение таблицы

Механик	Количество а/м	Вид работы				Итого
		смена масла	диагностика	полировка	установка акустики	
Петров	всего	14	12	8	5	39
	из них иномарок	8	7	2	2	19
Сидоров	всего	10	10	4	16	40
	из них иномарок	9	4	2	4	19
Прохоров	всего	15	13	8	12	48
	из них иномарок	8	6	3	6	23

1) Иванов
2) Петров
3) Сидоров
4) Прохоров

А8 Производится одноканальная (моно) звукозапись с частотой дискретизации 64 Гц. При записи использовались 32 уровня дискретизации. Запись длится 4 минуты 16 секунд, её результаты записываются в файл, причём каждый сигнал кодируется минимально возможным и одинаковым количеством битов. Какое из приведённых ниже чисел наиболее близко к размеру полученного файла, выраженному в килобайтах?

1) 10 2) 64 3) 80 4) 512

См. учебно-справочные материалы: Кодирование звуковой информации, с. 70.

А9 Для кодирования некоторой последовательности, состоящей из букв **А**, **Б**, **В**, **Г** и **Д**, решили использовать неравномерный двоичный код, позволяющий однозначно декодировать двоичную последовательность, появляющуюся на приёмной стороне канала связи. Использовали код: А – 11, Б – 100, В – 101, Г – 010. Укажите, каким кодовым словом должна быть закодирована буква **Д**.

Длина этого кодового слова должна быть наименьшей из всех возможных. Код должен удовлетворять свойству однозначного декодирования.

1) 00
2) 0101
3) 011
4) 10

См. учебно-справочные материалы: Префиксные коды, с. 8.

A10 Какое из приведённых названий садового инвентаря удовлетворяет логическому условию:

¬ ((первая буква согласная ∧ последняя буква согласная) → больше 6 букв)?

1) ЛОПАТА 2) СЕКАТОР 3) ГРАБЛИ 4) ШЛАНГ

См. учебно-справочные материалы: Основные законы алгебры логики, с. 92.

A11 Для регистрации на сайте пользователю требуется придумать пароль. Длина пароля — ровно 15 символов. В качестве символов используются десятичные цифры и 28 различных букв, причём все буквы используются в двух начертаниях — как строчные, так и заглавные (регистр буквы имеет значение!).

Под хранение каждого такого пароля на компьютере отводится минимально возможное и одинаковое целое количество байтов, при этом используется посимвольное кодирование и все символы кодируются одинаковым и минимально возможным количеством битов. Определите объём памяти, который требуется для хранения 20 паролей.

1) 240 байт 2) 260 байт 3) 263 байт 4) 280 байт

См. учебно-справочные материалы: Измерение информации, с. 11.

A12 В программе используется одномерный целочисленный массив **A** с индексами от 0 до 10 и целочисленные переменные **tmp** и **i**. Ниже представлен фрагмент программы, за-

писанный на разных языках программирования, в котором значения элементов сначала задаются, а затем меняются.

Бейсик	Паскаль

`FOR i=0 TO 10`
`  A(i)=i`
`NEXT i`
`FOR i=0 TO 8 STEP 2`
`  TMP = A(i)`
`  A(i) = A(i+2)`
`  A(i+2) = TMP`
`NEXT i`
	`for i:=0 to 10 do`
	`  A[i]:=i;`
	`for i:=0 to 8 do`
	`  if ( i mod 2 )=0`
	`  then`
	`  begin`
	`  tmp := A[i];`
	`  A[i]:=A[i+2];`
	`  A[i+2]:=tmp;`
	`  end;`

Си	**Алгоритмический язык**

`for (i=0;i<=10;i++)`
`  A[i]=i;`
`for (i=0;i<9;i+=2)`
`{`
`  tmp= A[i];`
`  A[i]=A[i+2];`
`  A[i+2]=tmp;`
`}`
	`нц для i от 0 до 10`
	`  A[i]:=i`
	`кц`
	`нц для i от 0 до 8 шаг 2`
	`  tmp := A[i]`
	`  A[i]:=A[i+2]`
	`  A[i+2]:=tmp`
	`кц`

Чему будут равны элементы этого массива после выполнения фрагмента программы?

1) 2 1 4 3 6 5 8 7 10 9 0
2) 2 1 0 3 6 5 4 7 10 9 8
3) 1 0 3 2 5 4 7 6 9 8 10
4) 1 2 3 4 5 6 7 8 9 10 0

См. учебно-справочные материалы: Одномерные массивы, с. 218.

A13 Система команд исполнителя РОБОТ, «живущего» в прямоугольном лабиринте на клетчатой плоскости:

вверх	вниз	влево	вправо

При выполнении любой из этих команд РОБОТ перемещается на одну клетку соответственно: вверх ↑, вниз ↓, влево ←, вправо →.

Четыре команды проверяют истинность условия отсутствия стены у каждой стороны той клетки, где находится РОБОТ:

сверху свободно	**снизу свободно**	**слева свободно**	**справа свободно**

Цикл ПОКА <*условие*> *команда* выполняется, пока условие истинно, иначе происходит переход на следующую строку.

Если РОБОТ начнёт движение в сторону стены, то он разрушится и программа прервётся.

Сколько клеток лабиринта соответствует требованию, что, выполнив предложенную программу, РОБОТ уцелеет и остановится в той же клетке, с которой он начал движение?

НАЧАЛО

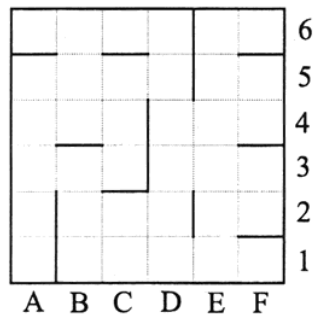
ПОКА **<справа свободно> вправо**

ПОКА **<снизу свободно> вниз**

ПОКА **<слева свободно> влево**

ПОКА **<сверху свободно> вверх**

КОНЕЦ



1) 1 2) 2 3) 3 4) 4

См. учебно-справочные материалы: Формальный исполнитель РОБОТ, с. 179.

A14 Определите, какое число будет напечатано в результате работы следущей программы (для вашего удобства программа представлена на четырёх языках):

Бейсик	Паскаль

`Module A14`
`Sub Main()`
`Dim d, a, b, t, M, R As Double`
`a = -2 : b = 2`
	`Program A14;`
	`Uses crt;`
	`Var d,a,b,t,M,R :real;`
	`Function F(x : real):real;`

Бейсик

```
        d = 0.1
        t = a: M = a: R = F(a)
          While t <= b
          If F(t) > R Then
                  M = t
                  R = F(t)
             End If
             t = t + d
          End While
         Console.Write(M)
 End Sub
 Function F(ByVal x As
Double)
As Double
     Return 4*x*(1-x)
 End Function
End Module
```

Паскаль

```
  begin
    F:= 4*x*(1-x);
  end;
BEGIN
a:=-2; b:=2;
d:=0.1;
t:=a; M:=a; R:=F(a);
while t<=b do
  begin
    if (F(t)> R) then
       begin
         M:=t;
         R:=F(t);
       end;
    t:=t+d;
  end;
write(M);
END.
```

Си

```
#include <stdio.h>
 double F(double x)
 {
   return 4*x*(1-x);
 }
 void main()
 {
  double d, a, b, t, M, R;
   a = -2; b = 2;
   d = 0.1;
   t = a; M = a; R = F(a);
   while (t<=b) {
     if ( F(t)> R ) {
       M = t; R = F(t);
     }
     t = t + d;
   }
   printf("%.1f", M);
 }
```

Алгоритмический язык

```
алг А14
нач
   вещ d, a, b, t, M, R
    a:= -2; b:= 2
    d:= 0.1
  t:= a; M:= a; R:= F(a)
    нц пока t<=b
      если F(t)> R то
        M:= t; R:= F(t)
      все
      t:= t + d
    кц
    вывод M
кон
алг вещ F(вещ x)
нач
    знач := 4*x*(1-x)
кон
```

1) –2 2) 0,5 3) 1 4) 2

См. учебно-справочные материалы: Подпрограммы, определяемые разработчиком, с. 232.

Часть 2

Ответом к заданиям части 2 (B1–B14) является число, последовательность букв или цифр. Впишите ответы сначала в текст работы, а затем перенесите их в бланк ответов № 1 справа от номера соответствующего задания, начиная с первой клеточки, без пробелов, запятых и других дополнительных символов. Каждую букву или цифру пишите в отдельной клеточке в соответствии с приведёнными в бланке образцами.

B1 Автоматическое устройство осуществило перекодировку информационного сообщения на русском языке длиной 80 символов, первоначально записанного в 8-битной кодировке КОИ-8, в 16-битный код Unicode. На сколько битов увеличилась длина сообщения? *В ответе запишите только число.*

Ответ: _____.

См. учебно-справочные материалы: Кодирование текстовой информации, с. 60.

B2 У исполнителя Вычислитель две команды, которым присвоены номера:

```
1. умножить на -2
2. вычесть 7
```

Первая из них умножает число на экране на –2, вторая вычитает из него 7.

Запишите порядок команд в алгоритме получения из числа 7 числа 84, содержащем не более 5 команд, указывая только номера команд (например, 2122 — это алгоритм:

```
2. вычесть 7
1. умножить на -2
2. вычесть 7
2. вычесть 7
```

который преобразует число 5 в число –10).

Если таких алгоритмов более одного, запишите любой из них.

Ответ: _____.

См. учебно-справочные материалы: Формальный исполнитель Вычислитель, с. 185.

B3 Определите, что будет выведено в результате работы следующего фрагмента программы:

Бейсик	Паскаль

`Module Module1`
`  Sub Main()`
`  Dim a, b As Integer`
`  a = 56`
`  b = a - 70`
`  While Not b > 0`
`  b = b + 13`
`  End While`
`  While a Mod b <> 0`
`  a = a - 4`
`  End While`
`  Console.Write(a)`
`  End Sub`
`End Module`
	`var a,b: integer;`
	`BEGIN`
	`  a:=56;`
	`  b:=a-70;`
	`  while not(b>0) do`
	`  b:=b+13;`
	`  while a mod b <> 0 do`
	`  a:=a-4;`
	`  write(a);`
	`END.`

Си	**Алгоритмический язык**

`{`
`  int a,b;`
`  a = 56;`
`  b = a - 70;`
`  while !(b>0)`
`  b = b+13;`
`  while (a % b != 0)`
`  a = a - 4;`
`  printf("%d", a);`
`}`
	`нач`
	`  цел a,b`
	`  a:=56`
	`  b:=a-70`
	`  нц пока не (b>0)`
	`  b:=b+13`
	`  кц`
	`  нц пока mod(a,b)<>0`
	`  a:=a-4`
	`  кц`
	`  вывод a`
	`кон`

Ответ: _____.

См. учебно-справочные материалы: Алгоритмы, заданные на алгоритмическом языке, с. 193.

B4 Все пятибуквенные слова, составленные из букв А, О, У, записаны в алфавитном порядке. Вот начало списка:

1. ААААА
2. ААААО
3. ААААУ
4. ААААОА

.....

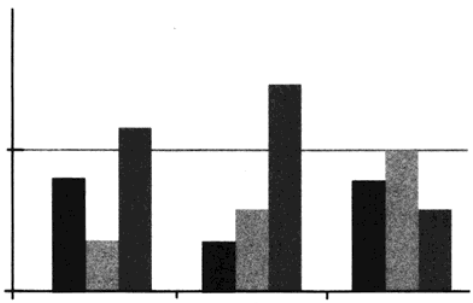
Запишите слово, которое стоит на **240-м месте** от начала списка.

Ответ: _____.

См. учебно-справочные материалы: Позиционные системы счисления, с. 40.

B5 Дан фрагмент электронной таблицы:

	A	B	C
1	4	2	4
2	2	3	
3	=A2+A1	=B2+C2	=(C1+C2–B2)/2



Какое число должно быть записано в ячейке C2, чтобы построенная по значениям диапазона ячеек A1:D3 диаграмма соответствовала рисунку.

Ответ: _____.

B6 Определите значение переменной *c* после выполнения следующего фрагмента программы, в котором *a*, *b* и *c* — переменные вещественного (действительного) типа.

Бейсик	Паскаль

a = 35
b = a - a mod 21
IF a - b >= 14
THEN
  c = 2 * a - b
ELSE
  c = 2 * b - a
ENDIF
	a:= 35;
	b:= a - a mod 21;
	if a - b >= 14
	then
	  c := 2 * a - b
	else
	  c := 2 * b - a;

Си	**Алгоритмический язык**

a = 35;
b = a - a % 21;
if a - b >= 14
  c = 2*a - b;
else
  c = 2*b - a;
	a := 35
	b := a – mod(a,21)
	если (a – b >= 14)
	то c := 2 * a – b
	иначе c := 2 * b – a
	все

Ответ: _____.

См. учебно-справочные материалы: Алгоритмы, заданные на алгоритмическом языке, с. 193.

B7 Ниже приведены тексты одной и той же программы, записанные на четырёх языках программирования. Что будет напечатано в результате выполнения этой программы?

Алгоритмический язык

```
алг Задача
нач
      цел L=5; целтаб R[1:L]; цел N, p
      R[1]:=2;R[2]:=2;R[3]:=1;R[4]:=3;R[5]:=2
      ArrProcessing_1(L,R,p)
      если p>5 то
            вывод "Переполнение"; стоп
      все
      N:=MultCounter(L,R)
      вывод N
кон
```

Продолжение таблицы

```
алг ArrProcessing_1(цел L, аргрез целтаб R[1:L],
рез цел p)
нач
      цел i,t
      p:=1
      нц для i от 1 до L - 1
            R[i+1]:=R[i]*p
            p:=i
      кц
кон
алг цел MultCounter(цел L, аргрез целтаб R[1:L])
нач
      цел N, i, T
      N:=0
      T:=1
      нц для i от 2 до L
            T:=R[i-1]
            N:=N+T*R[i]
      кц
      знач:=N
кон
```

Бейсик

```
Module Task
    Sub Main()
        Dim L As Integer = 5
        Dim R(5) As Integer
        Dim N, p As Integer
        R.SetValue(2, 1)
        R.SetValue(2, 2)
        R.SetValue(1, 3)
        R.SetValue(3, 4)
        R.SetValue(2, 5)
        ArrProcessing_1(L, R, p)
        If p > 5 Then
            Console.Write("Переполнение")
            Stop
        End If
        N = MultCounter(L, R)
        Console.Write(N)
    End Sub
```

Продолжение таблицы

```
    Sub ArrProcessing_1(ByVal L As Integer,
ByRef R As Array, ByRef p As Integer)
        Dim i, t As Integer
        p = 1
        For i = 1 To L - 1
            R.SetValue(R.GetValue(i) * p, i + 1)
            p = i
        Next
    End Sub
    Function MultCounter(ByVal L As Integer,
ByRef R As Array) As Integer
        Dim N, i, T As Integer
        N = 0
        T = 1
        For i = 2 To L
            T = R.GetValue(i - 1)
            N = N + T * R.GetValue(i)
        Next
        Return N
    End Function
End Module
```

Паскаль

```
Program Task;
Uses crt;
const L = 5;
type
    atype = array [1..L] of integer;
Var R: atype;
    N,p: integer;
Procedure  ArrProcessing_1(L:  integer;  var  R:
atype; var p:integer);
          var i,t: integer;
begin
     p:=1;
     for i:=1 to L-1 do
     begin
          R[i+1]:=R[i]*p;
          p:=i;
     end;
end;
```

Продолжение таблицы

```
Function MultCounter(L: integer; var R:
atype):integer;
        var N,i,T: integer;
begin
     N:=0;
     T:=1;
     for i:=2 to L do
     begin
          T:=R[i-1];
          N:=N+T*R[i];
     end;
     MultCounter:=N;
end;
BEGIN
     clrscr;
     R[1]:=2;R[2]:=2;R[3]:=1;R[4]:=3;R[5]:=2;
     ArrProcessing_1(L,R,p);
     if p>5 then
     begin
          write('Переполнение');
          halt;
     end;
     N:=MultCounter(L,R);
     write(N);
END.
```

Си

```
#include <stdio.h>
#include <stdlib.h>

void ArrProcessing_1(int L, int* R, int* p)
{
     int i, t;
     *p = 1;
     for(i=0; i<L-1; i++)
     {
               R[i+1]= R[i] * *p;
               *p = i + 1;
     }
 }
```

Продолжение таблицы

```
 int MultCounter(int L, int* R)
 {
     int N, i, T;
     N = 0;
     T = 1;
     for (i=1; i<L; i++)
     {
         T=R[i-1];
         N = N + T*R[i];
     }
     return N;
}

void main()
{
    int L = 5;
    int* R = (int*) calloc(L, sizeof(int));
    int N, p;
   R[0] = 2; R[1] = 2; R[2] = 1; R[3] = 3; R[4] = 2;
    ArrProcessing_1(L, R, &p);
    if (p>5)
    {
            printf("Переполнение");
            free(R);
            return;
    }
    N = MultCounter(L,R);
    printf("%d\n", N);
    free(R);
    return;
}
```

Ответ: _____.

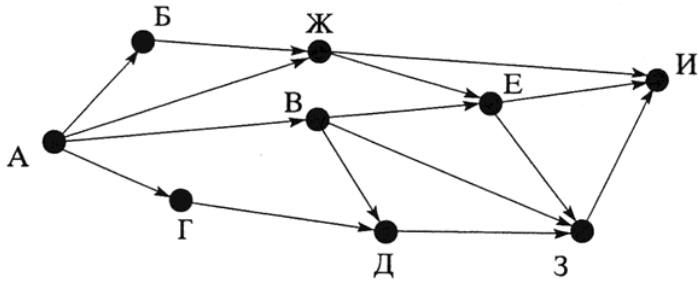
См. учебно-справочные материалы: Подпрограммы, определяемые разработчиком, с. 232.

B8 Запись числа 45_{10} в системе счисления с основанием N оканчивается на 0 и содержит 4 цифры. Чему равно основание этой системы счисления N?

Ответ: _____.

См. учебно-справочные материалы: Позиционные системы счисления, с. 40.

B9 На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город И?



Ответ: _____.

B10 У Васи есть доступ в Интернет по высокоскоростному одностороннему радиоканалу, обеспечивающему скорость получения им информации 2^{10} бит/с. У Миши нет скоростного доступа в Интернет, но есть возможность получать информацию от Васи по низкоскоростному телефонному каналу со средней скоростью 2^8 бит/с. Миша договорился с Васей, что тот будет скачивать для него данные объёмом 1 Мбайт по высокоскоростному каналу и ретранслировать их Мише по низкоскоростному каналу.

Компьютер Васи может начать ретрансляцию данных не раньше, чем им будут получены первые 256 Кбайт этих данных. Каков минимально возможный промежуток времени (в секундах) с момента начала скачивания Васей данных до полного их получения Мишей?

В ответе укажите только число, слово «секунд» или букву «с» добавлять не нужно.

Ответ: _____.

См. учебно-справочные материалы: Передача информации, с. 20.

B11 В терминологии сетей TCP/IP **маской сети** называется двоичное число, определяющее, какая часть IP-адреса узла

сети относится к адресу сети, а какая — к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес. Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске.

По заданным IP-адресу узла и маске определите адрес сети.
IP-адрес узла: 33.153.102.21
Маска: 255.255.248.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса и запишите в нужном порядке соответствующие им буквы, без использования точек.

A	B	C	D	E	F	G	H
102	232	33	96	153	21	0	248

Пример.

Пусть искомый IP-адрес сети — 192.168.120.0 и дана таблица

A	B	C	D	E	F	G	H
255	*11*	*166*	*120*	*127*	*0*	*192*	*168*

В этом случае правильный ответ будет записан в виде GHDF.

Ответ: _____.

См. учебно-справочные материалы: Подсети. Маски подсетей, с. 270.

B12 В языке запросов поискового сервера для обозначения логической операции «ИЛИ» используется символ «|», для логической операции «И» — символ «&», для логической операции «НЕ» — символ «-».

В таблице приведены запросы и количество найденных по ним страниц некоторого сегмента сети Интернет:

Запрос	Найдено страниц (в тысячах)
Кино & Театр	2560
Кино & Цирк	2310
Театр & Цирк	1590
Кино & Театр & Цирк	670
Кино | Театр | Цирк	8600

Какое количество страниц (в тысячах) будет найдено по запросу
((Кино & Театр) | (Кино & Цирк) | (Театр & Цирк)) & -(Кино & Театр & Цирк)?

Считается, что все запросы выполнялись практически одновременно, так что набор страниц, содержащих все искомые слова, не изменялся за время выполнения запросов.

Ответ: _____.

См. учебно-справочные материалы: Элементы теории множеств, с. 132.

B13 У исполнителя Вычислитель две команды:

```
1. прибавь 1
2. вычти 2
```

Первая из них увеличивает число на экране на 1, вторая уменьшает его на 2 (отрицательные числа допускаются).

Программа для Вычислителя — это последовательность команд. Сколько различных чисел можно получить из числа 5 с помощью программы, которая содержит ровно 12 команд?

Ответ: _____.

B14 Сколько различных решений имеет система уравнений

$$a \to b \vee c \wedge \neg d = 1$$
$$c \to d \vee e \wedge \neg f = 1$$
$$e \to f \vee g \wedge \neg h = 1$$
$$g \to h \vee i \wedge \neg j = 1$$

где $a, b, c, d, e, f, g, h, i, j$ — логические переменные?

В ответе запишите количество различных наборов значений $a, b, c, d, e, f, g, h, i, j$, при которых выполняется данная система равенств.

Ответ: _____.

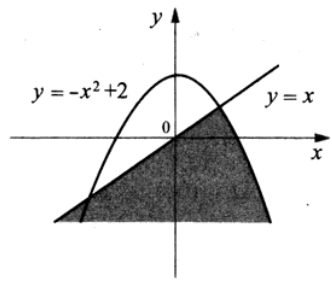
См. учебно-справочные материалы: Построение СДНФ логической функции, с. 102.

Не забудьте перенести все ответы в бланк ответов № 1.

Часть 3

Для записи ответов к заданиям части 3 (С1–С4) используйте бланк ответов № 2. Запишите сначала номер задания (С1, С2 и т.д.), а затем полное решение. Ответы записывайте чётко и разборчиво.

С1 Требовалось написать программу, при выполнении которой с клавиатуры считываются координаты точки на плоскости (x, y — действительные числа) и определяется принадлежность этой точки заданной закрашенной области (включая границы), неограниченной снизу. Программист торопился и написал программу неправильно.



Паскаль	Бейсик	Си

var x, y: real;
begin
readln(x, y);
if y <= x then
if y <= -x*x + 2
then
write
('принадлежит')
else
write
('не принадлежит')
end.
	INPUT x,y
	IF y <= x THEN
	IF
	y<=-x*x+2 THEN
	PRINT
	"принадлежит"
	ELSE
	PRINT
	"не принадлежит"
	ENDIF
	ENDIF
	END
		void main(void)
		{float x,y;
		scanf
		("%f%f",&x,&y);
		if (y <= x)
		if (y<=-x*x+2)
		printf
		("принадлежит");
		else
		printf
		("не принадлежит");
		}

Последовательно выполните следующее:

1) Приведите пример таких чисел x, y, при которых программа неверно решает поставленную задачу. Объясните, почему для указанных чисел программа неверно решает поставленную задачу.
2) Укажите, как нужно доработать программу, чтобы не было случаев её неправильной работы. (Это можно сделать не-

сколькими способами, достаточно указать любой способ доработки исходной программы.)

C2 Дан массив из 30 целых чисел. Элементы массива имеют значение от –100 до 100 включительно. Необходимо разработать программу, которая меняет местами элементы с минимальным и максимальным значениями. Гарантируется, что среди элементов массива нет элементов с одинаковым значением. Массив после перестановки элементов вывести на экран.

Фрагмент программы с объявлением переменных и заполнением массива приведён ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из описанных переменных.

Паскаль	Бейсик

`const n = 30;`
`var a:array[1..n] of integer;`
`  i, max, nmax:integer;`
`  min, nmin:integer;`
`begin`
`for i:=1 to n do`
`readln(a[i]);`
`  .....`
`end.`
	`N = 30`
	`DIM A(N) AS INTEGER`
	`DIM I, MAX, NMAX AS INTEGER`
	`DIM MIN, NMIN AS INTEGER`
	`FOR I = 0 TO N`
	`  INPUT A(I)`
	`NEXT I`
	`.....`
	`END`

Си	**Естественный язык**

`#include <stdio.h>`
`#define N  30`
`void main()`
`{`
`  int a[N], i, max,`
`  nmax, min, nmin;`
`  for(i=0; i<N; i++)`
`  scanf("%d",&a[i]);`
`  .....`
`}`
	Объявляем массив A из 30 элементов.
	Объявляем целочисленные переменные
	I, MAX, NMAX, MIN, NMIN.
	В цикле от 1 до 30 вводим элементы
	массива A с 1-го по 30-й.
	.....

В качестве ответа вам необходимо привести фрагмент программы (или описание алгоритма на естественном языке), который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например Borland Pascal 7.0) или в виде блок-схемы. В этом случае вы должны использовать переменные, аналогичные переменным, используемым в алгоритме, записанном на естественном языке, с учётом синтаксиса и особенностей используемого вами языка программирования.

C3 У исполнителя Утроитель две команды, которым присвоены номера:

```
1. прибавь 2
2. умножь на 3
```

Первая команда увеличивает число на 2, вторая — умножает на 3.

Программа Утроителя — это последовательность команд.

Сколько есть программ, которые число 3 преобразуют в число 35?

C4 Во всероссийском конкурсе театров принимают участие не более 12 театров из разных городов, по одному из каждого города. Победитель определяется при помощи терминалов электронного голосования для зрителей, которые указывают название города, где находится наиболее понравившийся им театр. Вам предлагается написать эффективную, в том числе по используемой памяти, программу, которая будет статистически обрабатывать приходящие с терминалов на сервер запросы, с целью определения победителя. Следует учитывать, что количество голосов в списке может быть очень велико.

Перед текстом программы кратко опишите используемый вами алгоритм решения задачи.

На вход программе в первой строке подаётся количество пришедших запросов **N**. В каждой из последующих **N** строк записано название города в виде текстовой строки. Длина строки не превосходит 100 символов, название может содержать буквы, пробелы, знаки препинания.

Пример входных данных:

```
4
Петропавловск-Камчатский
Новосибирск
Сургут
Новосибирск
```

Программа должна вывести названия всех городов, встречающихся в списке, в порядке возрастания (не убывания) количества голосов, отданных за тот или иной театр, с указанием количества отданных за него голосов. При этом каждый город должен быть выведен ровно один раз, вне зависимости от того, сколько за него было подано голосов.

Пример выходных данных для приведённого выше примера входных данных:

```
Петропавловск-Камчатский 1
Сургут 1
Новосибирск 2
```

Вариант 3

Часть 1

При выполнении заданий части 1 в бланке ответов № 1 под номером выполняемого вами задания (А1–А14) поставьте знак «×» в клеточке, номер которой соответствует номеру выбранного вами ответа.

А1 Сколько значащих нулей в двоичной записи числа 1088?

1) 7 2) 8 3) 9 4) 10

См. учебно-справочные материалы: Перевод числа из одной позиционной системы счисления в другую, с. 44.

А2 Между населёнными пунктами A, B, C, D, E, F построены дороги, протяжённость которых приведена в таблице. (Отсутствие числа в таблице означает, что прямой дороги между пунктами нет.)

	A	B	C	D	E	F
A		3		2	4	
B	3		8		3	15
C		8		8		7
D	2		8		3	
E	4	3		3		15
F		15	7		15	

Определите длину кратчайшего пути между пунктами A и F (при условии, что передвигаться можно только по построенным дорогам).

1) 16 2) 17 3) 18 4) 19

См. учебно-справочные материалы: Элементы теории графов, с. 124.

A3 Дан фрагмент таблицы истинности выражения **F**:

X	Y	Z	F
0	1	1	**0**
1	0	1	**1**
1	1	0	**1**

Каким может быть **F**?

1) $\neg X \vee \neg Y \vee Z$
2) $X \wedge \neg Y \wedge \neg Z$
3) $\neg X \vee Y \vee \neg Z$
4) $X \vee \neg Y \vee \neg Z$

См. учебно-справочные материалы: Нормальные формы логических функций, с. 97.

A4 Для групповых операций с файлами используются **маски имён файлов**. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы:

— символ «?» (вопросительный знак) означает ровно один произвольный символ;

— символ «*» (звёздочка) означает любую последовательность символов произвольной длины, в том числе символ «*» может задавать и пустую последовательность.

Определите, по какой из масок будет выбрана указанная группа файлов:

```
ddde.xls
dded.xls
da.xls
adde.xls
```

1) `*d*.????*?`
2) `*d?*.***`
3) `*dd*.???`
4) `*de*.xls`

См. учебно-справочные материалы: Поиск файлов, с. 36.

A5 Автомат получает на вход два трёхзначных числа. По этим числам строится новое число по следующим правилам:

1. Записывается результат сложения значений старших разрядов заданных чисел.
2. К нему дописывается результат сложения значений средних разрядов этих чисел по такому правилу: если он меньше первой суммы, то второе полученное число приписывается к первому справа, иначе — слева.
3. Итоговое число получают приписыванием справа к полученному после второго шага числу суммы значений младших разрядов исходных чисел.

Пример. Исходные трёхзначные числа: 721, 195. Поразрядные суммы: 8, 11, 6. Результат: 1186.

Определите, какое из предложенных чисел может быть результатом такой операции.

1) 1112 2) 151612 3) 19118 4) 12045

См. учебно-справочные материалы: Основы алгоритмизации, с. 161.

А6 Ниже приведены связанные между собой таблицы базы данных тестирования учеников 9-х и 11-х классов.

Таблица «Ученики»

ID_ученика	Фамилия	Имя	Класс
10112	Самсонова	Татьяна	11
10128	Фёдорова	Анастасия	9
10146	Иванов	Денис	11
10157	Сидоров	Илья	11
10161	Смирнов	Пётр	9
10168	Борисов	Александр	9
10172	Петрова	Елена	11
10179	Васильев	Павел	11

Таблица «Предметы»

ID_предмета	Наименование
1	Математика
2	Русский язык
5	Английский язык
7	Биология
9	Химия
11	Физика

Таблица «Оценки»

ID_ученика	ID_предмета	Баллы
10112	7	89
10146	5	96
10112	9	59
10161	1	98
10168	5	57
10179	11	74
10128	1	77
10157	1	81

Сколько учеников 11-х классов получили больше 70 баллов при тестировании по математике или по английскому языку?

1) 1 2) 2 3) 3 4) 4

См. учебно-справочные материалы: Реляционная база данных, с. 244.

A7 В динамической (электронной) таблице приведены значения расхода плюша (в погонных метрах) и пошитых мягких игрушек (в штуках) для четырёх компаний. Какое предприятие затратило в среднем меньше плюша на игрушку?

Наименование	Название компании							
	ОАО «Мягкая игрушка»		Комбинат «Играй с нами»		Фабрика «Мосплюш»		ООО «Плюшевый друг»	
	Расход	Пошито	Расход	Пошито	Расход	Пошито	Расход	Пошито
Котята	40	15	28	5	14	3	50	25
Слоники	60	30	40	12	50	12	34	15
Мишки	18	5	50	28	23	10	12	5
Ослики	40	23	54	29	12	2	15	3
Всего	**158**	**73**	**172**	**74**	**99**	**27**	**111**	**48**

1) ОАО «Мягкая игрушка»
2) комбинат «Играй с нами»
3) фабрика «Мосплюш»
4) ООО «Плюшевый друг»

A8 Производится одноканальная (моно) звукозапись с частотой дискретизации 16 КГц. При записи использовались 256 уровней дискретизации. Запись длится 2 минуты 8 секунд, её результаты записываются в файл, причём каждый сигнал кодируется минимально возможным и одинаковым количеством битов. Какое из приведённых ниже чисел наиболее близко к размеру полученного файла, выраженному в килобайтах?

1) 15,6 2) 2000 3) 16000 4) 512000

См. учебно-справочные материалы: Кодирование звуковой информации, с. 70.

A9 Для кодирования некоторой последовательности, состоящей из букв **А**, **Б**, **В**, **Г** и **Д**, решили использовать неравномерный двоичный код, позволяющий однозначно декоди-

ровать двоичную последовательность, появляющуюся на приёмной стороне канала связи. Использовали код: А – 10, Б – 110, В – 111, Г – 000. Укажите, каким кодовым словом должна быть закодирована буква **Д**.

Длина этого кодового слова должна быть наименьшей из всех возможных. Код должен удовлетворять свойству однозначного декодирования.

1) 00 3) 01
2) 011 4) 101

См. учебно-справочные материалы: Префиксные коды, с. 8.

A10 Для какого имени истинно высказывание:

¬ (первая буква имени гласная ⊕ четвёртая буква имени согласная)?

1) ЕЛЕНА 3) КСЕНИЯ
2) ВЛАДИМИР 4) АРСЕНИЙ

Примечание: ⊕ — обозначает логическую операцию XOR (исключающее ИЛИ).

См. учебно-справочные материалы: Основные законы алгебры логики, с. 92.

A11 На аэродроме 56 самолётов. Индивидуальный код самолёта состоит из 9 символов. В качестве символов используют 26 различных букв и десятичные цифры в произвольном порядке.

Каждый такой код в компьютерной программе записывается одинаковым минимально возможным целым количеством байтов, при этом используют посимвольное кодирование. Каждый символ кодируются одинаковым минимально возможным целым количеством битов. При вызове диспетчера пилотом индивидуальный код самолёта загружается в диспетчерское устройство. Какой объём памяти диспетчерского устройства задействуется, если диспетчера одновременно вызывают 16 пилотов?

1) 96 байт 2) 106 байт 3) 112 байт 4) 896 байт

См. учебно-справочные материалы: Измерение информации, с. 11.

A12 В программе используется одномерный целочисленный массив **A** с индексами от 0 до 10 и целочисленные переменные **tmp** и **i**. Ниже представлен фрагмент программы, записанный на разных языках программирования, в котором значения элементов сначала задаются, а затем меняются.

Бейсик	Паскаль

`FOR i=0 TO 10`
`  A(i)=i`
`NEXT i`
`FOR i=0 TO 4`
`  TMP = A(i)`
`  A(i) = A(2*i+2)`
`  A(2*i+2) = TMP`
`NEXT i`
	`for i:=0 to 10 do`
	`  A[i]:=i;`
	`for i:=0 to 4 do`
	`  begin`
	`  tmp := A[i];`
	`  A[i]:=A[2*i+2];`
	`  A[2*i+2]:=tmp;`
	`  end;`

Си	**Алгоритмический язык**

`for (i=0;i<=10;i++)`
`  A[i]=i;`
`for (i=0;i<5;i++)`
`{`
`  tmp= A[i];`
`  A[i]=A[2*i+2];`
`  A[2*i+2]=tmp;`
`}`
	`нц для i от 0 до 10`
	`  A[i]:=i`
	`кц`
	`нц для i от 0 до 4`
	`  tmp := A[i]`
	`  A[i]:=A[2*i+2]`
	`  A[2*i+2]:=tmp`
	`кц`

Чему будут равны элементы этого массива после выполнения фрагмента программы?

1) 2 1 4 3 6 5 8 7 0 9 10
2) 2 4 6 8 10 5 0 7 3 9 1
3) 1 0 3 2 5 4 7 6 9 8 10
4) 0 1 2 3 4 5 6 7 8 9 10

См. учебно-справочные материалы: Одномерные массивы, с. 218.

A13 Система команд исполнителя РОБОТ, «живущего» в прямоугольном лабиринте на клетчатой плоскости:

вверх	вниз	влево	вправо

При выполнении любой из этих команд РОБОТ перемещается на одну клетку соответственно: вверх ↑, вниз ↓, влево ←, вправо →.

Четыре команды проверяют истинность условия отсутствия стены у каждой стороны той клетки, где находится РОБОТ:

сверху свободно	**снизу свободно**	**слева свободно**	**справа свободно**

Цикл ПОКА <*условие*> *команда* выполняется, пока условие истинно, иначе происходит переход на следующую строку.

Если РОБОТ начнёт движение в сторону стены, то он разрушится и программа прервётся.

Сколько клеток лабиринта соответствует требованию, что, выполнив предложенную программу, РОБОТ уцелеет и остановится в той же клетке, с которой он начал движение?

НАЧАЛО

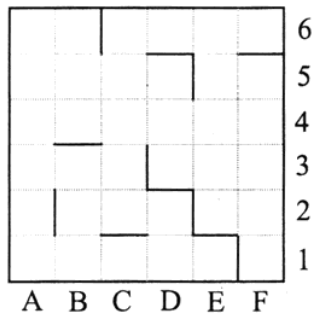
ПОКА **<сверху свободно> вверх**

ПОКА **<справа свободно> вправо**

ПОКА **<снизу свободно> вниз**

ПОКА **<слева свободно> влево**

КОНЕЦ



1) 5 2) 4 3) 3 4) 2

См. учебно-справочные материалы: Формальный исполнитель РОБОТ, с. 179.

A14 Определите, какое число будет напечатано в результате работы следующей программы (для вашего удобства программа представлена на четырёх языках):

Бейсик	Паскаль

`Module A14`
`Sub Main()`
`Dim d, a, b, t, M, R As Double`
`a = -7 : b = -3`
`d = 0.1`
`t = a: M = a: R = F(a)`
`While t <= b`
	`Program A14;`
	`Uses crt;`
	`Var d,a,b,t,M,R :real;`
	`Function F(x: real):real;`
	`begin`
	`F:= (1-x)*(x+6);`
	`end;`

Продолжение таблицы

Бейсик	Паскаль

`If F(t) > R Then`
`M = t`
`R = F(t)`
`End If`
`t = t + d`
`End While`
`Console.Write(M)`
`End Sub`
`Function F(ByVal x As Double) As Double`
`Return (1-x)*(x+6)`
`End Function`
`End Module`
	`BEGIN`
	`a:=-7; b:=-3;`
	`d:=0.1;`
	`t:=a; M:=a; R:=F(a);`
	`while t<=b do`
	`begin`
	`if (F(t)> R) then`
	`begin`
	`M:=t;`
	`R:=F(t);`
	`end;`
	`t:=t+d;`
	`end;`
	`write(M);`
	`END.`

Си	**Алгоритмический язык**

`#include <stdio.h>`
`double F(double x)`
`{`
`return (1-x)*(x+6);`
`}`
`void main()`
`{`
`double d, a, b, t, M, R;`
`a = -7; b = -3;`
`d = 0.1;`
`t = a; M = a; R = F(a);`
`while (t<=b) {`
`if ( F(t)> R ) {`
`M = t; R = F(t);`
`}`
`t = t + d;`
`}`
`printf("%.0f", M);`
`}`
	`алг А14`
	`нач`
	`вещ d, a, b, t, M, R`
	`a:= -7; b:= -3`
	`d:= 0.1`
	`t:=a; M:=a; R:= F(a)`
	`нц пока t<=b`
	`если F(t)> R то`
	`M:= t; R:= F(t)`
	`все`
	`t:= t + d`
	`кц`
	`вывод M`
	`кон`
	`алг вещ F(вещ x)`
	`нач`
	`знач := (1-x)*(x+6)`
	`кон`

1) –7 2) –3 3) –2,5 4) 12

См. учебно-справочные материалы: Подпрограммы, определяемые разработчиком, с. 232.

Часть 2

Ответом к заданиям части 2 (B1–B14) является число, последовательность букв или цифр. Впишите ответы сначала в текст работы, а затем перенесите их в бланк ответов № 1 справа от номера соответствующего задания, начиная с первой клеточки, без пробелов, запятых и других дополнительных символов. Каждую букву или цифру пишите в отдельной клеточке в соответствии с приведёнными в бланке образцами.

B1 Автоматическое устройство осуществило перекодировку информационного сообщения на русском языке длиной 120 символов, первоначально записанного в 16-битной кодировке Unicode, в 8-битный код КОИ-8. На сколько битов уменьшилась длина сообщения? *В ответе запишите только число.*

Ответ: _____.

См. учебно-справочные материалы: Кодирование текстовой информации, с. 60.

B2 У исполнителя Вычислитель две команды, которым присвоены номера:

```
1. умножить на 3
2. вычесть 2
```

Первая из них умножает число на экране на 3, вторая вычитает из него 2.

Запишите порядок команд в алгоритме получения из числа 4 числа 48, содержащем не более 5 команд, указывая только номера команд (например, 11121 — это алгоритм:

```
1. умножить на 3
1. умножить на 3
1. умножить на 3
2. вычесть 2
1. умножить на 3
```

который преобразует число 1 в число 75).

Если таких алгоритмов более одного, запишите любой из них.

Ответ: _____.

См. учебно-справочные материалы: Формальный исполнитель Вычислитель, с. 185.

B3 Определите, что будет напечатано в результате работы следующего фрагмента программы:

Бейсик

```
Module Module1
    Sub Main()
        Dim a, b As Integer
        a = 11
        b = 7
       While a > b Or b < a
            If a > b Then
                a = a - 1
                b = b + 1
            Else
                a = a + 1
                b = b - 1
            End If
        End While
       Console.Write(a * a)
    End Sub

End Module
```

Паскаль

```
var a,b:integer;
a:=11;
b:=7;
while a>b or b<a do
begin
  if a> b then
  begin
    a:=a-1;
    b:=b+1;
  end
  else
  begin
    a:=a+1;
    b:=b-1;
  end;
end;
write(a*a);
```

Си

```
{
  int a,b;
  a = 11;
  b = 7;
  while ((a>b)||(b<a))
  if (a>b)
  {
    a:=a-1;
    b:=b+1;
  }
  else
  {
    a:=a+1;
    b:=b-1;
  }
  printf("%d",a*a);
}
```

Алгоритмический язык

```
нач
  цел a,b
  a:=11
  b:=7
  нц пока a>b или b<a
    если a>b то
      a:=a-1
      b:=b+1
    иначе
      a:=a+1
      b:=b-1
    все
  кц
  вывод a*a
кон
```

Ответ: _____.

См. учебно-справочные материалы: Алгоритмы, заданные на алгоритмическом языке, с. 193.

B4 Все пятибуквенные слова, составленные из букв А, И, О, У, записаны в алфавитном порядке. Вот начало списка:

1. ААААА
2. АААА И
3. АААА О
4. АААА У
5. ААА И А
6. ААА И И

.....

Запишите слово, которое стоит на **512-м месте** от начала списка.

Ответ: _____.

См. учебно-справочные материалы: Позиционные системы счисления, с. 40.

B5 Дан фрагмент электронной таблицы:

	A	B	C	D
1	4	4	2	
2	=A1+B1	=B1+C1	=C1+D1	=D1+A1

Какое число должно быть записано в ячейке D1, чтобы построенная по значениям диапазона ячеек A2:D2 диаграмма соответствовала рисунку?

Ответ: _____.

B6 Определите значение переменной *c* после выполнения следующего фрагмента программы, в котором *a*, *b* и *c* — переменные вещественного (действительного) типа.

Бейсик	Паскаль

a = 144
b = 84
IF a / 3 > b - a / 3 THEN
  c = a - b / 2
  ELSE
  c = 2 * b - a
ENDIF
	a := 144;
	b := 84;
	if a / 3 > b - a / 3 then
	  c := a - b / 2
	else
	  c := 2 * b - a;

Си	**Алгоритмический язык**

a = 144;
b = 84;
if (a / 3 > b - a / 3)
  c = a - b / 2;
else
  c = 2 * b - a;
	a:= 144
	b:= 84
	если (a / 3 > b - a / 3)
	  то c := a - b / 2
	  иначе c := 2 * b - a
	все

Ответ: _____.

См. учебно-справочные материалы: Алгоритмы, заданные на алгоритмическом языке, с. 193.

B7 Ниже приведены тексты одной и той же программы, записанные на четырёх языках программирования. Что будет напечатано в результате выполнения этой программы?

Алгоритмический язык

```
алг Задача
нач
      цел L=5; целтаб R[1:L]; цел N, p
      p:=1
      R[1]:=2;R[2]:=2;R[3]:=1;R[4]:=3;R[5]:=2
      ArrProcessing_1(L,R,p)
      если p>5 то
            вывод "Переполнение"; стоп
      все
      N:=MultCounter(L,R)
      вывод N
кон
алг ArrProcessing_1(цел L, аргрез целтаб R[1:L],
рез цел p)
```

Продолжение таблицы

Алгоритмический язык

```
нач
      цел i,t
      p:=2
      нц для i от 1 до L - 1

            R[i+1]:=R[i]*p
            p:=i
      кц
кон
алг цел MultCounter(цел L, аргрез целтаб R[1:L])
нач
      цел N, i, T
      N:=0
      T:=1
      нц для i от 2 до L
            T:=R[i-1]
            N:=N+T*(T-1)
      кц
      знач:=N
кон
```

Бейсик

```
Module Task
    Sub Main()
        Dim L As Integer = 5
        Dim R(5) As Integer
        Dim N, p As Integer
        p = 1
        R.SetValue(2, 1)
        R.SetValue(2, 2)
        R.SetValue(1, 3)
        R.SetValue(3, 4)
        R.SetValue(2, 5)
        ArrProcessing_1(L, R, p)
        If p > 5 Then
            Console.Write("Переполнение")
            Stop
        End If
        N = MultCounter(L, R)
        Console.Write(N)
    End Sub
```

Продолжение таблицы

Бейсик

```
    Sub ArrProcessing_1(ByVal L As Integer, ByRef
R As Array, ByRef p As Integer)
        Dim i, t As Integer
        p = 2
        For i = 1 To L - 1
            R.SetValue(R.GetValue(i) * p, i + 1)
            p = i
        Next
    End Sub
    Function MultCounter(ByVal L As Integer, ByRef
R As Array) As Integer
        Dim N, i, T As Integer
        N = 0
        T = 1
        For i = 2 To L
            T = R.GetValue(i - 1)
            N = N + T * (T - 1)
        Next
        Return N
    End Function
End Module
```

Паскаль

```
Program Task;
Uses crt;
const L = 5;
type
    atype = array [1..L] of integer;
Var R: atype;
    N,p: integer;
Procedure  ArrProcessing_1(L:  integer;  var  R:
atype; var p:integer);
          var i,t: integer;
begin
     p:=2;
     for i:=1 to L-1 do
     begin
          R[i+1]:=R[i]*p;
          p:=i;
     end;
end;
```

Продолжение таблицы

Паскаль

```
Function   MultCounter(L:   integer;   var   R:
atype):integer;
         var N,i,T: integer;
begin
     N:=0;
     T:=1;
     for i:=2 to L do
     begin
          T:=R[i-1];
          N:=N+T*(T-1);
     end;
     MultCounter:=N;
end;
BEGIN
     clrscr;
     p:=1;
     R[1]:=2;R[2]:=2;R[3]:=1;R[4]:=3;R[5]:=2;
     ArrProcessing_1(L,R,p);
     if p>5 then
     begin
          write('Переполнение');
          halt;
     end;
     N:=MultCounter(L,R);
     write(N);
END.
```

Си

```
#include <stdio.h>
#include <stdlib.h>

void ArrProcessing_1(int L, int* R, int* p)
{
     int i, t;
     *p = 2;
     for(i=0; i<L-1; i++)
     {
               R[i+1]= R[i] * *p;
               *p = i + 1;
     }
 }
```

Продолжение таблицы

Си

`int MultCounter(int L, int* R)`
`{`
`  int N, i, T;`
`  N = 0;`
`  T = 1;`
`  for (i=1; i<L; i++)`
`  {`
`  T=R[i-1];`
`  N = N + T*(T-1);`
`  }`
`  return N;`
`}`
 	
`void main()`
`{`
`  int L = 5;`
`  int* R = (int*) calloc(L, sizeof(int));`
`  int N, p = 1;`
`  R[0] = 2; R[1] = 2; R[2] = 1; R[3] = 3; R[4] = 2;`
`  ArrProcessing_1(L, R, &p);`
`  if (p>5)`
`  {`
`  printf("Переполнение");`
`  free(R);`
`  return;`
`  }`
`  N = MultCounter(L,R);`
`  printf("%d\n", N);`
`  free(R);`
`  return;`
`}`

Ответ: _____.

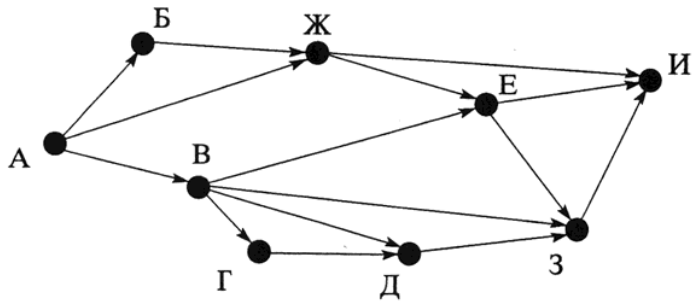
См. учебно-справочные материалы: Подпрограммы, определяемые разработчиком, с. 232.

B8 Запись числа 91_{10} в системе счисления с основанием N оканчивается на 1 и содержит 3 цифры. Сколько существует различных N, для которых это справедливо?

Ответ: _____.

См. учебно-справочные материалы: Позиционные системы счисления, с. 40.

B9 На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город И?



Ответ: _____.

B10 У Васи есть доступ в Интернет по высокоскоростному одностороннему радиоканалу, обеспечивающему скорость получения им информации 2^{20} бит/с. У Миши нет скоростного доступа в Интернет, но есть возможность получать информацию от Васи по низкоскоростному телефонному каналу со средней скоростью 2^{16} бит/с. Миша договорился с Васей, что тот будет скачивать для него данные объёмом 10 Мбайт по высокоскоростному каналу и ретранслировать их Мише по низкоскоростному каналу.

Компьютер Васи может начать ретрансляцию данных не раньше, чем им будут получены первые 2 Мбайт этих данных. Каков минимально возможный промежуток времени (в секундах) с момента начала скачивания Васей данных до полного их получения Мишей?

В ответе укажите только число, слово «секунд» или букву «с» добавлять не нужно.

Ответ: _____.

См. учебно-справочные материалы: Передача информации, с. 20.

B11 В терминологии сетей TCP/IP **маской сети** называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес. Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске.

По заданным IP-адресу узла и маске определите адрес сети.
IP-адрес узла: 111.115.238.168
Маска: 255.255.255.240

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса и запишите в нужном порядке соответствующие им буквы, без использования точек.

A	B	C	D	E	F	G	H
240	114	168	115	101	160	111	238

Пример.

Пусть искомый IP-адрес сети — 192.168.120.0 и дана таблица

A	B	C	D	E	F	G	H
255	*11*	*166*	*120*	*127*	*0*	*192*	*168*

В этом случае правильный ответ будет записан в виде GHDF.

Ответ: _____.

См. учебно-справочные материалы: Подсети. Маски подсетей, с. 270.

B12 В языке запросов поискового сервера для обозначения логической операции «ИЛИ» используется символ «|», для логической операции «И» — символ «&», для логической операции «НЕ» — символ «-».

В таблице приведены запросы и количество найденных по ним страниц некоторого сегмента сети Интернет:

Запрос	Найдено страниц (в тысячах)
Математика & Химия	630
Математика & Физика	750
Химия & Физика	400
Математика & Химия & Физика	305
Математика | Химия | Физика	4500

Какое количество страниц (в тысячах) будет найдено по запросу

(*Математика* | *Химия* | *Физика*) &
-((*Математика* & *Химия*) | (*Математика* & *Физика*) |
(*Химия* & *Физика*))?

Считается, что все запросы выполнялись практически одновременно, так что набор страниц, содержащих все искомые слова, не изменялся за время выполнения запросов.

Ответ: _____.

См. учебно-справочные материалы: Элементы теории множеств, с. 132.

B13 У исполнителя Вычислитель две команды:

```
1. прибавь 2
2. вычти 3
```

Первая из них увеличивает число на экране на 2, вторая уменьшает его на 3 (отрицательные числа допускаются).

Программа для Вычислителя — это последовательность команд. Сколько различных чисел можно получить из числа 7 с помощью программы, которая содержит ровно 10 команд?

Ответ: _____.

B14 Сколько различных решений имеет система уравнений

$$(a \wedge \neg b) \vee (\neg a \wedge b) \vee (c \wedge d) \vee (\neg c \wedge \neg d) = 1$$
$$(c \wedge \neg d) \vee (\neg c \wedge d) \vee (e \wedge f) \vee (\neg e \wedge \neg f) = 1$$
$$(e \wedge \neg f) \vee (\neg e \wedge f) \vee (g \wedge h) \vee (\neg g \wedge \neg h) = 1$$

$(g \wedge \neg h) \vee (\neg g \wedge h) \vee (i \wedge j) \vee (\neg i \wedge \neg j) = 1$
$(i \wedge \neg j) \vee (\neg i \wedge j) \vee (a \wedge b) \vee (\neg a \wedge \neg b) = 1$
где $a, b, c, d, e, f, g, h, i, j$ — логические переменные?

В ответе запишите количество различных наборов значений $a, b, c, d, e, f, g, h, i, j$, при которых выполняется данная система равенств.

Ответ: _____.

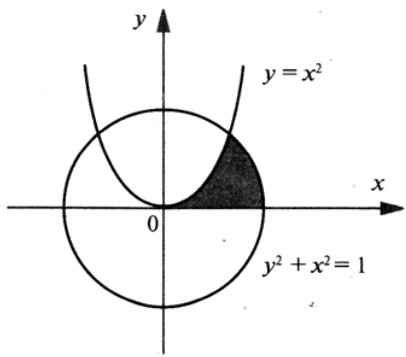
См. учебно-справочные материалы: Построение СДНФ логической функции, с. 102.

Не забудьте перенести все ответы в бланк ответов № 1.

Часть 3

Для записи ответов к заданиям части 3 (С1–С4) используйте бланк ответов № 2. Запишите сначала номер задания (С1, С2 и т.д.), а затем полное решение. Ответы записывайте чётко и разборчиво.

С1 Требовалось написать программу, при выполнении которой с клавиатуры считываются координаты точки на плоскости (x, y — действительные числа) и определяется принадлежность этой точки заданной закрашенной области (включая границы). Программист торопился и написал программу неправильно.



Паскаль	Бейсик	Си

var x, y: real;
begin
readln(x, y);
if y <= x*x then
if y*y + x*x <= 1
then
if y >= 0 then
write
('принадлежит')
else
write
('не принадлежит')
end.
	INPUT x,y
	IF y <= x*x THEN
	IF y*y + x*x <= 1 THEN
	IF y >= 0 THEN
	PRINT
	'принадлежит'
	ELSE
	PRINT _
	'не принадлежит'
	ENDIF
	ENDIF
	ENDIF
	END
		void main(void)
		{float x,y;
		scanf
		("%f%f",&x,&y);
		if (y <= x*x)
		if (y*y + x*x <= 1)
		if (y >= 0)
		printf
		("принадлежит");
		else
		printf
		("не принадлежит");
		}

Последовательно выполните следующее:

1. Приведите пример таких чисел x, y, при которых программа неверно решает поставленную задачу. Объясните, почему для указанных чисел программа неверно решает поставленную задачу.
2. Укажите, как нужно доработать программу, чтобы не было случаев её неправильной работы. (Это можно сделать несколькими способами, достаточно указать любой способ доработки исходной программы.)

C2 Дан массив из 30 целых чисел. Элементы массива имеют значение от –100 до 100 включительно. Необходимо разработать программу, которая вычисляет среднее арифметическое всех элементов с чётными значениями. Гарантируется, что среди элементов массива есть хотя бы один элемент с чётным значением. Количество элементов с чётным значением и среднее значение этих элементов вывести на экран.

Фрагмент программы с объявлением переменных и заполнением массива приведён ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из описанных переменных.

Паскаль	Бейсик

const n = 30;
var a:array[1..n] of integer;
i, s, k:integer;
sr:real;
begin
for i:=1 to n do
readln(a[i]);
.....
end.
	N = 30
	DIM A(N) AS INTEGER
	DIM I, S, K AS INTEGER
	DIM SR AS SINGLE
	FOR I = 1 TO N
	INPUT A(I)
	NEXT I
	.....
	END

Си	**Естественный язык**

#include <stdio.h>
#define N 30
void main()
{
int a[N], i, s, k;
float sr;
for(i=0; i<N; i++)
scanf("%d",&a[i]);
.....
}
	Объявляем массив A из 30 элементов.
	Объявляем целочисленные переменные I, S, K и вещественную переменную SR.
	В цикле от 1 до 30 вводим элементы массива A с 1-го по 30-й.
	.....

В качестве ответа вам необходимо привести фрагмент программы (или описание алгоритма на естественном языке), который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например Borland Pascal 7.0) или в виде блок-схемы. В этом случае вы должны использовать переменные, аналогичные переменным, используемым в алгоритме, записанном на естественном языке, с учётом синтаксиса и особенностей используемого вами языка программирования.

C3 У исполнителя Вычислитель две команды, которым присвоены номера:

1. прибавь 3
2. умножь на 2

Первая команда увеличивает число на 3, вторая — увеличивает число вдвое.

Программа для Вычислителя — это последовательность команд.

Сколько есть программ, которые число 1 преобразуют в число 28?

C4 На сервер локальной сети приходят запросы не более чем от 10 фирм. Сервер ведёт регистрацию всех поступающих за день запросов. Вам предлагается написать эффективную, в том числе по используемой памяти, программу, которая будет статистически обрабатывать приходящие на сервер запросы, с целью определения числа запросов, приходящих от той или иной фирмы. Следует учитывать, что число запросов в списке может быть очень велико.

Перед текстом программы кратко опишите используемый вами алгоритм решения задачи.

На вход программе в первой строке подаётся количество пришедших запросов **N**. В каждой из последующих **N** строк записано название фирмы в виде текстовой строки. Длина строки не превосходит 100 символов, название может содержать буквы, цифры, пробелы, знаки препинания.

Пример входных данных:

```
6
Рога и Копыта
Рожки да ножки
Рога и Копыта
Пиявки да Лягушки
Нитки и Катушки
Пряники да Плюшки
```

Программа должна вывести названия всех фирм, встречающихся в списке, в порядке возрастания (не убывания) количества запросов, пришедших за день от той или иной фирмы, с указанием количества запросов. При этом информация о каждой фирме должна быть выведена ровно один раз.

Пример выходных данных для приведённого выше примера входных данных:

```
Нитки и Катушки 1
Пряники да Плюшки 1
Пиявки да Лягушки 1
Рожки да ножки 1
Рога и Копыта 2
```

Вариант 4

Часть 1

При выполнении заданий части 1 в бланке ответов № 1 под номером выполняемого вами задания (А1–А14) поставьте знак «×» в клеточке, номер которой соответствует номеру выбранного вами ответа.

А1 Сколько значащих нулей в двоичной записи числа 272?

1) 7 2) 8 3) 5 4) 6

См. учебно-справочные материалы: Перевод числа из одной позиционной системы счисления в другую, с. 44.

А2 Между четырьмя местными аэропортами: ЮЖНЫЙ, СЕВЕРНЫЙ, ЗАПАДНЫЙ и ВОСТОЧНЫЙ — ежедневно выполняются авиарейсы. Приведено расписание полётов между ними:

Аэропорт вылета	Аэропорт прилёта	Время вылета	Время прилёта
ЗАПАДНЫЙ	СЕВЕРНЫЙ	13:10	14:20
ЮЖНЫЙ	СЕВЕРНЫЙ	13:30	15:00
ЮЖНЫЙ	ЗАПАДНЫЙ	14:20	17:55
ВОСТОЧНЫЙ	ЮЖНЫЙ	14:45	16:55
ВОСТОЧНЫЙ	СЕВЕРНЫЙ	15:30	16:20
СЕВЕРНЫЙ	ЗАПАДНЫЙ	16:05	17:45
ЮЖНЫЙ	ЗАПАДНЫЙ	16:15	18:00
СЕВЕРНЫЙ	ВОСТОЧНЫЙ	17:10	19:50
ЗАПАДНЫЙ	ВОСТОЧНЫЙ	18:20	19:05
ЗАПАДНЫЙ	ВОСТОЧНЫЙ	18:45	19:25
ЮЖНЫЙ	ВОСТОЧНЫЙ	19:20	20:05

Путешественник оказался в аэропорту ЮЖНЫЙ в 13:00 и хочет попасть в аэропорт ВОСТОЧНЫЙ. Считается, что путешественник успевает совершить пересадку между рейсами в каком-либо промежуточном аэропорту, если между временем прилёта в этот аэропорт и временем вылета из него проходит не менее часа. Определите самое раннее время, когда путешественник может попасть в аэропорт ВОСТОЧНЫЙ.

1) 19:05 2) 19:25 3) 19:50 4) 20:05

См. учебно-справочные материалы: Элементы теории графов, с. 124.

A3 Дан фрагмент таблицы истинности выражения **F**:

X	Y	Z	F
0	0	0	**0**
0	0	1	**1**
1	1	1	**0**

Каким может быть **F**?

1) $\neg X \vee \neg Y \vee Z$
2) $\neg X \wedge Y \wedge Z$
3) $X \vee Y \vee \neg Z$
4) $\neg X \wedge \neg Y \wedge Z$

См. учебно-справочные материалы: Нормальные формы логических функций, с. 97.

A4 Для групповых операций с файлами используются **маски имён файлов**. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы:

— символ «?» (вопросительный знак) означает ровно один произвольный символ;

— символ «*» (звёздочка) означает любую последовательность символов произвольной длины, в том числе символ «*» может задавать и пустую последовательность.

Определите, по какой из масок будет выбрана указанная группа файлов:

```
ddddb.dll
bdddd.dll
dddd.dll
dddb.cs
```

1) `*dd*.d?` 2) `*d*.*` 3) `*ddb*.?**` 4) `*b*.??*`

См. учебно-справочные материалы: Подсети. Маски подсетей, с. 270.

A5 Автомат получает на вход два трёхзначных числа. По этим числам строится новое число по следующим правилам:

1. Записывается результат сложения значений старших разрядов заданных чисел.
2. К нему дописывается результат сложения значений средних разрядов этих чисел по такому правилу: если он меньше первой суммы, то второе полученное число приписывается к первому справа, иначе — слева.
3. Итоговое число получают приписыванием слева к полученному после второго шага числу суммы значений младших разрядов исходных чисел.

Пример. Исходные трёхзначные числа: 583, 724. Поразрядные суммы: 12, 10, 7. Результат: 71210.

Определите, какое из предложенных чисел может быть результатом такой операции.

1) 191211 2) 16127 3) 81215 4) 12912

См. учебно-справочные материалы: Основы алгоритмизации, с. 161.

A6 Ниже приведены фрагменты таблиц базы данных абитуриентов.

Таблица «Абитуриент»

ID_abit	ID_City	Mat	Inf	Russ	Ball
1015	02	82	71	95	248
1026	77	80	90	65	235
1034	77	78	86	67	231

Продолжение таблицы

ID_abit	ID_City	Mat	Inf	Russ	Ball
1041	24	82	86	70	238
1046	65	78	95	76	249
1050	77	85	90	68	243
1067	65	96	86	95	277
1093	08	95	78	92	265
1094	65	84	87	75	246
1102	02	98	80	67	245
1105	65	82	78	75	235

Таблица «Направления»

ID_Napr	Name_Napr	Ball
231000	Программная инженерия	249
080500	Бизнес-информатика	245
230700	Прикладная информатика	240
010400	Прикладная математика и информатика	242

Таблица «Город»

ID_ City	Name_City
02	Уфа
24	Красноярск
65	Саратов
77	Москва

Таблица «Заявления»

ID_abit	ID_Napr
1015	231000
1026	080500

Продолжение таблицы

ID_abit	ID_Napr
1026	230700
1034	010400
1034	080500
1034	231000
1041	080500
1046	230700
1046	080500
1050	231000
1050	010400
1067	231000
1067	010400
1093	230700
1094	010400
1102	231000
1102	010400
1105	231000
1105	010400
1105	080500

Сколько абитуриентов из Саратова набрали проходной балл и могут быть зачислены на направление «Программная инженерия»?

1) 1 2) 2 3) 3 4) 4

См. учебно-справочные материалы: Реляционная база данных, с. 244.

A7 В динамической (электронной) таблице приведены количества товара, поставленного в некоторые компании, и количества проданного товара. Какая из компаний наименее успешно реализовала товар в своих магазинах? Успешность реализации определяется отношением проданного товара к поставленному. Чем оно выше, тем успешнее работает магазин.

Компания		Товар				
		ноутбуки	смартфоны	приставки	телевизоры	всего
«Цифровой мир»	поставлено	800	300	200	2000	**3300**
	продано	700	150	50	1700	**2600**
«Сети для дома»	поставлено	500	430	320	1500	**2750**
	продано	450	300	300	1500	**2550**
«Интеллект»	поставлено	900	680	410	2500	**4490**
	продано	700	540	270	2000	**3510**
«Домашний досуг»	поставлено	800	530	180	1800	**3310**
	продано	500	350	100	1100	**2050**

1) «Цифровой мир»
2) «Сети для дома»
3) «Интеллект»
4) «Домашний досуг»

A8 Производится одноканальная (моно) звукозапись с частотой дискретизации 16 кГц. Запись длится 2 минуты 8 секунд, её результаты записываются в файл, сжатие данных не производится. Получен файл размером 4000 Кбайт. Определите глубину звука (разрешение) в битах.

1) 2 2) 8 3) 16 4) 24

См. учебно-справочные материалы: Кодирование звуковой информации, с. 70.

A9 Для кодирования некоторой последовательности, состоящей из букв **А**, **Б**, **В**, **Г** и **Д**, решили использовать неравномерный двоичный код, позволяющий однозначно декодировать двоичную последовательность, появляющуюся на приёмной стороне канала связи. Использовали код: А — 01, Б — 000, В — 001, Г — 11. Укажите, каким кодовым словом должна быть закодирована буква **Д**.

Длина этого кодового слова должна быть наименьшей из всех возможных. Код должен удовлетворять свойству однозначного декодирования.

1) 00
2) 111
3) 011
4) 10

См. учебно-справочные материалы: Префиксные коды, с. 8.

A10 Какое из приведённых названий рек удовлетворяет логическому условию:

(вторая буква гласная → первая буква гласная) ∧ последняя буква гласная?

1) ВЯТКА 2) ДНЕПР 3) ЯУЗА 4) ВОЛГА

См. учебно-справочные материалы: Основные законы алгебры логики, с. 92.

A11 Для регистрации на аукционе участнику требуется придумать пароль. Длина пароля — ровно 17 символов. В качестве символов используются десятичные цифры и буквы русского алфавита, причём все буквы используются в двух

начертаниях — как строчные, так и заглавные (регистр буквы имеет значение!).

Под хранение каждого такого пароля на компьютере отводится минимально возможное и одинаковое целое количество байтов, при этом используется посимвольное кодирование и все символы кодируются одинаковым и минимально возможным количеством битов. Определите объём памяти, который требуется для хранения 60 паролей.

1) 900 байт
2) 840 байт
3) 780 байт
4) 765 байт

См. учебно-справочные материалы: Измерение информации, с. 11.

A12 В программе используется одномерный целочисленный массив **A** с индексами от 0 до 10 и целочисленные переменные **tmp** и **i**. Ниже представлен фрагмент программы, записанный на разных языках программирования, в котором значения элементов сначала задаются, а затем меняются.

Бейсик	Паскаль

`FOR i=0 TO 10`
`  A(i)=i`
`NEXT i`
`FOR i=0 TO 4 STEP 2`
`  TMP = A(i)`
`  A(i) = A(10-i)`
`  A(10-i) = TMP`
`NEXT i`
	`for i:=0 to 10 do`
	`  A[i]:=i;`
	`for i:=0 to 4 do`
	`  if ( i mod 2 )=0`
	`  then`
	`  begin`
	`  tmp := A[i];`
	`  A[i]:=A[10-i];`
	`  A[10-i]:=tmp;`
	`  end;`

Си	**Алгоритмический язык**

`for (i=0;i<=10;i++)`
`  A[i]=i;`
`for (i=0;i<5;i+=2)`
`{`
`  tmp= A[i];`
`  A[i]=A[10-i];`
`  A[10-i]=tmp;`
`}`
	`нц для i от 0 до 10`
	`  A[i]:=i`
	`кц`
	`нц для i от 0 до 4 шаг 2`
	`  tmp := A[i]`
	`  A[i]:=A[10-i]`
	`  A[10-i]:=tmp`
	`кц`

Чему будут равны элементы этого массива после выполнения фрагмента программы?

1) 2 1 4 3 6 5 8 7 0 9 10
2) 10 9 8 7 6 5 4 3 2 1 0
3) 1 0 3 2 5 4 7 6 9 8 10
4) 10 1 8 3 6 5 4 7 2 9 0

См. учебно-справочные материалы: Одномерные массивы, с. 218.

A13 Система команд исполнителя РОБОТ, «живущего» в прямоугольном лабиринте на клетчатой плоскости:

вверх	**вниз**	**влево**	**вправо**

При выполнении любой из этих команд РОБОТ перемещается на одну клетку соответственно: вверх ↑, вниз ↓, влево ←, вправо →.

Четыре команды проверяют истинность условия отсутствия стены у каждой стороны той клетки, где находится РОБОТ:

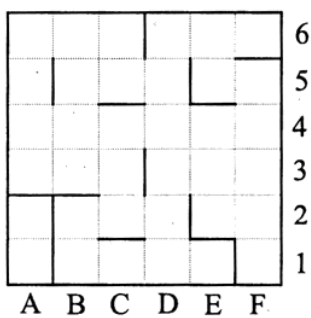
сверху свободно	**снизу свободно**	**слева свободно**	**справа свободно**

Цикл ПОКА <*условие*> *команда* выполняется, пока условие истинно, иначе происходит переход на следующую строку.

Если РОБОТ начнёт движение в сторону стены, то он разрушится и программа прервётся.

Сколько клеток лабиринта соответствует требованию, что, выполнив предложенную программу, РОБОТ уцелеет и остановится в той же клетке, с которой он начал движение?

НАЧАЛО
ПОКА <**справа свободно**> **вниз**
ПОКА <**снизу свободно**> **влево**
ПОКА <**слева свободно**> **вверх**
ПОКА <**сверху свободно**> **вправо**
КОНЕЦ



1) 0 2) 1 3) 2 4) 3

См. учебно-справочные материалы: Формальный исполнитель РОБОТ, с. 179.

A14 Определите, какое число будет напечатано в результате работы следующей программы (для вашего удобства программа представлена на четырёх языках):

Бейсик

```
Module A14
 Sub Main()
 Dim d, a, b, t, M, R As Double
    a = -3 : b = 2
    d = 0.1
    t = a: M = a: R = F(a)
    While t <= b
    If F(t) < R Then
              M = t
              R = F(t)
    End If
    t = t + d
    End While
        Console.Write(M)
 End Sub
 Function F(ByVal x
As Double) As Double
     Return (1-x)*(x+6)
 End Function
End Module
```

Паскаль

```
Program A14;
Uses crt;
Var d,a,b,t,M,R :real;
Function F(x : real):real;
  begin
    F:= (1-x)*(x+6);
  end;
BEGIN
a:=-3; b:=2;
d:=0.1;
t:=a; M:=a; R:=F(a);
while t<=b do
  begin
    if (F(t)< R) then
       begin
         M:=t;
         R:=F(t);
       end;
    t:=t+d;
  end;
write(M);
END.
```

Си

```
#include <stdio.h>
 double F(double x)
 {
   return (1-x)*(x+6);
 }
 void main()
 {
    double d, a, b, t, M, R;
   a = -3; b = 2;
   d = 0.1;
   t = a; M = a; R = F(a);
  while (t<=b) {
```

Алгоритмический язык

```
алг A14
 нач
   вещ d, a, b, t, M, R
    a:= -3; b:= 2
    d:= 0.1
    t:=a; M:=a; R:=F(a)
    нц пока t<=b
      если F(t) < R то
        M:= t; R:= F(t)
      все
      t:= t + d
    кц
```

Продолжение таблицы

Си	Алгоритмический язык

`if ( F(t) < R ) {`
`M = t; R = F(t);`
`}`
`t = t + d;`
`}`
`printf("%.0f", M);`
`}`
	`вывод M`
	`кон`
	`алг вещ F(вещ x)`
	`нач`
	`знач := (1-x)*(x+6)`
	`кон`

1) –8 2) –3 3) –2,5 4) 2

См. учебно-справочные материалы: Подпрограммы, определяемые разработчиком, с. 232.

Часть 2

Ответом к заданиям части 2 (B1–B14) является число, последовательность букв или цифр. Впишите ответы сначала в текст работы, а затем перенесите их в бланк ответов № 1 справа от номера соответствующего задания, начиная с первой клеточки, без пробелов, запятых и других дополнительных символов. Каждую букву или цифру пишите в отдельной клеточке в соответствии с приведёнными в бланке образцами.

B1 Следующее информационное сообщение на русском языке:

размéр сэмпла в звуковых файлах равен 8, 16 или 24 битам

— первоначально было записано в 16-битной кодировке Unicode. Затем оно было перекодировано в 8-битную кодировку ASCII. На сколько бит уменьшилась длина сообщения? *В ответе запишите только число.*

Ответ: _____.

См. учебно-справочные материалы: Кодирование текстовой информации, с. 60.

B2 У исполнителя Вычислитель две команды, которым присвоены номера:

```
1. делить на 2
2. прибавить 3
```

Первая из них делит нацело число на экране на 2, вторая прибавляет к нему 3.

Запишите порядок команд в алгоритме получения из числа 4 числа 14, содержащем не более 5 команд, указывая только номера команд (например, 12212 — это алгоритм:

```
1. делить на 2
2. прибавить 3
2. прибавить 3
1. делить на 2
2. прибавить 3
```

который преобразует число 40 в число 16).

Если таких алгоритмов более одного, запишите любой из них.

Ответ: _____.

См. учебно-справочные материалы: Формальный исполнитель Вычислитель, с. 185.

B3 Определите, что будет напечатано в результате работы следующего фрагмента программы:

Бейсик	Паскаль

`Module Module1`
`  Sub Main()`
`  Dim a, b As Integer`
`  a = 2`
`  b = 20`
`  While 5 * a - b <= 0`
`  If a > b Then`
`  a = a + 1`
`  Else`
`  a = a + 2`
`  End If`
`  End While`
`  Console.Write`
`  (a * a - 3 * a + b)`
`  End Sub`
 	
`End Module`
	`var a,b: integer;`
	`BEGIN`
	`  a:=2;`
	`  b:=20;`
	`  while 5*a-b<=0 do`
	`  begin`
	`  if a>b then`
	`  a:=a+1`
	`  else`
	`  a:=a+2;`
	`  end;`
	`  write(a*a-3*a+b);`
	`END.`

Си	Алгоритмический язык

`{`
`  int a,b;`
`  a = 2;`
`  b = 20;`
`  while (5*a-b<=0)`
`  if (a>b)`
`  a = a + 1;`
`  else`
`  a = a + 2;`
`  printf("%d", a*a - 3*a +b);`
`}`
	`нач`
	`  цел a,b`
	`  a:=2`
	`  b:=20`
	`  нц пока 5*a-b<=0`
	`  если a>b то`
	`  a:=a+1`
	`  иначе`
	`  a:=a+2`
	`  все`
	`  кц`
	`  вывод a*a - 3*a +b`
	`кон`

Ответ: _____.

См. учебно-справочные материалы: Алгоритмы, заданные на алгоритмическом языке, с. 193.

B4 Шестибуквенные слова, составленные из букв А, О, У, записаны в алфавитном порядке. Вот начало списка:

1. АААААО
2. АААААУ
3. ААААОА
4. ААААОО

.....

Запишите слово, которое стоит на **347-м месте** от начала списка.

Ответ: _____.

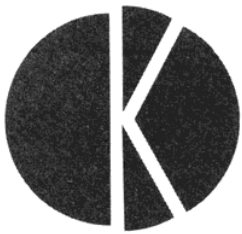
См. учебно-справочные материалы: Позиционные системы счисления, с. 40.

B5 Дан фрагмент электронной таблицы:

	A	B	C	D
19	3	4	1	
20	=A19–1	=B19+1	=C19	=D19*2
21				

Диапазон ячеек A20:D20 скопировали в диапазон A21:D21.

Какое число должно быть записано в ячейке D19, чтобы построенная по значениям диапазона ячеек A21:D21 диаграмма соответствовала рисунку?



Ответ: _____.

B6 Определите значение переменной *c* после выполнения следующего фрагмента программы, в котором *a*, *b* и *c* — переменные вещественного (действительного) типа.

Бейсик	Паскаль

`a = 48`
`b = 24`
`IF b > a / 2 - 1 THEN`
`  c = b + a / 4`
`ELSE`
`  c = b + a`
`ENDIF`
	`a:= 48;`
	`b:= 24;`
	`if b > a / 2 - 1`
	`then`
	`  c := b + a / 4`
	`else`
	`  c:= b + a;`

Си	**Алгоритмический язык**

`a = 48;`
`b = 24;`
`if (b > a / 2 - 1)`
`  c = b + a / 4;`
`else`
`  c = b + a;`
	`a := 48`
	`b := 24`
	`если (b > a / 2 – 1)`
	`то c := b + a / 4`
	`иначе c := b + a`
	`все`

Ответ: _____.

См. учебно-справочные материалы: Алгоритмы, заданные на алгоритмическом языке, с. 193.

B7 Ниже приведены тексты одной и той же программы, записанные на четырёх языках программирования. Что будет напечатано в результате выполнения этой программы?

Алгоритмический язык

```
алг Задача
нач
      цел L=2
      цел i
      целтаб A[1:L], B[1:L]
      цел N, p
      p:=15
      нц для i от 1 до L
            A[i]:=div(p,i)
      кц
      ArrCr(L,A,B,p)
      если p>18 то
            вывод "Переполнение"; стоп
      все
      N:=Combine(L,A,B)
      вывод N
кон
алг ArrCr(цел L, аргрез целтаб A[1:L], аргрез
целтаб B[1:L], рез цел p)
нач
      цел i
      p:=1
      нц для i от 1 до L
            B[i]:=A[i]+p
            p:=p+i-1
      кц
кон
алг цел Combine(цел L, аргрез целтаб A[1:L],
аргрез целтаб B[1:L])
нач
      цел N, i
      N:=0
      нц для i от 1 до L
            N:=N+A[i]+2*B[L-i+1]
      кц
      знач:=N+i
кон
```

Бейсик

```
Module Task
    Sub Main()
        Dim L As Integer = 2
        Dim i As Integer
        Dim A(L) As Integer
        Dim B(L) As Integer
        Dim N, p As Integer
        p = 15
        For i = 1 To L
            A.SetValue(p \ i, i)
        Next
        ArrCr(L, A, B, p)
        If p > 18 Then
            Console.Write("Переполнение")
            Stop
        End If
        N = Combine(L, A, B)
        Console.Write(N)
    End Sub
   Sub ArrCr(ByVal L As Integer, ByRef A As Array,
ByRef B As Array, ByRef p As Integer)
        Dim i As Integer
        p = 1
        For i = 1 To L
            B.SetValue(A.GetValue(i) + p, i)
            p = p + i - 1
        Next
    End Sub
    Function Combine(ByVal L As Integer, ByRef A
As Array, ByRef B As Array) As Integer
        Dim N, i As Integer
        N = 0
        For i = 1 To L
         N = N + A.GetValue(i) + 2 * B.GetValue(L - i + 1)
        Next
        Return N + i - 1
    End Function
End Module
```

Паскаль

```
Program Task;
Uses crt;
const L = 2;
type
    atype = array [1..L] of integer;
Var A,B: atype;
    N,p,i: integer;
Procedure ArrCr(L: integer; var A: atype; var
B:atype; var p:integer);
          var i: integer;
begin
     p:=1;
     for i:=1 to L do
     begin
          B[i]:=A[i]+p;
          p:=p+i-1;
     end;
end;
Function Combine(L: integer; var A: atype; var B:
atype):integer;
         var N,i: integer;
begin
     N:=0;
     for i:=1 to L do
     begin
          N:=N+A[i]+2*B[L-i+1];
     end;
     Combine:=N+i;
end;
BEGIN
     clrscr;
     p:=15;
     for i:=1 to L do
     begin
          A[i]:=p div i;
     end;
     ArrCr(L,A,B,p);
     if p>18 then
     begin
          write('Переполнение');
          halt;
     end;
     N:=Combine(L,A,B);
     write(N);
END.
```

Си

```
#include <stdio.h>
#include <stdlib.h>

void ArrCr(int L, int* A, int* B, int* p)
{
     int i;
     *p = 1;
     for(i = 0; i < L; i++)
     {
           B[i] = A[i] + *p;
           *p = *p+i;
     }
 }
int Combine(int L, int* A, int* B)
 {
     int N, i;
     N = 0;
     for (i = 0; i < L; i++)
          N = N + A[i] + 2*B[L-i-1];
     return N+i;
}

void main()
{
    int L = 2, i;
    int N, p = 15;
    int* A = (int*) calloc(L, sizeof(int));
    int* B = (int*) calloc(L, sizeof(int));
    for(i = 1; i <= L; i++)
          A[i-1] = p / i;
    ArrCr(L, A, B, &p);
    if (p>=18)
    {
            printf("Переполнение");
            free(A); free(B);
            return;
    }
    N = Combine(L, A, B);
    printf("%d\n", N);
    free(A);
    free(B);
}
```

Ответ: _____.

См. учебно-справочные материалы: Подпрограммы, определяемые разработчиком, с. 232.

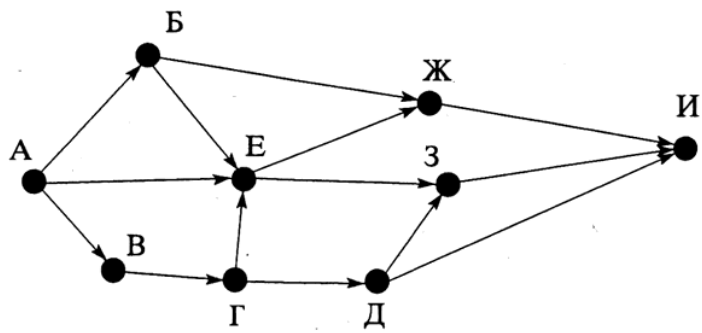
B8 Запись числа 107_{10} в системе счисления с основанием N оканчивается на 2 и содержит 3 цифры. Определите N, для которых это справедливо. Если таких чисел несколько, запишите их в ответе без пробелов. Число 10 в ответ записывать не надо.

Например, если N может принимать значения 2 или 3, в ответе будет записано 23.

Ответ: _____.

См. учебно-справочные материалы: Позиционные системы счисления, с. 40.

B9 На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город И?



Ответ: _____.

B10 У Васи есть доступ в Интернет по высокоскоростному одностороннему радиоканалу, обеспечивающему скорость получения им информации 2^{18} бит/с. У Миши нет скоростного доступа в Интернет, но есть возможность получать информацию от Васи по низкоскоростному телефонному каналу со средней скоростью 2^{14} бит/с. Миша договорился с Васей, что тот будет скачивать для него данные объёмом 7 Мбайт по высокоскоростному каналу и ретранслировать их Мише по низкоскоростному каналу.

Компьютер Васи может начать ретрансляцию данных не раньше, чем им будет получен 1 Мбайт этих данных. Каков минимально возможный промежуток времени (в секундах) с момента начала скачивания Васей данных до полного их получения Мишей?

В ответе укажите только число, слово «секунд» или букву «с» добавлять не нужно.

Ответ: _____.

См. учебно-справочные материалы: Передача информации, с. 20.

B11 В терминологии сетей TCP/IP **маской сети** называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес. Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске.

По заданным IP-адресу узла и маске определите адрес сети.
IP-адрес узла: 30.115.36.28
Маска: 255.255.240.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса и запишите в нужном порядке соответствующие им буквы, без использования точек.

A	B	C	D	E	F	G	H
32	30	250	115	28	0	240	36

Пример.

Пусть искомый IP-адрес сети — 192.168.120.0 и дана таблица

A	B	C	D	E	F	G	H
255	*11*	*166*	*120*	*127*	*0*	*192*	*168*

В этом случае правильный ответ будет записан в виде GHDF.

Ответ: _____.

См. учебно-справочные материалы: Подсети. Маски подсетей, с. 270.

B12 В языке запросов поискового сервера для обозначения логической операции «ИЛИ» используется символ «|», для логической операции «И» — символ «&», для логической операции «НЕ» — символ «-».

В таблице приведены запросы и количество найденных по ним страниц некоторого сегмента сети Интернет:

Запрос	Найдено страниц (в тысячах)
Огурец	4000
Салат & Помидор	3800
Салат & Огурец	3100
Помидор & Огурец	2300
Салат & Помидор & Огурец	1900

Какое количество страниц (в тысячах) будет найдено по запросу

(Салат & Помидор) | Огурец?

Считается, что все запросы выполнялись практически одновременно, так что набор страниц, содержащих все искомые слова, не изменялся за время выполнения запросов.

Ответ: _____.

См. учебно-справочные материалы: Элементы теории множеств, с. 132.

B13 У исполнителя Вычислитель две команды:

```
1. вычти 2
2. прибавь 3
```

Первая из них уменьшает число на экране на 2, вторая увеличивает его на 3 (отрицательные числа допускаются).

Программа для Вычислителя — это последовательность команд. Сколько различных чисел можно получить из числа 7 с помощью программы, которая содержит ровно 10 команд?

Ответ: _____.

B14 Сколько различных решений имеет уравнение

$((A \vee \neg B \vee C) \to \neg(D \to \neg E)) \wedge ((D \wedge E) \to (A \vee \neg B \vee C)) \wedge (D \vee E \vee \neg B) = 1$

где A, B, C, D, E — логические переменные?

Ответ: _____.

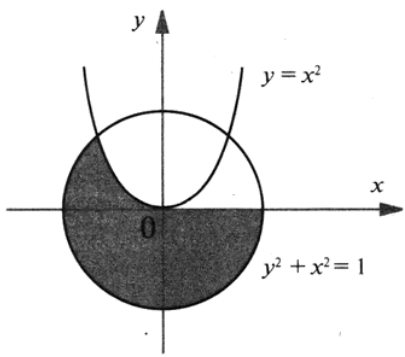
См. учебно-справочные материалы: Построение СДНФ логической функции, с. 102.

Не забудьте перенести все ответы в бланк ответов № 1.

Часть 3

Для записи ответов к заданиям части 3 (C1–C4) используйте бланк ответов № 2. Запишите сначала номер задания (C1, C2 и т. д.), а затем полное решение. Ответы записывайте чётко и разборчиво.

C1 Требовалось написать программу, при выполнении которой с клавиатуры считываются координаты точки на плоскости (x, y — действительные числа) и определяется принадлежность этой точки заданной закрашенной области (включая границы). Программист торопился и написал программу неправильно.



Паскаль	Бейсик	Си

var x, y: real;
begin
readln(x, y);
if y <= 0 then
if y <= x*x
then
if y*y+x*x<=1 then
write
('принадлежит')
else
write
('не принадлежит')
end.
	INPUT x,y
	IF y <= 0
	THEN
	IF y <= x*x
	THEN
	IF y*y+x*x<=1 THEN
	PRINT
	"принадлежит"
	ELSE
	PRINT
	"не принадлежит"
	ENDIF
	ENDIF
	ENDIF
	END
		void main(void)
		{float x,y;
		scanf
		("%f%f",&x,&y);
		if (y <= 0)
		if (y <= x*x)
		if (y*y+x*x<=1)
		printf
		("принадлежит");
		else
		printf
		("не принадлежит");
		}

Последовательно выполните следующее:

1. Приведите пример таких чисел x, y, при которых программа неверно решает поставленную задачу. Объясните, почему для указанных чисел программа неверно решает поставленную задачу.
2. Укажите, как нужно доработать программу, чтобы не было случаев её неправильной работы. (Это можно сделать несколькими способами, достаточно указать любой способ доработки исходной программы.)

C2 Дан массив из 40 целых чисел. Элементы массива имеют только одно из трёх значений: 1, 2, 3. Необходимо разработать программу, которая выполняет сортировку массива по правилу: в начале массива располагаются элементы со значением 3, далее элементы со значением 2 и далее элементы со значением 1. Запрещается использовать стандартные подпрограммы сортировки и любые массивы, кроме исходного массива. Отсортированный массив вывести на экран. Фрагмент программы с объявлением переменных и заполнением массива приведён ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из описанных переменных.

Паскаль	Бейсик

const n = 40;
var a:array[1..n] of integer;
  i,k1,k2,k3:integer;
begin
for i:=1 to n do
readln(a[i]);
.....
end.
	N = 40
	DIM A(N) AS INTEGER
	DIM I, K1, K2, K3, AS INTEGER
	FOR I = 1 TO N
	  INPUT A(I)
	NEXT I
	.....
	END

Си	**Естественный язык**

#include <stdio.h>
#define N 40
void main()
{
  int a[N], i, k1,k2,k3;
  for(i=0; i<N; i++)
  scanf("%d",&a[i]);
  .....
}
	Объявляем массив A из 40 элементов.
	Объявляем целочисленные переменные I, K1, K2, K3.
	В цикле от 1 до 40 вводим элементы массива A с 1-го по 40-й.
	.....

В качестве ответа вам необходимо привести фрагмент программы (или описание алгоритма на естественном языке), который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например Borland Pascal 7.0) или в виде блок-схемы. В этом случае вы должны использовать переменные, аналогичные переменным, используемым в алгоритме, записанном на естественном языке, с учётом синтаксиса и особенностей используемого вами языка программирования.

C3 У исполнителя Вычислитель две команды, которым присвоены номера:

1. прибавь 3
2. умножь на 2

Первая команда увеличивает число на 3, вторая — увеличивает число вдвое.

Программа для Вычислителя — это последовательность команд.

Сколько есть программ, которые число 23 преобразуют в число 58?

C4 В кафе подают кофе 12 разных сортов. При заказе в электронной учётной системе фиксируется название заказанного сорта кофе и стоимость покупки. Сорта пользуются разным спросом, поэтому один и тот же сорт может фигурировать в разных покупках, а некоторые могут вообще не покупаться.

Требуется написать как можно более эффективную программу, которая по данным о покупках из учётной системы формирует сводный отчёт со списком проданных сортов кофе и суммарной стоимостью проданного кофе каждого сорта. Названия сортов кофе в сводном отчёте не должны повторяться. Сводный отчёт должен быть отсортирован в порядке возрастания (не убывания) суммарной стоимости каждого проданного сорта кофе.

На вход программе подаётся строка, в которой указано количество покупок **N**. В каждой из последующих **N** строк записано название сорта кофе и сумма покупки в виде текстовой строки. Длина строки не превосходит 100 символов, название сорта кофе может содержать только буквы без пробелов.

Пример входных данных:

```
8
Суаре 500
Робуста 120
Арабика 700
Суаре 300
Суаре 100
Арабика 130
Арабика 100
Арабика 50
```

Пример выходных данных:

```
Робуста 120
Суаре 900
Арабика 980
```

При разработке программы следует учесть, что список сортов кофе может иметь объём, превышающий объём основной памяти компьютера.

Вариант 5

Часть 1

При выполнении заданий части 1 в бланке ответов № 1 под номером выполняемого вами задания (А1–А14) поставьте знак «×» в клеточке, номер которой соответствует номеру выбранного вами ответа.

А1 Сколько значащих нулей в двоичной записи числа 258?

1) 7 2) 8 3) 5 4) 6

См. учебно-справочные материалы: Перевод числа из одной позиционной системы счисления в другую, с. 44.

А2 Путешественник пришёл в 9:00 на автостанцию населённого пункта ЛИСЬЕ и обнаружил следующее расписание автобусов для всей районной сети маршрутов:

Пункт отправления	Пункт прибытия	Время отправления	Время прибытия
ЛИСЬЕ	ЕЖОВО	9:10	10:15
ЛИСЬЕ	СОБОЛЕВО	10:25	11:20
ЕЖОВО	СОБОЛЕВО	10:40	11:25
ЛИСЬЕ	ВОЛКОВО	10:55	12:50
СОБОЛЕВО	ЛИСЬЕ	10:20	11:10
СОБОЛЕВО	ЗАЙЦЕВО	11:30	12:20
ЕЖОВО	ЗАЙЦЕВО	11:35	13:30
СОБОЛЕВО	ЕЖОВО	11:40	12:25
ВОЛКОВО	ЗАЙЦЕВО	11:45	12:35
ЕЖОВО	ЗАЙЦЕВО	12:30	12:55

Считается, что путешественник успевает совершить пересадку на другой автобус, если разница между временем прибытия и отправления не менее 15 минут. Определите самое раннее время, когда путешественник сможет оказаться в пункте ЗАЙЦЕВО согласно этому расписанию.

1) 12:20 2) 12:55 3) 12:35 4) 13:30

См. учебно-справочные материалы: Элементы теории графов, с. 124.

A3 Дан фрагмент таблицы истинности выражения **F**:

X	Y	Z	F
0	1	0	**1**
1	0	1	**1**
1	1	0	**1**

Каким может быть **F**?

1) $X \vee \neg Y \vee Z$
2) $\neg X \vee Y \vee Z$
3) $\neg X \vee Y \vee \neg Z$
4) $\neg X \wedge \neg Y \wedge \neg Z$

См. учебно-справочные материалы: Нормальные формы логических функций, с. 97.

A4 Для групповых операций с файлами используются **маски имён файлов**. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы:

— символ «?» (вопросительный знак) означает ровно один произвольный символ;

— символ «*» (звёздочка) означает любую последовательность символов произвольной длины, в том числе символ «*» может задавать и пустую последовательность.

Определите, по какой из масок будет выбрана указанная группа файлов:

```
cccc.xml
bcc.xls
dca.xls
cdc.xml
```

1) `*c*.x???`
2) `?*c*.****`
3) `*c*a*.?**?`
4) `*c*c*.???`

См. учебно-справочные материалы: Подсети. Маски подсетей, с. 270.

A5 Автомат получает на вход два трёхзначных числа. По этим числам строится новое число по следующим правилам:

1. Записывается результат сложения значений старших разрядов заданных чисел.
2. К нему дописывается результат сложения значений средних разрядов этих чисел по такому правилу: если он меньше первой суммы, то второе полученное число приписывается к первому слева, иначе — справа.
3. Итоговое число получают приписыванием суммы значений младших разрядов исходных чисел по следующему правилу: если она меньше и первой, и второй суммы, то третье полученное число приписывается слева, если она больше и первой, и второй суммы, то справа, иначе — между двумя полученными суммами.

Пример. Исходные трёхзначные числа: 734, 928. Поразрядные суммы: 16, 5, 12. Результат: 51216.

Определите, какое из предложенных чисел может быть результатом такой операции.

1) 161210 2) 51219 3) 12815 4) 91317

См. учебно-справочные материалы: Основы алгоритмизации, с. 161.

A6 База данных о заказах кондитерских изделий ресторанами и кафе состоит из трёх связанных таблиц. Ниже приведены фрагменты этих таблиц.

Таблица «Заказы»

ID_заказа	ID_покупателя	ID_товара	Количество (кг)	Дата заказа	Дата доставки
11112	2148	65	36,6	22.06.2011	30.06.2011
11128	3945	11	6,8	24.06.2011	27.06.2011
11146	2148	42	14	25.06.2011	28.06.2011

Продолжение таблицы

ID_заказа	ID_покупателя	ID_товара	Количество (кг)	Дата заказа	Дата доставки
11157	6893	65	89,5	27.06.2011	30.06.2011
11161	4589	27	31,9	29.06.2011	04.07.2011
11168	5678	11	34,7	02.07.2011	07.07.2011
11172	2148	65	17	04.07.2011	09.07.2011
11179	3945	34	25	07.07.2011	12.07.2011

Таблица «Покупатели»

ID_покупателя	Наименование
2148	Кафе «Шоколадница»
2894	Ресторан «Сахалин»
3945	Ресторан «Оазис»
4589	Кафе «Кофе Хауз»
5678	Кафе «Эрна»
6893	Ресторан «Европа»

Таблица «Товары»

ID_товара	Наименование	Цена за 1 кг (руб.)
11	Конфеты «Чародейка»	480,00
27	Печенье «Клубничное»	510,00
34	Торт «Птичье молоко»	590,00
42	Печенье «Фруктовое»	390,00
58	Пастила	450,00
65	Конфеты «Мишка косолапый»	500,00

Сколько потратило кафе «Шоколадница» на покупку конфет «Мишка косолапый»?

1) 8500 2) 18300 3) 26800 4) 27388

См. учебно-справочные материалы: Реляционная база данных, с. 244.

А7 В динамической (электронной) таблице приведены количества абитуриентов, поступивших в четыре вуза, в том числе по результатам ЕГЭ. В каком вузе наименьшая часть студентов поступила по ЕГЭ?

Факультет	Количество студентов	Вуз			
		вуз-1	вуз-2	вуз-3	вуз-4
Математика	всего	260	240	180	150
	из них по ЕГЭ	200	200	170	100
Физика	всего	220	220	160	130
	из них по ЕГЭ	210	210	150	120
Психология	всего	230	210	170	120
	из них по ЕГЭ	200	170	140	100
Социология	всего	200	170	145	110
	из них по ЕГЭ	190	150	120	80
Итого	**всего**	**910**	**840**	**655**	**510**
	из них по ЕГЭ	**800**	**730**	**580**	**400**

1) вуз-1
2) вуз-2
3) вуз-3
4) вуз-4

А8 Производится одноканальная (моно) звукозапись с 24-битным разрешением. Запись длится 1 минуту 4 секунды, её результаты записываются в файл, сжатие данных не производится. Получен файл размером 3000 Кбайт. Определите частоту дискретизации в килогерцах.

1) 2 2) 6 3) 16 4) 44,1

См. учебно-справочные материалы: Кодирование звуковой информации, с. 70.

А9 Для кодирования некоторой последовательности, состоящей из букв **А**, **Б**, **В**, **Г** и **Д**, решили использовать неравномерный двоичный код, позволяющий однозначно декоди-

ровать двоичную последовательность, появляющуюся на приёмной стороне канала связи. Использовали код: А — 0, Б — 100, В — 101, Г — 110. Укажите, каким кодовым словом должна быть закодирована буква **Д**.

Длина этого кодового слова должна быть наименьшей из всех возможных. Код должен удовлетворять свойству однозначного декодирования.

1) 00
2) 10
3) 111
4) 1111

См. учебно-справочные материалы: Префиксные коды, с. 8.

A10 Какое из приведённых географических названий удовлетворяет логическому условию:

(первая буква согласная → вторая буква согласная) ∧ ¬ последняя буква гласная?

1) НОВГОРОД
2) МЦЕНСК
3) ВОРОНЕЖ
4) РЕВДА

См. учебно-справочные материалы: Основные законы алгебры логики, с. 92.

A11 В олимпиаде по программированию участвуют 67 команд. Индивидуальный код команды состоит из 13 символов. В качестве символов используют 25 различных букв и десятичные цифры в произвольном порядке.

Каждый такой код в компьютерной программе записывается одинаковым минимально возможным целым количеством байтов, при этом используют посимвольное кодирование. Каждый символ кодируется одинаковым минимально возможным целым количеством битов. Для изменения рейтинга команды индивидуальный код команды загружается в специальную программу. Какой объём памяти будет задействован, если одновременно будет загружена информация о 23 командах?

1) 207 байт
2) 224 байт
3) 230 байт
4) 299 байт

См. учебно-справочные материалы: Измерение информации, с. 11.

A12 В программе используется одномерный целочисленный массив **A** с индексами от 0 до 10 и целочисленные переменные **tmp** и **i**. Ниже представлен фрагмент программы, записанный на разных языках программирования, в котором значения элементов сначала задаются, а затем меняются.

Бейсик	Паскаль

`FOR i=0 TO 10`
`  A(i)=i`
`NEXT i`
`FOR i=0 TO 10 STEP 1`
`  TMP = A(i)`
`  A(i) = A((10-i)\2)`
`  A((10-i)\2) = TMP`
`NEXT i`
	`for i:=0 to 10 do`
	`  A[i]:=i;`
	`for i:=0 to 10 do`
	`  begin`
	`  tmp := A[i];`
	`  A[i]:=A[(10-i)div 2];`
	`  A[(10-i)div 2]:=tmp;`
	`  end;`

Си	**Алгоритмический язык**

`for (i=0;i<=10;i++)`
`  A[i]=i;`
`for (i=0;i<11;i++)`
`{`
`  tmp= A[i];`
`  A[i]=A[(10-i)/2];`
`  A[(10-i)/2]=tmp;`
`}`
	`нц для i от 0 до 10`
	`  A[i]:=i`
	`кц`
	`нц для i от 0 до 10 шаг 1`
	`  tmp := A[i]`
	`  A[i]:=A[div((10-i),2)]`
	`  A[div((10-i),2)]:=tmp`
	`кц`

Чему будут равны элементы этого массива после выполнения фрагмента программы?

1) 2 1 4 3 6 5 8 7 0 9 10
2) 10 9 8 7 6 5 4 3 2 1 0
3) 10 8 6 2 3 1 0 4 7 5 9
4) 4 0 2 3 1 5 6 7 8 9 10

См. учебно-справочные материалы: Одномерные массивы, с. 218.

A13 Система команд исполнителя РОБОТ, «живущего» в прямоугольном лабиринте на клетчатой плоскости:

вверх	вниз	влево	вправо

При выполнении любой из этих команд РОБОТ перемещается на одну клетку соответственно: вверх ↑, вниз ↓, влево ←, вправо →.

Четыре команды проверяют истинность условия отсутствия стены у каждой стороны той клетки, где находится РОБОТ:

сверху свободно	**снизу свободно**	**слева свободно**	**справа свободно**

Цикл ПОКА *<условие> команда* выполняется, пока условие истинно, иначе происходит переход на следующую строку.

Если РОБОТ начнёт движение в сторону стены, то он разрушится и программа прервётся.

Сколько клеток лабиринта соответствует требованию, что, выполнив предложенную программу, РОБОТ уцелеет и остановится в той же клетке, с которой он начал движение?

НАЧАЛО

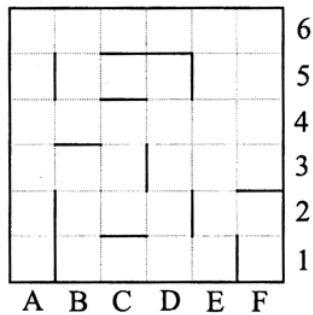
ПОКА < **сверху свободно**> **влево**

ПОКА < **слева свободно**> **вниз**

ПОКА <**снизу свободно**> **вправо**

ПОКА <**справа свободно**> **вверх**

КОНЕЦ



1) 1 2) 2 3) 3 4) 4

См. учебно-справочные материалы: Формальный исполнитель РОБОТ, с. 179.

A14 Определите, какое число будет напечатано в результате работы следующей программы (для вашего удобства программа представлена на четырёх языках):

Бейсик	Паскаль

`Module A14`
`Sub Main()`
`Dim d, a, b, t, M, R As Double`
	`Program A14;`
	`Uses crt;`
	`Var d,a,b,t,M,R :real;`

Продолжение таблицы

Бейсик	Паскаль

`a = -3 : b = 2`
`d = 0.1`
`t = a: M = a: R = F(a)`
`While t <= b`
`If F(t) < R Then`
`M = t`
`R = F(t)`
`End If`
`t = t + d`
`End While`
`Console.Write(M)`
`End Sub`
 	
`Function F(ByVal x`
`(As Double) As Double`
`Return (x+3)*(x-4)`
`End Function`
`End Module`
	`Function F(x: real):real;`
	`begin`
	`F:= (x+3)*(x-4);`
	`end;`
	`BEGIN`
	`a:=-3; b:=2;`
	`d:=0.1;`
	`t:=a; M:=a; R:=F(a);`
	`while t<=b do`
	`begin`
	`if (F(t) < R) then`
	`begin`
	`M:=t;`
	`R:=F(t);`
	`end;`
	`t:=t+d;`
	`end;`
	`write(M);`
	`END.`

Си	**Алгоритмический язык**

`#include <stdio.h>`
`double F(double x)`
`{`
`return (x+3)*(x-4);`
`}`
`void main()`
`{`
`double d, a, b, t, M, R;`
`a = -3; b = 2;`
`d = 0.1;`
`t = a; M = a; R = F(a);`
`while (t<=b) {`
`if ( F(t) < R ) {`
`M = t; R = F(t);`
`}`
`t = t + d;`
`}`
`printf("%.1f", M);`
`}`
	`алг A14`
	`нач`
	`вещ d,a,b,t,M,R`
	`a:= -3; b:= 2`
	`d:= 0.1`
	`t:=a; M:=a; R:=F(a)`
	`нц пока t<=b`
	`если F(t)<R то`
	`M:=t; R:=F(t)`
	`все`
	`t:= t + d`
	`кц`
	`вывод M`
	`кон`
	`алг вещ F(вещ x)`
	`нач`
	`знач := (x+3)*(x-4)`
	`кон`

1) −12,25 2) −3 3) 0,5 4) 2

См. учебно-справочные материалы: Подпрограммы, определяемые разработчиком, с. 232.

Часть 2

Ответом к заданиям части 2 (B1–B14) является число, последовательность букв или цифр. Впишите ответы сначала в текст работы, а затем перенесите их в бланк ответов № 1 справа от номера соответствующего задания, начиная с первой клеточки, без пробелов, запятых и других дополнительных символов. Каждую букву или цифру пишите в отдельной клеточке в соответствии с приведёнными в бланке образцами.

B1 Автоматическое устройство осуществило перекодировку информационного сообщения на русском языке, первоначально записанного в 8-битной кодировке ASCII, в 16-битный Unicode. В результате преобразования информационное сообщение увеличилось на 1 Килобит. Какова длина сообщения в символах? *В ответе запишите только число.*

Ответ: _____.

См. учебно-справочные материалы: Кодирование текстовой информации, с. 60.

B2 У исполнителя Вычислитель две команды, которым присвоены номера:

```
1. умножить на 3
2. вычесть 2
```

Первая из них умножает число на экране на 3, вторая вычитает из него 2.

Запишите порядок команд в алгоритме получения из числа 7 числа 27, содержащем не более 4 команд, указывая только номера команд (например, 11121 — это алгоритм:

```
1. умножить на 3
1. умножить на 3
1. умножить на 3   .
```

```
2. вычесть 2
1. умножить на 3
```

который преобразует число 1 в число 75).

Если таких алгоритмов более одного, запишите любой из них.

Ответ: _____.

См. учебно-справочные материалы: Формальный исполнитель Вычислитель, с. 185.

B3 Определите, что будет напечатано в результате работы следующего фрагмента программы:

Бейсик	Паскаль

`Module Module1`
`  Sub Main()`
`  Dim a, b As Integer`
`  a = 8`
`  b = -4`
`  While Not a = 2`
`  If b > 0 Then`
`  a = a \ 2`
`  Else`
`  a = a \ 2`
`  b = -b`
`  End If`
`  End While`
`  a = -a + 2 * b`
`  Console.Write(a)`
`  End Sub`
 	
`End Module`
	`var a,b: integer;`
	`BEGIN`
	`  a:=8;`
	`  b:=-4;`
	`  while not(a=2) do`
	`  if b>0 then`
	`  a:= a div 2`
	`  else`
	`  begin`
	`  a:= a div 2;`
	`  b:=-b;`
	`  end;`
	`  a:=-a+2*b;`
	`  write(a);`
	`END.`

Си	**Алгоритмический язык**

`{`
`  int a,b;`
`  a = 8;`
`  b = -4;`
`  while !(a==2)`
`  if(b>0)`
`  a = a / 2;`
`  else`
`  {`
	`нач`
	`  цел a,b`
	`  a:=8`
	`  b:=-4`
	`  нц пока не(a=2)`
	`  если b>0 то`
	`  a:=div(a,2)`
	`  иначе`
	`  a:=div(a,2)`

Продолжение таблицы

Си	Алгоритмический язык

`a = a / 2;`
`b = -b;`
`}`
`a = -a+2*b;`
`printf("%d", a);`
`}`
	`b:=-b`
	`все`
	`кц`
	`a:=-a+2*b`
	`вывод a`
	`кон`

Ответ: _____.

См. учебно-справочные материалы: Алгоритмы, заданные на алгоритмическом языке, с. 193.

B4 Световое табло состоит из лампочек, каждая из которых может находиться в одном из 3 состояний (включена, выключена, мигает). Какое наименьшее количество лампочек *n* должно находиться на табло, чтобы с его помощью можно было передать **122 различных сигнала**, используя не менее одной и не более *n* лампочек?

Ответ: _____.

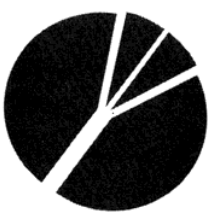
См. учебно-справочные материалы: Позиционные системы счисления, с. 40.

B5 Дан фрагмент электронной таблицы:

	A	B	C	D
1	2	-1		-2
2	= –A1	=B1*C1	=C1*D1	=D1*A1
3				

Диапазон ячеек A2:D2 скопировали в диапазон A3:D3.

Какое число должно быть записано в ячейке C1, чтобы построенная по значениям диапазона ячеек A3:D3 диаграмма соответствовала рисунку?



Ответ: _____.

B6 Определите значение переменной *c* после выполнения следующего фрагмента программы, в котором *a*, *b* и *c* — переменные вещественного (действительного) типа.

Бейсик	Паскаль

`a = 18`
`b = 7`
`IF b > a / 3 - 1 THEN`
`  c = (b - a) * 3`
`  ELSE`
`  c = b + a / 2`
`ENDIF`
	`a:= 18;`
	`b:= 7;`
	`if b > a / 3 - 1`
	`then`
	`  c := (b - a) * 3`
	`else`
	`  c := b + a / 2;`

Си	**Алгоритмический язык**

`a = 18;`
`b = 7;`
`if (b > a / 3 - 1)`
`  c = (b - a) * 3;`
`else`
`  c = b + a / 2;`
	`a := 18`
	`b := 7`
	`если (b > a / 3 - 1)`
	` то c := (b - a) * 3`
	` иначе c := b + a / 2`
	`все`

См. учебно-справочные материалы: Алгоритмы, заданные на алгоритмическом языке, с. 193.

B7 Ниже приведены тексты одной и той же программы, записанные на четырёх языках программирования. Что будет напечатано в результате выполнения этой программы?

Алгоритмический язык

```
алг Задача
нач
      цел L=3
      цел i
      целтаб A[1:L], B[1:2*L]
      цел N, p
      p:=2
      нц для i от 1 до L
            A[i]:= p+i
      кц
      ArrCr(L,A,B,p)
      если p>5 то
            вывод "Переполнение"; стоп
```

Продолжение таблицы

Алгоритмический язык

```
      все
      N:=Combine(2*L,B)
      вывод N
кон
алг ArrCr(цел L, аргрез целтаб A[1:L], аргрез
целтаб B[1:2*L], рез цел p)
нач
      цел i
      p:=0
      нц для i от 1 до 2*L
            B[i]:=2*A[mod(i+1,2)+1]+p
            p:=i-1
      кц
кон
алг цел Combine(цел L, аргрез целтаб A[1:L])
нач
      цел N, i
      N:=0
      нц для i от 1 до L
            N:=N+A[i]
      кц
      знач:=N
кон
```

Бейсик

```
Module Task
    Sub Main()
        Dim L As Integer = 3
        Dim i As Integer
        Dim A(L) As Integer
        Dim B(2 * L) As Integer
        Dim N, p As Integer
        p = 2
        For i = 1 To L
            A.SetValue(p + i, i)
        Next
        ArrCr(L, A, B, p)
        If p > 5 Then
            Console.Write("Переполнение")
            Stop
        End If
```

Продолжение таблицы

Бейсик

```
        N = Combine(2 * L, B)
        Console.Write(N)
    End Sub
      Sub ArrCr(ByVal L As Integer, ByRef A As
Array, ByRef B As Array, ByRef p As Integer)
        Dim i As Integer
        p = 0
        For i = 1 To 2 * L
          B.SetValue(2*A.GetValue((i+1) Mod 2 +1)+p, i)
          p = i - 1
        Next
    End Sub
    Function Combine(ByVal L As Integer, ByRef A
As Array) As Integer
        Dim N, i As Integer
        N = 0
        For i = 1 To L
            N = N + A.GetValue(i)
        Next
        Return N
    End Function
End Module
```

Паскаль

```
Program Task;
Uses crt;
const L = 3;
type
    atype = array [1..L] of integer;
    btype = array [1..2*L] of integer;
Var A: atype;
    B: btype;
    N,p,i: integer;
Procedure ArrCr(L: integer; var A: atype; var
B:btype; var p:integer);
          var i: integer;
begin
     p:=0;
     for i:=1 to 2*L do
```

Продолжение таблицы

Паскаль

```
     begin
          B[i]:=2*A[((i+1)mod 2)+1]+p;
          p:=i-1;
     end;
end;
Function Combine(L: integer; var A: btype):integer;
          var N,i: integer;
begin
     N:=0;
     for i:=1 to L do
     begin
          N:=N+A[i];
     end;
     Combine:=N;
end;
BEGIN
     clrscr;
     p:=2;
     for i:=1 to L do
     begin
          A[i]:=p + i;
     end;
     ArrCr(L,A,B,p);
     if p>5 then
     begin
          write('Переполнение');
          halt;
     end;
     N:=Combine(2*L,B);
     write(N);
END.
```

Си

```
#include <stdio.h>
#include <stdlib.h>

void ArrCr(int L, int* A, int* B, int* p)
{
     int i;
     *p = 0;
     for(i = 0; i < 2*L; i++)
```

Продолжение таблицы

Си

```
    {

            B[i] = 2*A[i%2] + *p;
            *p = i;
    }
 }
 int Combine(int L, int* A)
 {
     int N, i;
     N = 0;
     for (i = 0; i < L; i++)
          N = N + A[i];
     return N;
}

void main()
{
    int L = 3, i;
    int N, p = 2;
    int* A = (int*) calloc(L, sizeof(int));
    int* B = (int*) calloc(2*L, sizeof(int));
    for(i = 1; i <= L; i++)
          A[i-1] = p + i ;
    ArrCr(L, A, B, &p);
    if (p>5)
    {
            printf("Переполнение");
            free(A); free(B);
            return;
    }
    N = Combine(2*L, B);
    printf("%d\n", N);
    free(A);
    free(B);
}
```

Ответ: _____.

См. учебно-справочные материалы: Подпрограммы, определяемые разработчиком, с. 232.

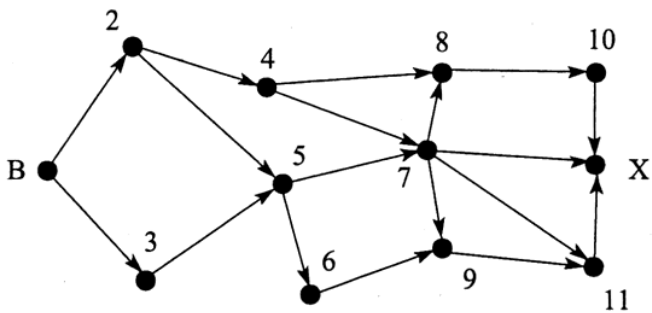
B8 Запись числа 128_{10} в системе счисления с основанием N оканчивается на 2 и содержит 3 цифры. Определите N, для которых это справедливо. Если таких чисел несколько, запишите их в ответе без пробелов. Число 10 в ответ записывать не надо.

Например, если N может принимать значения 2 или 3, в ответе будет записано 23.

Ответ: _____.

См. учебно-справочные материалы: Позиционные системы счисления, с. 40.

B9 На рисунке — схема лабиринта с развилками в точках 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, входом B и выходом X. По каждой дорожке лабиринта можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей прохода по лабиринту от входа B к выходу X?



Ответ: _____.

B10 У Васи есть доступ в Интернет по высокоскоростному одностороннему радиоканалу, обеспечивающему скорость получения им информации 2^{12} бит/с. У Миши нет скоростного доступа в Интернет, но есть возможность получать информацию от Васи по низкоскоростному телефонному каналу со средней скоростью 2^{11} бит/с. Миша договорился с Васей, что тот будет скачивать для него данные объёмом 2 Мбайт по высокоскоростному каналу и ретранслировать их Мише по низкоскоростному каналу.

Компьютер Васи может начать ретрансляцию данных не раньше, чем им будут получены первые 128 Кбайт этих данных. Каков минимально возможный промежуток времени (в секундах) с момента начала скачивания Васей данных до полного их получения Мишей?

В ответе укажите только число, слово «секунд» или букву «с» добавлять не нужно.

Ответ: _____.

См. учебно-справочные материалы: Передача информации, с. 20.

B11 В терминологии сетей TCP/IP **маской сети** называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес. Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске.

По заданным IP-адресу узла и маске определите адрес сети.
IP-адрес узла: 105.37.27.96
Маска: 255.255.255.192

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса и запишите в нужном порядке соответствующие им буквы, без использования точек.

A	B	C	D	E	F	G	H
27	37	17	64	113	96	105	192

Пример.

Пусть искомый IP-адрес сети — 192.168.120.0 и дана таблица

A	B	C	D	E	F	G	H
255	*11*	*166*	*120*	*127*	*0*	*192*	*168*

В этом случае правильный ответ будет записан в виде GHDF.

Ответ: _____.

См. учебно-справочные материалы: Подсети. Маски подсетей, с. 270.

B12 В языке запросов поискового сервера для обозначения логической операции «ИЛИ» используется символ «|», для логической операции «И» — символ «&», для логической операции «НЕ» — символ «-».

В таблице приведены запросы и количество найденных по ним страниц некоторого сегмента сети Интернет:

Запрос	Найдено страниц (в тысячах)
Уролог	5600
Диетолог & Окулист	4700
Диетолог & Уролог	4000
Окулист & Уролог	2300
Диетолог & Окулист & Уролог	1600

Какое количество страниц (в тысячах) будет найдено по запросу

(Диетолог | Окулист) & Уролог?

Считается, что все запросы выполнялись практически одновременно, так что набор страниц, содержащих все искомые слова, не изменялся за время выполнения запросов.

Ответ: _____.

См. учебно-справочные материалы: Элементы теории множеств, с. 132.

B13 У исполнителя Вычислитель две команды:

```
1. вычти 5
2. прибавь 3
```

Первая из них уменьшает число на экране на 5, вторая увеличивает его на 3 (отрицательные числа допускаются).

Программа для Вычислителя — это последовательность команд. Сколько различных чисел можно получить из числа 4 с помощью программы, которая содержит ровно 7 команд?

Ответ: _____.

B14 Сколько различных решений имеет уравнение

$\neg((A \to B) \to (C \wedge D \wedge E)) \vee \neg((A \wedge \neg B) \to \neg(C \wedge D \wedge E)) \vee (C \wedge A) = 0$

где A, B, C, D, E — логические переменные?

Ответ: _____.

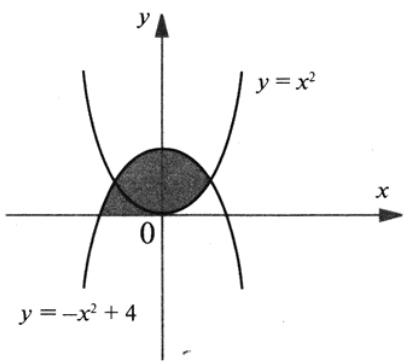
См. учебно-справочные материалы: Построение СДНФ логической функции, с. 102.

Не забудьте перенести все ответы в бланк ответов № 1.

Часть 3

Для записи ответов к заданиям части 3 (C1–C4) используйте бланк ответов № 2. Запишите сначала номер задания (C1, C2 и т. д.), а затем полное решение. Ответы записывайте чётко и разборчиво.

C1 Требовалось написать программу, при выполнении которой с клавиатуры считываются координаты точки на плоскости (x, y — действительные числа) и определяется принадлежность этой точки заданной закрашенной области (включая границы). Программист торопился и написал программу неправильно.



Паскаль	Бейсик	Си

var x, y: real;
begin
readln(x, y);
if y>=0 then
if y>=x*x then
if y<=-x*x+4 then
write
('принадлежит')
else
write
('не принадлежит')
end.
	INPUT x,y
	IF y>=0 THEN
	IF y>=x*x THEN
	IFy<=-x*x+4 THEN
	PRINT
	'принадлежит'
	ELSE
	PRINT
	'не принадлежит'
	ENDIF
	ENDIF
	ENDIF
	END
		void main(void)
		{float x,y;
		scanf
		("%f%f",&x,&y);
		if (y >= 0)
		if (y >= x*x)
		if (y<=-x*x+4)
		printf
		("принадлежит");
		else
		printf
		("не принадлежит");
		}

Последовательно выполните следующее:

1. Приведите пример таких чисел x, y, при которых программа неверно решает поставленную задачу. Объясните, почему для указанных чисел программа неверно решает поставленную задачу.
2. Укажите, как нужно доработать программу, чтобы не было случаев её неправильной работы. (Это можно сделать несколькими способами, достаточно указать любой способ доработки исходной программы.)

C2 Дан массив из 50 целых чисел. Элементы массива имеют значение от −100 до 100 включительно. Необходимо разработать программу, которая вычисляет и выводит количество перемен знака во всех парах соседних элементов. Гарантируется, что среди элементов массива нет элементов с нулевым значением.

Пример заполнения массива:

−1 4 2 −4 −1 2 3 4 5 6 1 2 3 4 1 2 3 4 5 6

Пара элементов **−1 4**: Есть перемена знака.
Пара элементов **4 2**: Нет перемены знака.
Пара элементов **2 −4**: Есть перемена знака.
Пара элементов **−4 −1**: Нет перемены знака.
Пара элементов **−1 2**: Есть перемена знака.
Далее во всех парах элементов перемены знака нет.
Итого количество перемен знака = 3.

Фрагмент программы с объявлением переменных и заполнением массива приведён ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из описанных переменных.

Паскаль	Бейсик

const n = 50;
var a:array[1..n] of integer;
i, k:integer;
begin
for i:=1 to n do
readln(a[i]);
.....
end.
	N = 50
	DIM A(N) AS INTEGER
	DIM I, K AS INTEGER
	FOR I = 1 TO N
	INPUT A(I)
	NEXT I
	.....
	END

Си	**Естественный язык**

#include <stdio.h>
#define N 50
void main()
{
int a[N], i, k;
for(i=0; i<N; i++)
scanf("%d",&a[i]);
.....
}
	Объявляем массив A из 50 элементов.
	Объявляем целочисленные переменные
	I, K.
	В цикле от 1 до 50 вводим элементы
	массива A с 1-го по 50-й.
	.....

В качестве ответа вам необходимо привести фрагмент программы (или описание алгоритма на естественном языке), который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например Borland Pascal 7.0) или в виде блок-схемы. В этом случае вы должны использовать переменные, аналогичные переменным, используемым в алгоритме, записанном на естественном языке, с учётом синтаксиса и особенностей используемого вами языка программирования.

C3 У исполнителя Калькулятор две команды, которым присвоены номера:

1. прибавь 1
2. умножь на 4

Первая команда увеличивает число на 1, вторая — увеличивает его в 4 раза.

Программа для Калькулятора — это последовательность команд.

Сколько есть программ, которые число 2 преобразуют в число 23?

C4 В локальной сети зарегистрировано не более 10 компьютеров. При поступлении запроса от каждого компьютера сервером регистрируется имя сетевого компьютера и число пришедших от него пакетов. Компьютеры проявляют различную сетевую активность, поэтому имя одного и того же компьютера может фигурировать несколько раз, а некоторые компьютеры могут вообще не встречаться.

Требуется написать как можно более эффективную программу, которая по записям о сетевой активности компьютера сформирует отчёт об именах, которые проявляли сетевую активность, и о суммарном количестве пакетов, пришедших от каждого из них. Названия компьютеров в отчёте не должны повторяться. Отчёт должен быть отсортирован в порядке убывания (не возрастания) суммарного количества пакетов.

На вход программе подаётся строка, в которой указано количество сетевых запросов **N**. В каждой из последующих **N** строк записано название компьютера и число пакетов, разделённые пробелом, в виде текстовой строки. Длина строки не превосходит 100 символов, название компьютера может содержать буквы, цифры и знаки препинания, но не пробелы.

Пример входных данных:

```
5
MyHost 1000
192.169.100.69 230
vH-14 1200
MyHost 1200
MyHost 10
```

Пример выходных данных:

```
MyHost 2210
vH-14 1200
192.169.100.69 230
```

При разработке программы следует учесть, что список числа запросов может иметь объём, превышающий объём основной памяти компьютера.

Вариант 6

Часть 1

При выполнении заданий части 1 в бланке ответов № 1 под номером выполняемого вами задания (А1–А14) поставьте знак «×» в клеточке, номер которой соответствует номеру выбранного вами ответа.

А1 Сколько значащих нулей в двоичной записи числа $254{,}8_{16}$?

1) 4 2) 8 3) 1 4) 6

См. учебно-справочные материалы: Перевод числа из одной позиционной системы счисления в другую, с. 44.

А2 Между четырьмя аэропортами, обозначенными кодами MSK, SPB, EKG, NSB, ежедневно выполняются авиарейсы. Ниже приведён фрагмент расписания.

Аэропорт вылета	Аэропорт прилёта	Время вылета	Время прилёта
MSK	SPB	9:05	9:40
SPB	NSB	10:45	12:50
MSK	SPB	10:55	14:30
MSK	EKG	11:20	14:45
NSB	MSK	14:05	17:00
SPB	NSB	14:45	18:05
EKG	MSK	15:55	18:50
MSK	NSB	17:05	20:15
EKG	NSB	17:35	18:40
SPB	NSB	18:20	19:15
NSB	EKG	19:05	20:20

Путешественник находится в 10:00 в аэропорту MSK. Считается, что путешественник успевает совершить пересадку между рейсами в каком-либо промежуточном аэропорту, если между временем прилёта в этот аэропорт и временем вылета из него проходит не менее часа. Определите самое позднее время отправления из MSK, если ему требуется быть в аэропорту NSB к 20:00.

1) 9:05 2) 10:55 3) 11:20 4) 17:05

См. учебно-справочные материалы: Элементы теории графов, с. 124.

A3 Дан фрагмент таблицы истинности выражения **F**:

X	Y	Z	F
0	1	1	**1**
1	0	0	**0**
1	1	1	**0**

Каким может быть **F**?

1) $\neg X \wedge Y \wedge Z$
2) $X \wedge Y \wedge Z$
3) $X \vee \neg Y \vee \neg Z$
4) $X \wedge \neg Y \wedge \neg Z$

См. учебно-справочные материалы: Нормальные формы логических функций, с. 97.

A4 Для групповых операций с файлами используются маски имён файлов. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы:

— символ «?» (вопросительный знак) означает ровно один произвольный символ;

— символ «*» (звёздочка) означает любую последовательность символов произвольной длины, в том числе символ «*» может задавать и пустую последовательность.

Определите, по какой из масок будет выбрана указанная группа файлов:

```
bbbddab.rar
dbbda.rar
```

```
bbba.rar
bbddabb.rar
```

1) `*bd*.rar` 2) `*a*.ra??` 3) `*b*a*.?ar**` 4) `*d*.rar*`

См. учебно-справочные материалы: Подсети. Маски подсетей, с. 270.

A5 Автомат получает на вход два трёхзначных числа. По этим числам строится новое число по следующим правилам:

1. Записывается результат сложения значений старших разрядов заданных чисел.
2. К нему дописывается результат сложения значений средних разрядов этих чисел по такому правилу: если он меньше первой суммы, то второе полученное число приписывается к первому справа, иначе — слева.
3. Итоговое число получают приписыванием суммы значений младших разрядов исходных чисел по следующему правилу: если она меньше и первой, и второй суммы, то третье полученное число приписывается справа, если она больше и первой, и второй суммы, то слева, иначе — между двумя полученными суммами.

Пример. Исходные трёхзначные числа: 937, 988. Поразрядные суммы: 18, 11, 15. Результат: 181511.

Определите, какое из предложенных чисел может быть результатом такой операции.

1) 171210 2) 121417 3) 161415 4) 191713

См. учебно-справочные материалы: Основы алгоритмизации, с. 161.

A6 Ниже приведены три связанные между собой таблицы базы данных склада канцтоваров.

Таблица «Товары»

ID_товара	Наименование товара
1034	Ластик
1075	Тетрадь общая, 96 л.
1090	Тетрадь общая, 48 л.

Продолжение таблицы

ID_товара	Наименование товара
1167	Линейка, 16 см
1189	Тетрадь общая, 12 л.
1297	Ручка синяя

Таблица «Поступление товаров»

ID_товара	Количество	Цена за ед. (руб.)	Дата
1034	1200	3.50	22.06.2011
1167	2000	12.00	24.06.2011
1297	6400	1.40	25.06.2011
1075	700	2.20	27.06.2011
1189	1900	3.80	29.06.2011
1090	1300	8.40	04.07.2011
1297	3500	2.10	08.07.2011

Таблица «Отгрузка товаров»

ID_товара	Количество	Цена за ед. (руб.)	Дата
1167	800	14.00	27.06.2011
1297	1200	2.80	28.06.2011
1297	700	3.40	30.06.2011
1075	700	4.20	30.06.2011
1189	300	5.60	30.06.2011
1297	2000	2.30	02.07.2011
1167	1000	16.60	04.07.2011
1189	800	4.80	04.07.2011

Сколько тетрадей было на складе 1 июля 2011 года?

1) 1000 2) 1600 3) 2100 4) 2600

См. учебно-справочные материалы: Реляционная база данных, с. 244.

А7 В динамической (электронной) таблице приведены данные о количестве сотрудников, обеспечивающих выполнение услуг ИТ-компаниями, и их суммарной заработной плате. Укажите компанию, которая выплачивает самую низкую среднюю заработную плату.

Услуги			Создание сайта	Поддержка сайта	Наполнение сайта	Продвижение сайта	Всего
Студии	У-Проджектс	Кол-во	3	2	1	2	**8**
		З/п	12000	8000	6000	7000	**33000**
	ООО «На коленке»	Кол-во	4	1	2	2	**9**
		З/п	10000	10000	16000	6000	**42000**
	Промо студиос	Кол-во	4	2	1	3	**10**
		З/п	20000	6000	5000	9000	**40000**
	Лаборатория Василия Уткина	Кол-во	3	1	2	2	**8**
		З/п	15000	5000	7000	8000	**35000**

1) Лаборатория Василия Уткина
2) Промо студиос
3) ООО «На коленке»
4) У-Проджектс

A8 Производится стереозвукозапись с частотой дискретизации 16 кГц и 16-битным разрешением. Запись длится 1 минуту 4 секунды, её результаты записываются в файл, сжатие данных не производится. Какое из приведённых ниже чисел наиболее близко к размеру полученного файла, выраженному в мегабайтах?

1) 0,4 2) 2 3) 3 4) 4

См. учебно-справочные материалы: Кодирование звуковой информации, с. 70.

A9 Для кодирования некоторой последовательности, состоящей из букв **А**, **Б**, **В**, **Г** и **Д**, решили использовать неравномерный двоичный код, позволяющий однозначно декодировать двоичную последовательность, появляющуюся на приёмной стороне канала связи. Использовали код: А–001, Б–011, В–100, Г–101. Укажите, каким кодовым словом должна быть закодирована буква **Д**.

Длина этого кодового слова должна быть наименьшей из всех возможных. Код должен удовлетворять свойству однозначного декодирования.

1) 01 2) 11 3) 000 4) 010

См. учебно-справочные материалы: Префиксные коды, с. 8.

A10 Какое из приведённых названий птиц удовлетворяет логическому условию:

есть буква И $\wedge$ (вторая буква согласная $\rightarrow$ первая буква согласная)?

1) ИНДЮК 2) АИСТ 3) ЦАПЛЯ 4) УТКА

См. учебно-справочные материалы: Основные законы алгебры логики, с. 92.

A11 В университете база данных хранит информацию о номерах студенческих билетов учащихся. Длина номера — ровно 10 символов. В качестве символов используются десятичные

цифры, знак тире и буквы русского алфавита (кроме «й», «ё», «ь», «ъ», «ы»), причём используются как строчные, так и заглавные буквы. (Регистр буквы имеет значение!)

Под хранение каждого такого номера на компьютере отводится минимально возможное и одинаковое целое количество байтов, при этом используется посимвольное кодирование и все символы кодируются одинаковым и минимально возможным количеством битов. Определите объём памяти, который необходим для хранения 90 номеров студенческих билетов.

1) 900 байт 2) 810 байт 3) 788 байт 4) 712 байт

См. учебно-справочные материалы: Измерение информации, с. 11.

A12 В программе используется одномерный целочисленный массив **A** с индексами от 0 до 10 и целочисленные переменные **tmp** и **i**. Ниже представлен фрагмент программы, записанный на разных языках программирования, в котором значения элементов сначала задаются, а затем меняются.

Бейсик	Паскаль

`FOR i=0 TO 10`
`  A(i)=i`
`NEXT i`
`FOR i=0 TO 4 STEP 1`
`  TMP = A(2*i)`
`  A(2*i) = A(2*i+1)`
`  A(2*i+1) = TMP`
`NEXT i`
	`for i:=0 to 10 do`
	`  A[i]:=i;`
	`for i:=0 to 4 do`
	`  begin`
	`  tmp := A[2*i];`
	`  A[2*i]:=A[2*i+1];`
	`  A[2*i+1]:=tmp;`
	`  end;`

Си	**Алгоритмический язык**

`for (i=0;i<=10;i++)`
`  A[i]=i;`
`for (i=0;i<5;i++)`
`{`
`  tmp= A[2*i];`
`  A[2*i]=A[2*i+1];`
`  A[2*i+1]=tmp;`
`}`
	`нц для i от 0 до 10`
	`  A[i]:=i`
	`кц`
	`нц для i от 0 до 4 шаг 1`
	`  tmp := A[2*i]`
	`  A[2*i]:=A[2*i+1]`
	`  A[2*i+1]:=tmp;`
	`кц`

Чему будут равны элементы этого массива после выполнения фрагмента программы?

1) 2 1 4 3 6 5 8 7 9 10 0
2) 1 0 3 2 5 4 7 6 9 8 10
3) 1 0 3 2 5 4 7 6 8 9 10
4) 0 2 1 4 3 6 5 8 7 10 9

См. учебно-справочные материалы: Одномерные массивы, с. 218.

A13 Система команд исполнителя РОБОТ, «живущего» в прямоугольном лабиринте на клетчатой плоскости:

вверх	**вниз**	**влево**	**вправо**

При выполнении любой из этих команд РОБОТ перемещается на одну клетку соответственно: вверх ↑, вниз ↓, влево ←, вправо →.

Четыре команды проверяют истинность условия отсутствия стены у каждой стороны той клетки, где находится РОБОТ:

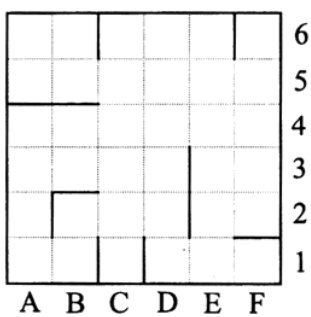
сверху свободно	**снизу свободно**	**слева свободно**	**справа свободно**

Цикл ПОКА <*условие*> *команда* выполняется, пока условие истинно, иначе происходит переход на следующую строку.

Если РОБОТ начнёт движение в сторону стены, то он разрушится и программа прервётся.

Сколько клеток лабиринта соответствует требованию, что, выполнив предложенную программу, РОБОТ уцелеет и остановится в той же клетке, с которой он начал движение?

НАЧАЛО
ПОКА **<сверху свободно> вверх**
ПОКА **<слева свободно> влево**
ПОКА **<снизу свободно> вниз**
ПОКА **<справа свободно> вправо**
КОНЕЦ



1) 1 2) 2 3) 3 4) 4

См. учебно-справочные материалы: Формальный исполнитель РОБОТ, с. 179.

A14 Определите, какое число будет напечатано в результате работы следующей программы (для вашего удобства программа представлена на четырёх языках):

Бейсик

```
Module A14
 Sub Main()
 Dim d,a,b,t,M, R As Double
      a = -3 : b = 2
      d = 0.1
      t = a: M=a: R= F(a)
         While t <= b
          If F(t) < R Then
               M = t
               R = F(t)
          End If
          t = t + d
         End While
         Console.Write(M)
 End Sub
 Function F(ByVal x
 As Double) As Double
    Return (x+1)*(x-1)
 End Function
End Module
```

Паскаль

```
Program A14;
Uses crt;
Var d,a,b,t,M,R :real;
Function F(x : real):real;
  begin
    F:= (x+1)*(x-1);
  end;
BEGIN
a:=-3; b:=2;
d:=0.1;
t:=a; M:=a; R:=F(a);
while t<=b do
  begin
    if (F(t) < R) then
       begin
         M:=t;
         R:=F(t);
       end;
    t:=t+d;
  end;
write(M);
END.
```

Си

```
 #include <stdio.h>
 double F(double x)
 {
   return (x+1)*(x-1);
 }
 void main()
 {
   double d,a,b,t,M,R;
```

Алгоритмический язык

```
 алг A14
 нач
     вещ d, a, b, t, M, R
     a:= -3; b:= 2
     d:= 0.1
     t:= a; M:= a; R:= F(a)
     нц пока t<=b
       если F(t) < R то
```

Продолжение таблицы

Си	Алгоритмический язык

`a = -3; b = 2;`
`d = 0.1;`
`t = a; M = a; R = F(a);`
`while (t<=b) {`
`if ( F(t) < R ) {`
`M = t; R = F(t);`
`}`
`t = t + d;`
`}`
`printf("%.1f", M);`
`}`
	`M:= t; R:= F(t)`
	`все`
	`t:= t + d`
	`кц`
	`вывод M`
	`кон`
	`алг вещ F(вещ x)`
	`нач`
	`знач := (x+1)*(x-1)`
	`кон`

1) –3 2) –1 3) 0 4) 2

См. учебно-справочные материалы: Подпрограммы, определяемые разработчиком, с. 232.

Часть 2

Ответом к заданиям части 2 (B1–B14) является число, последовательность букв или цифр. Впишите ответы сначала в текст работы, а затем перенесите их в бланк ответов № 1 справа от номера соответствующего задания, начиная с первой клеточки, без пробелов, запятых и других дополнительных символов. Каждую букву или цифру пишите в отдельной клеточке в соответствии с приведёнными в бланке образцами.

B1 Сообщение в 8-битной кодировке UTF-8 состояло из 36 символов. Затем его перекодировали в 16-битный Unicode и объединили с другим Unicode-сообщением объёмом 4 Кбит. Определите, на сколько байтов увеличилось итоговое сообщение по сравнению с первоначальным UTF-8 сообщением. *В ответе запишите только число.*

Ответ: _____.

См. учебно-справочные материалы: Кодирование текстовой информации, с. 60.

B2 У исполнителя Вычислитель две команды, которым присвоены номера:

```
1. умножить на 2
2. вычесть 1
```

Первая из них умножает число на экране на 2, вторая вычитает из него 1.

Запишите порядок команд в алгоритме получения из числа 7 числа 51, содержащем не более 5 команд, указывая только номера команд (например, 22111 — это алгоритм:

```
2. вычесть 1
2. вычесть 1
1. умножить на 2
1. умножить на 2
1. умножить на 2
```

который преобразует число 5 в число 24).

Если таких алгоритмов более одного, запишите любой из них.

Ответ: _____.

См. учебно-справочные материалы: Формальный исполнитель Вычислитель, с. 185.

B3 Определите, что будет напечатано в результате работы следующего фрагмента программы:

Бейсик	Паскаль

`Module Module1`
`  Sub Main()`
`  Dim a, b As Integer`
`  a = 8`
`  b = 5`
`  While Not (a - b < 0)`
`  If a > b Then`
`  a = a - 1`
`  Else`
`  b = b + 1`
`  End If`
`  End While`
`  Console.Write(a + b)`
`  End Sub`
 	
`End Module`
	`var a,b: integer;`
	`BEGIN`
	`  a:=8;`
	`  b:=5;`
	`  while not(a-b<0)`
	`  if a>b then`
	`  a:=a-1`
	`  else`
	`  b:=b+1;`
	`  write(a+b);`
	`END.`

Си	Алгоритмический язык

`{`
`  int a,b;`
`  a = 8;`
`  b = 5;`
`  while !(a - b < 0)`
`  if (a>b)`
`  a = a - 1;`
`  else`
`  b = b + 1;`
`  printf("%d", a+b);`
`}`
	`нач`
	`  цел a,b`
	`  a:=8`
	`  b:=5`
	`  нц пока не(a-b <0)`
	`  если a>b то`
	`  a:=a-1`
	`  иначе`
	`  b:=b+1`
	`  все`
	`  кц`
	`  вывод a+b`
	`кон`

Ответ: _____.

См. учебно-справочные материалы: Алгоритмы, заданные на алгоритмическом языке, с. 193.

B4 Для кодирования решено использовать последовательности разной длины из трёх знаков: D, O и G. Сколько различных символов можно закодировать, используя подобный код длиной не менее двух и не более четырёх знаков (D, O и G)?

Ответ: _____.

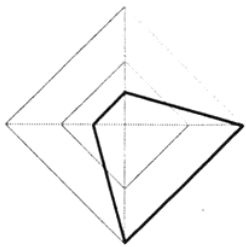
См. учебно-справочные материалы: Позиционные системы счисления, с. 40.

B5 Дан фрагмент электронной таблицы:

	C	D	E	F
9	1	–0,5	2	
10	=C9*D9	=D9*E9	= –E9	=F9*C9
11				

Диапазон ячеек C10:F10 скопировали в диапазон C11:F11.

Какое число должно быть записано в ячейке F9, чтобы построенная по значениям диапазона ячеек C11:F11 диаграмма соответствовала рисунку?



Ответ: _____.

B6 Определите значение произведения переменных *a* и *b* после выполнения следующего фрагмента программы, в котором *a* и *b* — переменные вещественного (действительного) типа.

Бейсик	Паскаль

`a = 5`
`b = a + 2*2`
`WHILE b >=a - 1`
`  b = b - a`
`  a = a + 1`
`END WHILE`
	`a := 5;`
	`b := a + 2*2;`
	`while b >= a - 1 do`
	`begin`
	`  b := b - a;`
	`  a := a + 1;`
	`end;`

Си	**Алгоритмический язык**

`a = 5;`
`b = a + 2*2;`
`while (b >= a - 1)`
`{`
`  b = b - a;`
`  a = a + 1;`
`}`
	`a := 5`
	`b := a + 2*2`
	`нц пока (b >= a - 1)`
	`  b := b - a;`
	`  a := a + 1;`
	`кц`

Ответ: _____.

См. учебно-справочные материалы: Алгоритмы, заданные на алгоритмическом языке, с. 193.

B7 Ниже приведены тексты одной и той же программы, записанные на четырёх языках программирования. Что будет напечатано в результате выполнения этой программы?

Алгоритмический язык

```
алг Задача
нач
      цел L=4
      цел i
      целтаб A[1:L]
      цел N, p
      p:=2
      нц для i от 1 до L
            A[i]:= 2*(i-1)+1
      кц
      ArrCr(L,A,p)
      если p=2 то
            вывод "Переполнение"; стоп
      все
      N:=Count(L,A)
      вывод N
кон
алг ArrCr(цел L, аргрез целтаб A[1:L], рез цел p)
нач
      цел i
      p:=7
      нц для i от 2 до L
            A[i]:=A[i-1]+3*p
            p:=mod(i-1,4)
      кц
кон
алг цел Count(цел L, аргрез целтаб A[1:L])
нач
      цел N, i
      N:=0
      нц для i от 1 до L
            N:=N+A[i]
      кц
      знач:=N
кон
```

Бейсик

```
Module Task
    Sub Main()
        Dim L As Integer = 4
        Dim i As Integer
        Dim A(L) As Integer
        Dim N, p As Integer
        p = 2
        For i = 1 To L
            A.SetValue(2 * (i - 1) + 1, i)
        Next
```

Продолжение таблицы

Бейсик

```
        ArrCr(L, A, p)
        If p = 2 Then
            Console.Write("Переполнение")
            Stop
        End If
        N = Count(L, A)
        Console.Write(N)
    End Sub
      Sub ArrCr(ByVal L As Integer, ByRef A As
Array, ByRef p As Integer)
        Dim i As Integer
        p = 7
        For i = 2 To L
          A.SetValue(A.GetValue(i - 1) + 3 * p, i)
          p = (i - 1) Mod 4
        Next
    End Sub
    Function Count(ByVal L As Integer, ByRef A As
Array) As Integer
        Dim N, i As Integer
        N = 0
        For i = 1 To L
            N = N + A.GetValue(i)
        Next
        Return N
    End Function
End Module
```

Паскаль

```
Program Task;
Uses crt;
const L = 4;
type
    atype = array [1..L] of integer;
Var A: atype;
    N,p,i: integer;
Procedure ArrCr(L: integer; var A: atype; var
p:integer);
          var i: integer;
begin
     p:=7;
     for i:=2 to L do
     begin
          A[i]:=A[i-1]+3*p;
          p:=(i-1) mod 4;
     end;
end;
```

Продолжение таблицы

Паскаль

```
Function Count(L: integer; var A: atype):integer;
         var N,i: integer;
begin
     N:=0;
     for i:=1 to L do
     begin
          N:=N+A[i];
     end;
     Count:=N;
end;
BEGIN
     clrscr;
     p:=2;
     for i:=1 to L do
     begin
          A[i]:= 2*(i-1)+1;
     end;
     ArrCr(L,A,p);
     if p=2 then
     begin
          write('Переполнение');
          halt;
     end;
     N:=Count(L,A);
     write(N);
END.
```

Си

```
#include <stdio.h>
#include <stdlib.h>

void ArrCr(int L, int* A, int* p)
{
     int i;
     *p = 7;
     for(i = 1; i < L; i++)
     {

           A[i] = A[i-1] + 3**p;
           *p = i % 4;
     }
 }
int Count(int L, int* A)
 {
     int N, i;
     N = 0;
```

Продолжение таблицы

Си

```
      for (i = 0; i < L; i++)
            N = N + A[i];
      return N;
}

void main()
{
    int L = 4, i;
    int N, p = 2;
    int* A = (int*) calloc(L, sizeof(int));
    for(i = 0; i < L; i++)
          A[i] = 2*i+1;
    ArrCr(L, A, &p);
    if (p==2)
    {
            printf("Переполнение");
            free(A);
            return;
    }
    N = Count(L, A);
    printf("%d\n", N);
    free(A);
}
```

Ответ: _____.

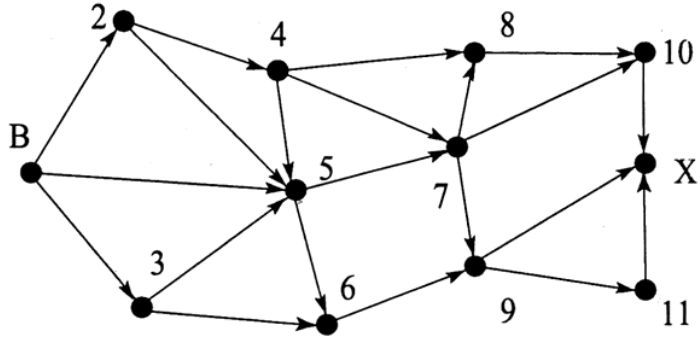
См. учебно-справочные материалы: Подпрограммы, определяемые разработчиком, с. 232.

B8 Запись числа 212_{10} в системе счисления с основанием N оканчивается на 2 и содержит 4 цифры. Чему равно основание этой системы счисления N?

Ответ: _____.

См. учебно-справочные материалы: Позиционные системы счисления, с. 40.

B9 На рисунке — схема лабиринта с развилками в точках 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, входом B и выходом X. По каждой дорожке лабиринта можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей прохода по лабиринту от входа B к выходу X?



Ответ: _____.

B10 У Васи есть доступ в Интернет по высокоскоростному одностороннему радиоканалу, обеспечивающему скорость получения им информации 2^{24} бит/с. У Миши нет скоростного доступа в Интернет, но есть возможность получать информацию от Васи по низкоскоростному телефонному каналу со средней скоростью 2^{18} бит/с. Миша договорился с Васей, что тот будет скачивать для него данные объёмом 15 Мбайт по высокоскоростному каналу и ретранслировать их по низкоскоростному каналу.

Компьютер Васи может начать ретрансляцию данных не раньше, чем им будут получены первые 2 Мбайт этих данных. Каков минимально возможный промежуток времени (в секундах) с момента начала скачивания Васей данных до полного их получения Мишей?

В ответе укажите только число, слово «секунд» или букву «с» добавлять не нужно.

Ответ: _____.

См. учебно-справочные материалы: Передача информации, с. 20.

B11 В терминологии сетей TCP/IP **маской сети** называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес. Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске.

По заданным IP-адресу узла и маске определите адрес сети.
IP-адрес узла: 122.79.115.170
Маска: 255.255.255.224

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса и запишите в нужном порядке соответствующие им буквы, без использования точек.

A	B	C	D	E	F	G	H
186	170	79	115	224	160	98	122

Пример.

Пусть искомый IP-адрес сети — 192.168.120.0 и дана таблица

A	B	C	D	E	F	G	H
255	*11*	*166*	*120*	*127*	*0*	*192*	*168*

В этом случае правильный ответ будет записан в виде GHDF.

Ответ: _____.

См. учебно-справочные материалы: Подсети. Маски подсетей, с. 270.

B12 В языке запросов поискового сервера для обозначения логической операции «ИЛИ» используется символ «|», а для логической операции «И» — символ «&».

В таблице приведены запросы и количество найденных по ним страниц некоторого сегмента сети Интернет.

Запрос	Найдено страниц (в тысячах)
Литва	2600
Швеция & Финляндия	850
Литва & Финляндия	670
Швеция & Литва	560
Финляндия & Швеция & Литва	153

Какое количество страниц (в тысячах) будет найдено по запросу

(Швеция & Финляндия) | Литва?

Считается, что все запросы выполнялись практически одновременно, так что набор страниц, содержащих все искомые слова, не изменялся за время выполнения запросов.

Ответ: _____.

См. учебно-справочные материалы: Элементы теории множеств, с. 132.

B13 У исполнителя Вычислитель две команды:

1. вычти 2
2. прибавь 2

Первая из них уменьшает число на экране на 2, вторая увеличивает его на 2 (отрицательные числа допускаются).

Программа для Вычислителя — это последовательность команд. Сколько различных чисел можно получить из числа 7 с помощью программы, которая содержит ровно 6 команд?

Ответ: _____.

B14 Сколько различных решений имеет уравнение

$$((A \to B) \to CDE)(CDE \to (-A \vee B))(C \to A) = 1$$

где A, B, E, C, D — логические переменные?

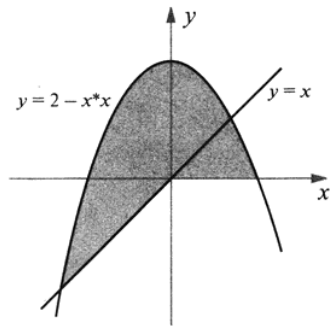
См. учебно-справочные материалы: Построение СДНФ логической функции, с. 102.

Не забудьте перенести все ответы в бланк ответов № 1.

Часть 3

Для записи ответов к заданиям части 3 (С1–С4) используйте бланк ответов № 2. Запишите сначала номер задания (С1, С2 и т. д.), а затем полное решение. Ответы записывайте чётко и разборчиво.

С1 Требовалось написать программу, при выполнении которой с клавиатуры считываются координаты точки на плоскости (x, y — действительные числа) и определяется принадлежность этой точки заданной закрашенной области (включая границы). Программист торопился и написал программу неправильно.



Паскаль	Бейсик	Си

var x, y: real;
begin
readln(x, y);
if y >= x then
if y >= 0 then
if y <= 2-x*x
then
write
('принадлежит')
else
write
('не принадлежит')
end.
	INPUT x,y
	IF y >= x
	THEN
	IF y >= 0
	THEN
	IF y <= 2-x*x
	THEN
	PRINT
	"принадлежит"
	ELSE
	PRINT
	"не принадлежит"
	ENDIF
	ENDIF
	ENDIF
	END
		void main(void)
		{float x,y;
		scanf
		("%f%f",&x,&y);
		if (y >= x)
		if (y >= 0)
		if (y <= 2-x*x
		printf
		("принадлежит");
		else
		printf
		("не принадлежит");
		}

Последовательно выполните следующее:

1. Приведите пример таких чисел x, y, при которых программа неверно решает поставленную задачу. Объясните, почему для указанных чисел программа неверно решает поставленную задачу.

2. Укажите, как нужно доработать программу, чтобы не было случаев её неправильной работы. (Это можно сделать несколькими способами, достаточно указать любой способ доработки исходной программы.)

C2 Дан целочисленный массив из 30 элементов. Элементы массива могут принимать целые значения от –500 до 500. Опишите на естественном языке или на одном из языков программирования алгоритм, позволяющий найти и вывести минимальное значение среди элементов массива, которые имеют чётное значение и не делятся на пять. Гарантируется, что в исходном массиве есть хотя бы один элемент, значение которого чётно и не кратно пяти.

Фрагмент программы с объявлением переменных и заполнением массива приведён ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из описанных переменных.

Паскаль	Бейсик

Const
N=30;
var
a: array [1..N] of
integer;
i, j, min: integer;
begin
for i:=1 to N do
readln(a[i]);
…
	N=30
	DIM A(N) AS INTEGER
	DIM I, J, MIN AS INTEGER
	FOR I = 1 TO N
	INPUT A(I)
	NEXT I
	…
	END

Си	**Естественный язык**

#include <stdio.h>
#define N 30
void main(void){
int a[N];
int i, j, min;
for (i=0; i<N; i++)
scanf("%d", &a[i]);
…
}
	Объявляем массив A из 30 элементов.
	Объявляем целочисленные переменные I, J, MIN. В цикле от 1 до 30 вводим элементы массива A с 1-го по 30-й.
	…

В качестве ответа вам необходимо привести фрагмент программы (или описание алгоритма на естественном языке), который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например Borland Pascal 7.0) или в виде блок-схемы. В этом случае вы должны использовать переменные, аналогичные переменным, используемым в алгоритме, записанном на естественном языке, с учётом синтаксиса и особенностей используемого вами языка программирования.

C3 Исполнитель Вычислитель имеет следующую систему пронумерованных команд:

```
1. умножь на 2
2. прибавь 1
```

Первая команда увеличивает число в 2 раза, а вторая прибавляет к нему 1.

Программа для Вычислителя — это последовательность команд.

Сколько есть программ, которые преобразуют число 1 в число 25?

C4 Пользователю системы электронных денег доступно не более 12 счетов, каждый счёт имеет уникальное имя. При добавлении средств сервер системы регистрирует номер счёта и сумму, при этом на счёт поступает 95% от суммы вносимых средств, а 5% уходят на комиссию системы. Вам предлагается написать эффективную, в том числе по используемой памяти, программу, которая будет составлять отчёт по пришедшим на сервер запросам и вычислять для каждого счёта сумму, поступившую на счёт, и размер комиссии. Следует учитывать, что количество запросов в списке может быть очень велико.

Перед текстом программы кратко опишите используемый вами алгоритм решения задачи.

На вход программе в первой строке подается количество пришедших запросов **N**. В каждой из последующих **N** строк записаны имя счёта и сумма, разделённые запятой. Длина строки не превосходит 100 символов, уникальное имя счёта может содержать буквы и цифры.

Пример входных данных:

```
6
WQ001,100
ZT123,200
10567,300
WQ001,130
ZT123,400
WQ001,270
```

Программа должна вывести уникальные имена всех счетов, встречающихся в списке, в порядке убывания (не возрастания) средств на счёте, с указанием суммы на счёте и суммы комиссии по счёту за период, процент вывести с точностью до двух десятичных знаков после запятой. При этом каждый счёт должен быть выведен ровно один раз.

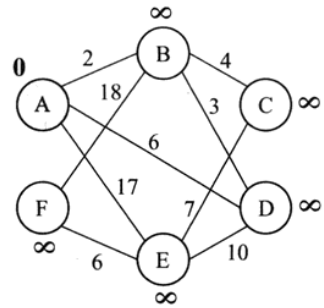
Пример выходных данных для приведённого выше примера входных данных:

```
ZT123  570  30
WQ001  475  25
10567  285  15
```

Ответы и комментарии

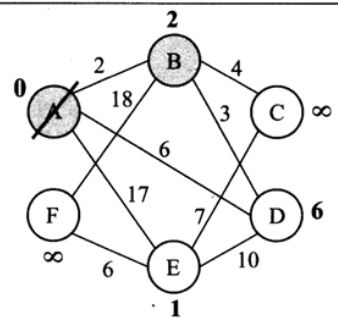
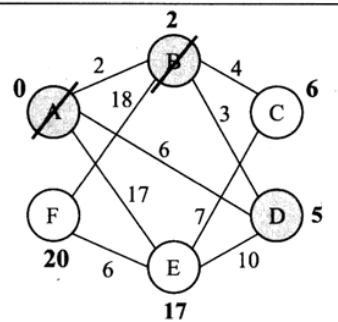
Вариант 2

Часть 1

<table>
<tr><th>Задание</th><th>Номер ответа</th><th>Комментарии</th></tr>
<tr><td>1</td><td>2</td><td>3</td></tr>
<tr><td>A1</td><td>2</td><td>Переведём число 1537 в двоичную систему счисления. Для того чтобы сделать это быстро, выделим максимальные степени двойки: 1537 – 1024 = 513 513 – 512 = 1. Итак, 1537 = 1024 + 512 + 1 = $2^{10} + 2^9 + 2^0$. Десятый, девятый и нулевой разряды заполним единицами, остальные — нулями, получаем: $1537_{10} = 11000000001_2$. Количество значащих нулей равно 8</td></tr>
<tr><td>A2</td><td>3</td><td>
<table>
<tr><th>Шаг</th><th>Метки</th><th>Комментарии</th></tr>
<tr><td>0</td><td> 
</td><td>Начальные метки пунктов. Минимальная метка в A</td></tr>
</table>
</td></tr>
</table>

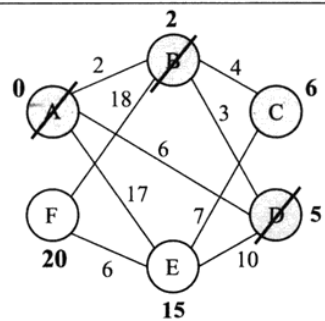
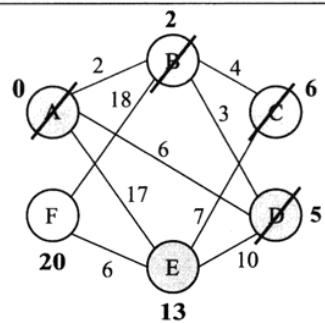
Продолжение таблицы

1	2	3

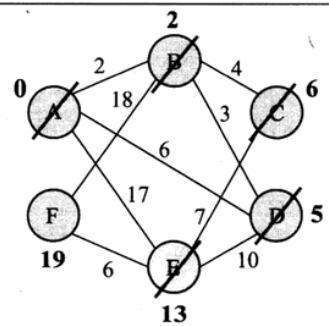
Шаг	Метки	Комментарии
1-1		Для соседей A: AB = 2 AD = 6 AE = 17 Вершина A – посещённая. Минимальная метка – в B
1-2		Для соседей B: AC = 2 + 4 = 6 AD = 2 + 3 = 5; 5 < 6. Вершины A и B – посещённые. Минимальная метка – в D

Продолжение таблицы

1	2	3

Шаг	Метки	Комментарии
1-3		Для соседей D: AE = 5 + 10 = 15; 15 < 17. Вершины A, B, D – посещённые. Минимальная метка – в C
1-4		Для соседей C: AE = 6 + 7 = 13; 13 < 15. Вершины A, B, D, C – посещённые. Минимальная метка – в E

Продолжение таблицы

<table>
<tr><th>1</th><th>2</th><th>3</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>Шаг</th><th>Метки</th><th>Комментарии</th></tr>
<tr><td>1-5</td><td> 
</td><td>Для соседей E:
AF = 13 + 6 = 19; 19 < 20.
Вершины A, B, D, C, E – посещённые.
Минимальная метка – в F, непосещённых вершин нет</td></tr>
</table>
Длина пути 19</td></tr>
<tr><td>A3</td><td>1</td><td>
<table>
<tr><th>X</th><th>Y</th><th>Z</th><th>F</th><th>¬X∨¬Y∨¬Z</th><th>¬X∧¬Y∧¬Z</th><th>X∨Y∨¬Z</th><th>¬X∧Y∧Z</th></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
</table>
Значения выражения варианта ответа № 1 совпадают с **F** на всех заданных наборах</td></tr>
<tr><td>A4</td><td>4</td><td>См. решение варианта 1</td></tr>
</table>

Продолжение таблицы

<table>
<tr><th>1</th><th>2</th><th>3</th></tr>
<tr><td>A5</td><td>4</td><td>Сумма цифр старших разрядов принадлежит отрезку [2, 18], остальных разрядов — отрезку [0, 18]. Итоговое число состоит из трёх частей. Из условия задачи следует, что вторая часть числа должна быть больше первой. Этим условиям удовлетворяет число 11146 (разобьём его на три части, каждая из которых принадлежит отрезку [0, 18]: 11, 14 и 6, где 11 < 14)</td></tr>
<tr><td>A6</td><td>3</td><td>По таблице 1 определим ID Ивановой А.И. – 23.
В таблице 2 найдём строки, в которых ID_Ребёнка = 23, и определим соответствующие ID_Родителя – это значения 13 и 85.
<table>
<tr><th>ID_Родителя</th><th>ID_Ребёнка</th></tr>
<tr><td>13</td><td>23</td></tr>
<tr><td>85</td><td>23</td></tr>
</table>
Затем найдём в таблице 2 строки с ID_Ребёнка 13 и 85 и определим ID_Родителя. Для ID_Ребёнка 13 — это значения 82 и 95. Для ID_Ребенка 85 в таблице 2 не приводятся данные о родителях.
<table>
<tr><th>ID_Родителя</th><th>ID_Ребёнка</th></tr>
<tr><td>82</td><td>13</td></tr>
<tr><td>95</td><td>13</td></tr>
</table></td></tr>
</table>

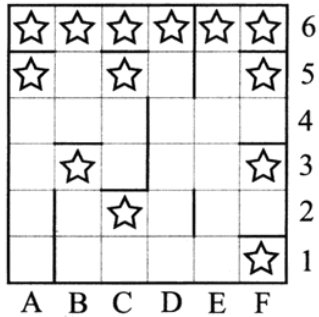
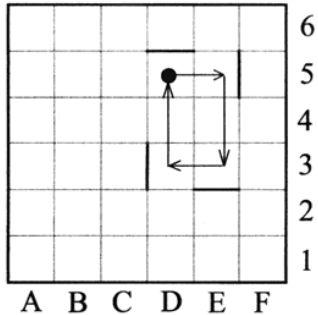
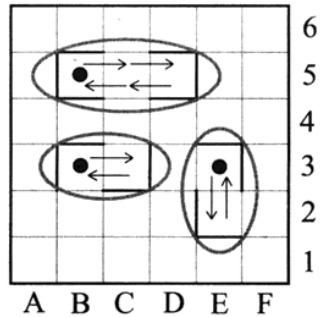
<table>
<tr><th>1</th><th>2</th><th>3</th></tr>
<tr><td></td><td></td><td>В таблице 1 значению ID = 82 соответствует мужчина Черных А.Н., это дедушка Ивановой А.И. Значению ID = 95 соответствует женщина Логинова Т.Н. – бабушка Ивановой А.И.
<table>
<tr><th>ID</th><th>Фамилия_И.О.</th><th>Пол</th></tr>
<tr><td>82</td><td>Черных А.Н.</td><td>М</td></tr>
<tr><td>95</td><td>Логинова Т.Н.</td><td>Ж</td></tr>
</table></td></tr>
<tr><td>А7</td><td>2</td><td>Следует оценить отношение общего количества иностранных автомобилей к общему количеству машин для каждого мастера</td></tr>
<tr><td>А8</td><td>1</td><td>Для записи сигнала, имеющего $N = 32$ возможных значения (уровня дискретизации), требуется
$I = \log_2 N = \log_2 32 = 5$ бит.
Записывается 64 сигнала в секунду, продолжительность записи $4 \cdot 60 + 16 = 256$ с. Учитывая, что в одном килобайте 2^{13} бит, определим размер файла в килобайтах:
$64 \cdot 256 \cdot 5 / 2^{13} = 2^6 \cdot 2^8 \cdot 5 / 2^{13} = 10$ Кбайт</td></tr>
</table>

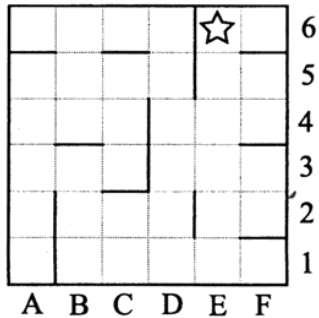
Продолжение таблицы

<table>
<tr><th>1</th><th>2</th><th>3</th></tr>
<tr><td>A9</td><td>1</td><td>Проверим, являются ли предложенные ответы началом заданных кодовых слов или заданные коды началом кодов, предложенных в ответах.
1) 00 – не является началом ни одного из заданных кодов символов, ни один из заданных кодов не является началом этого кода.
2) 0101 – началом этого кода является код символа Г (010).
3) 011 – не является началом ни одного из заданных кодов символов, ни один из заданных кодов не является началом этого кода.
4) 10 – является началом кодов символов Б (100) и В (101).
Кодом буквы Д может быть 00 и 011. Кодовое слово минимальной длины 00</td></tr>
<tr><td>A10</td><td>4</td><td>Введём обозначения логических высказываний:
A – первая буква согласная;
B – последняя буква согласная;
C – больше 6 букв.
Получим выражение
$\neg((A \wedge B) \rightarrow C) = \neg(\neg(A \wedge B) \vee C) = A \wedge B \wedge \neg C$
Построим таблицу истинности:
<table>
<tr><th></th><th>A</th><th>B</th><th>C</th><th>¬C</th><th>A ∧ B ∧¬C</th></tr>
<tr><td>ЛОПАТА</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>СЕКАТОР</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>ГРАБЛИ</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>ШЛАНГ</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td></tr>
</table>
Правильный ответ — ШЛАНГ</td></tr>
</table>

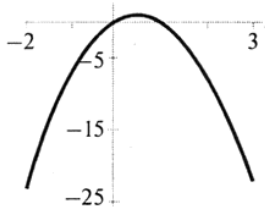
<table>
<tr><th>1</th><th>2</th><th>3</th></tr>
<tr><td>A11</td><td>4</td><td>Всего символов 28 · 2 + 10 = 66. На один символ потребуется
$\lceil \log_2 66 \rceil = 7$ бит,
тогда на 15 символов потребуется 15 · 7 = 105 бит. Минимально возможное количество байтов на 15 символов: 105 : 8 = 13,1... Округляя до ближайшего большего целого, получим 14 байт. Под хранение 20 паролей потребуется: 14 · 20 = 280 байт</td></tr>
<tr><td>A12</td><td>1</td><td>В задаче используются два оператора цикла с параметром. В первом цикле массив A заполняется числами от 0 до 10:
<table>
<tr><td>i</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td></tr>
<tr><td>A[i]</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td></tr>
</table>
Во втором цикле параметр цикла i изменяется с шагом 2 и принимает значения 0, 2, 4, 6, 8. В теле цикла меняются местами значения переменных A[i] и A[i+2], т.е. значения A[0] и A[2], A[2] и A[4], A[4] и A[6], A[6] и A[8], A[8] и A[10].
Значения A[2]=2, A[3]=3, A[5]=5, A[7]=7, A[9]=9 остались без изменений. В результате получим:
<table>
<tr><td>i</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td></tr>
<tr><td>A[i]</td><td>2</td><td>1</td><td>4</td><td>3</td><td>6</td><td>5</td><td>8</td><td>7</td><td>10</td><td>9</td><td>0</td></tr>
</table></td></tr>
</table>

Продолжение таблицы

1	2	3
A13	1	Максимальное количество клеток, требующих проверки:  Траектории движения РОБОТА:  

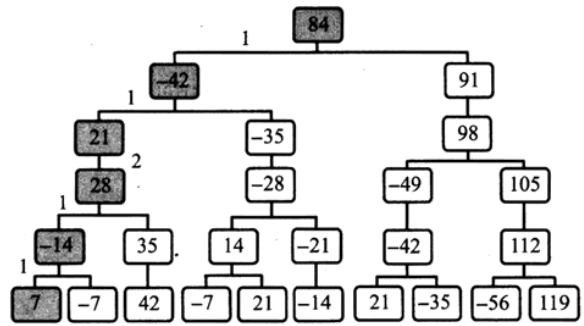
1	2	3
		Результат проверки клеток: 
A14	2	Переменные a, b, d используются для организации цикла. В программе используется функция пользователя $F(x)$, которая возвращает значение выражения $4*x*(1 - x)$. Значение аргумента в теле функции не изменяется. В цикле вычисляется максимальное значение функции $F(t)$ на интервале $[a, b]$ с шагом d. В переменной R сохраняется значение функции $F(t)$, претендующее на максимальное. В переменной M сохраняется значение аргумента t, при котором функция принимает максимальное значение. Таким образом, в программе определяется значение аргумента функции F, при котором функция принимает максимальное значение, на отрезке $[-2; 2]$. Запишем функцию $F(x) = 4*x*(1 - x) = -4*x^2 + 4*x$.

Окончание таблицы

1	2	3
		Получили уравнение параболы ($px^2 + qx$), ветви которой направлены вниз ($p < 0$), а вершина находится в точке $x_в = -q/2p = -4/(-8) = 1/2$, $x_в \in [-2;2]$.  В вершине параболы её значение максимально. Таким образом, на печать будет выведено число 0,5

Часть 2

Задание	Ответ	Комментарии
1	2	3
B1	640	Сообщение состоит из 80 символов. Первоначально оно имело объём 80 · 1 байт, а после перекодирования 80 · 2 байт. Следовательно, объём сообщения увеличился на 80 · 2 – 80 = 80 байт = 640 бит

1	2	3
B2	11211	Не все числа делятся на 2 без остатка, следовательно, количество ветвей и узлов обратного дерева решения будет меньше, чем прямого. Построим дерево решения с использованием обратных команд 1. делить на -2 2. прибавить 7  Решение выделено цветом. Последовательность команд запишем от листа 7 к корню 84

Продолжение таблицы

<table>
<tr><th>1</th><th>2</th><th>3</th></tr>
<tr><td>B3</td><td>48</td><td>
<table>
<tr><th>Команда или условие (логическое выражение)</th><th>a</th><th>b</th></tr>
<tr><td>a := 56</td><td>56</td><td></td></tr>
<tr><td>b := a - 70</td><td></td><td>–14</td></tr>
<tr><td>не(b > 0) = да</td><td></td><td></td></tr>
<tr><td>b := b + 13</td><td></td><td>–1</td></tr>
<tr><td>не(b > 0) = да</td><td></td><td></td></tr>
<tr><td>b := b + 13</td><td></td><td>12</td></tr>
<tr><td>не(b > 0) = нет</td><td></td><td></td></tr>
<tr><td>(mod(a,b) <> 0) = да</td><td></td><td></td></tr>
<tr><td>a := a - 4</td><td>52</td><td></td></tr>
<tr><td>(mod(a,b) <> 0) = да</td><td></td><td></td></tr>
<tr><td>a := a - 4</td><td>48</td><td></td></tr>
<tr><td>(mod(a,b) <> 0) = нет</td><td></td><td></td></tr>
<tr><td>Результат</td><td>48</td><td>12</td></tr>
</table>
</td></tr>
<tr><td>B4</td><td>УУУОУ</td><td>Для записи слов используются три символа. Эти символы можно заменить цифрами 0, 1 и 2 соответственно (А=0, О=1, У=2). При этом список слов соответствует значениям неотрицательных целых чисел в троичной системе счисления:</td></tr>
</table>

<table>
<tr><th>1</th><th>2</th><th>3</th></tr>
<tr><td></td><td></td><td>1. $00000_3 = 0_{10}$
2. $00001_3 = 1_{10}$
3. $00002_3 = 2_{10}$
4. $00010_3 = 3_{10}$
На 240-м месте находится число 239_{10}, так как числа начинаются с нуля. Переведём его в троичную систему счисления (делением на 3 – основание системы счисления):
<table>
<tr><th>Делимое</th><th>Целая часть частного</th><th>Остаток от деления</th></tr>
<tr><td>239</td><td>79</td><td>2</td></tr>
<tr><td>79</td><td>26</td><td>1</td></tr>
<tr><td>26</td><td>8</td><td>2</td></tr>
<tr><td>8</td><td>2</td><td>2</td></tr>
<tr><td>2</td><td>0 останов</td><td>2</td></tr>
</table>
Результат – запись остатков от деления — равен 22212, что соответствует записи слова с использованием заданных букв — УУУОУ</td></tr>
<tr><td>B5</td><td>5</td><td>Проведём возможные вычисления в ячейках таблицы, получим
<table>
<tr><th></th><th>A</th><th>B</th><th>C</th></tr>
<tr><td>1</td><td>4</td><td>2</td><td>4</td></tr>
<tr><td>2</td><td>2</td><td>3</td><td></td></tr>
<tr><td>3</td><td>6</td><td>=3+C2</td><td>=(1+C2)/2</td></tr>
</table></td></tr>
</table>

Продолжение таблицы

<table>
<tr><th>1</th><th>2</th><th>3</th></tr>
<tr><td></td><td></td><td>Из анализа диаграммы следует, что каждая группа столбцов на диаграмме соответствует значениям столбцов таблицы. По рисунку также можно определить, что C3 = B2 = 3. Тогда (1+C2)/2 = 3, и в ячейке C2 должно находиться число 5. Тогда B3 = 8, что соответствует диаграмме</td></tr>
<tr><td>B6</td><td>49</td><td>Построим трассировочную таблицу первого вида:
<table>
<tr><th>№</th><th>Команда или условие (логическое выражение)</th><th>Вычисление правой части команды присваивания или условия (логического выражения)</th><th>a</th><th>b</th><th>c</th></tr>
<tr><td>1</td><td>a = 35</td><td>35</td><td>35</td><td></td><td></td></tr>
<tr><td>2</td><td>b = a - a mod 21</td><td>35 - 35 mod 21 =
= 35 - 14 = 21</td><td></td><td>21</td><td></td></tr>
<tr><td>3</td><td>a - b >= 14</td><td>(14 >=14) = да</td><td></td><td></td><td></td></tr>
<tr><td>4</td><td>c = 2 * a - b</td><td>2*35-21 = 49</td><td></td><td></td><td>49</td></tr>
<tr><td colspan="3">Результат</td><td>35</td><td>21</td><td>49</td></tr>
</table></td></tr>
<tr><td>B7</td><td>64</td><td>Шаг 1. Исходные значения элементов массива R:
<table>
<tr><th>R[0]</th><th>R[1]</th><th>R[2]</th><th>R[3]</th><th>R[4]</th></tr>
<tr><td>2</td><td>2</td><td>1</td><td>3</td><td>2</td></tr>
</table></td></tr>
</table>

1	2	3
		Шаг 2. Трассировка вызова функции: `ArrProcessing_1()`:

i	R[0]	R[1]	R[2]	R[3]	R[4]	p
	2	2	1	3	2	1
0		R[0] * *p = = 2*1 = 2				i+1=0+1=1
1			R[1] * *p = = 2 * 1 = 2			i+1 = 1 + 1 = 2
2				R[2] * *p = = 2 * 2 = 4		i+1 = 2 + 1 = 3
3					R[3] * *p = = 4 * 3 = 12	i+1 = 3 + 1 = 4
Итог:	**2**	**2**	**2**	**4**	**12**	**4**

Продолжение таблицы

<table>
<tr><th>1</th><th>2</th><th>3</th></tr>
<tr><td></td><td></td><td>

Шаг 3. Трассировка вызова функции: `MultCounter()`:

<table>
<tr><th>i</th><th>T</th><th>N</th></tr>
<tr><td></td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>R[0] = 2</td><td>N+T*R[1] = 0+2*2=4</td></tr>
<tr><td>2</td><td>R[1] = 2</td><td>N+T*R[2] = 4 + 2*2 = 8</td></tr>
<tr><td>3</td><td>R[2] = 2</td><td>N+T*R[3] = 8 + 2*4 = 16</td></tr>
<tr><td>4</td><td>R[3] = 4</td><td>N+T*R[4] = 16 + 4*12 = 64</td></tr>
<tr><td>Итог:</td><td>4</td><td>64</td></tr>
</table>

Из последней таблицы видно, что в переменную N вызывающей программы попадёт значение 64, именно оно будет выведено на экран

</td></tr>
<tr><td>B8</td><td>3</td><td>

Запишем число в N-ичной системе счисления $XYZ0_N$, где X, Y, Z – цифры, которые могут принимать значения от 0 до N–1.

Развёрнутая запись числа имеет вид:

$$X \cdot N^3 + Y \cdot N^2 + Z \cdot N = 45 \text{ или}$$

$$N \cdot (X \cdot N^2 + Y \cdot N + Z) = 3 \cdot 3 \cdot 5.$$

N может принимать значения 3, 5, 9 (3 · 3), 15 (3 · 5).

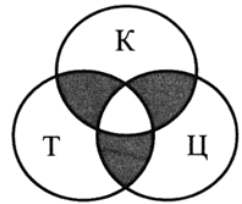
Переведём 45 в троичную систему счисления, получим 1200_3, что соответствует условию задачи

</td></tr>
</table>

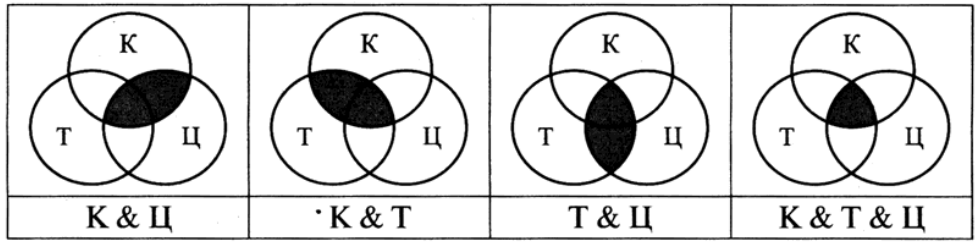
Продолжение таблицы

1	2	3
B9	11	$F(А) = F(Б) + F(В) + F(Г) + F(Ж) = \lvert F(Б) = F(Ж) \rvert =$ $= F(В) + F(Г) + 2 \cdot F(Ж) = \lvert F(В) = F(Е) + F(З) + F(Д) \rvert =$ $= F(Г) + F(Д) + F(Е) + 2 \cdot F(Ж) + F(З) = \lvert F(Г) = F(Д) \rvert =$ $= 2 \cdot F(Д) + F(Е) + 2 \cdot F(Ж) + F(З) = \lvert F(Д) = F(З) \rvert =$ $= F(Е) + 2 \cdot F(Ж) + 3 \cdot F(З) = \lvert F(Ж) = F(И) + F(Е) \rvert =$ $= 3 \cdot F(Е) + 3 \cdot F(З) + 2 \cdot F(И) = \lvert F(Е) = F(И) + F(З) \rvert =$ $= 6 \cdot F(З) + 5 \cdot F(И) = \lvert F(З) = F(И) \rvert =$ $= 11 \cdot F(И) = 11$
B10	34816	Объём полученной (переданной) информации вычисляется по формуле: V [бит] $= v$ [бит/с]$\cdot t$ [с], где v – скорость, а t – время передачи информации. Для нахождения общего времени передачи данных от Васи к Мише составим уравнение. Пусть t_1 – время, которое необходимо компьютеру Васи, чтобы скачать информацию объёмом $V_1 = 256$ Кб со скоростью $v_1 = 2^{10}$ бит/с, после чего начать ретрансляцию Мише; t_2 – время, которое необходимо компьютеру Миши, чтобы скачать информацию объёмом $V_2 = 1$ Мб со скоростью $v_2 = 2^8$ бит/с с компьютера Васи. Тогда $t_{общ} = t_1 + t_2 = \frac{V_1}{v_1} + \frac{V_2}{v_2}$. Выразим все заданные в задаче объёмы информации в битах: 256 Кб $= 2^{21}$ бит 1 Мб $= 2^{23}$ бит

Продолжение таблицы

1	2	3
		Вычислим время: $t_1 = 2^{21}$ бит / 2^{10} бит/с = 2^{11} с = 2048 с $t_2 = 2^{23}$ бит / 2^8 бит/с = 2^{15}с = 32768 с $t_{общ}$ = 34816 с
B11	CEDG	01100110 & 11111000 ——————— 01100000 01100000_2=96 IP-адрес сети: 33.153.96.0
B12	4450	Введём обозначения: К – множество страниц, полученных по запросу ***Кино***, Т – множество страниц, полученных по запросу ***Театр***, Ц – множество страниц, полученных по запросу ***Цирк***. |X| – мощность множества X, например: |K| – это количество страниц, найденных по запросу ***Кино***. Чтобы получить число страниц, которые отобразятся в результате запроса ***((Кино & Театр) | (Кино & Цирк) | (Театр & Цирк)) & – (Кино & Театр & Цирк)***, используем круги Эйлера, каждый из которых будет соответствовать множеству (К, Т и Ц). Результатом требуемого запроса будет являться закрашенная область. 

Продолжение таблицы

1	2	3
		Просуммируем количество страниц, найденных по запросам ***Кино & Цирк***, ***Кино & Театр*** и ***Театр & Цирк***. Следует учесть, что пересечение всех трёх множеств было просуммировано трижды, поэтому отнимем от полученного числа утроенное количество страниц, полученных по запросу ***Кино & Театр & Цирк***.  Результат определяется по формуле: $\lvert К \,\&\, Ц\rvert + \lvert К \,\&\, Т\rvert + \lvert Т \,\&\, Ц\rvert - 3 \cdot \lvert К \,\&\, Т \,\&\, Ц\rvert$
B13	13	Пусть x – количество команд 1 в программе; $0 \le x \le 12$, тогда количество команд 2 равно $(12 - x)$. В результате выполнения какой-либо программы из 12 команд будет получено число $k = 5 + x - 2 \cdot (12 - x) = 3 \cdot x - 19$. x может принимать 13 разных значений (0, 1, 2, ..., 12), следовательно, в результате выполнения программ, состоящих из 12 команд, будет получено 13 разных чисел

Продолжение таблицы

1	2	3
B14	364	Система уравнений: $a \to b \vee c\bar{d} = 1$ $c \to d \vee e\bar{f} = 1$ $e \to f \vee g\bar{h} = 1$ $g \to h \vee i\,\bar{j} = 1$ Заметим, что $x \to y = \bar{x} \vee y$ $\overline{\bar{x} \vee y} = x\bar{y}$ Введём обозначения: $K = a \to b$ $\quad \bar{K} = a\bar{b}$ $L = c \to d$ $\quad \bar{L} = c\bar{d}$ $M = e \to f$ $\quad \bar{M} = e\bar{f}$ $N = g \to h$ $\quad \bar{N} = g\bar{h}$ $O = i \to j$ $\quad \bar{O} = i\,\bar{j}$

1	2	3

Система уравнений примет вид:

$K \vee \bar{L} = 1$

$L \vee \bar{M} = 1$

$M \vee \bar{N} = 1$

$N \vee \bar{O} = 1$

Для того чтобы все 4 выражения были одновременно истинны, достаточно объединить их в одно операцией конъюнкции. Получим:

$(K \vee \bar{L})(L \vee \bar{M})(M \vee \bar{N})(N \vee \bar{O}) = 1.$

Решим полученное уравнение.

Способ 1. Таблица истинности полученного выражения полностью совпадает с таблицей, построенной при решении задания B10 в варианте 1. Приведём неполную таблицу истинности

№ набора	K	L	M	N	O	$K \vee \neg L$	$L \vee \neg M$	$M \vee \neg N$	$N \vee \neg O$	Результат
0	0	0	0	0	0	1	1	1	1	1
16	1	0	0	0	0	1	1	1	1	1
24	1	1	0	0	0	1	1	1	1	1
28	1	1	1	0	0	1	1	1	1	1
30	1	1	1	1	0	1	1	1	1	1
31	1	1	1	1	1	1	1	1	1	1

Окончание таблицы

1	2	3
		Способ 2. Выполним преобразования, аналогичные решению задания B10 варианта 1, получим $KLMNO \vee KLMN\bar{O} \vee KLM\bar{N}\bar{O} \vee KL\bar{M}\bar{N}\bar{O} \vee K\bar{L}\bar{M}\bar{N}\bar{O} \vee \bar{K}\bar{L}\bar{M}\bar{N}\bar{O}.$ Все переменные таблицы истинности (и сомножители каждого слагаемого в СДНФ) взаимно независимы. Каждая из пяти переменных K, L, M, N, O в соответствии с таблицей истинности операции импликации принимает значение «истина» при трёх наборах переменных, а их отрицания – при одном наборе исходных переменных. Таким образом, количество наборов исходных переменных определяется как $3 \cdot 3 \cdot 3 \cdot 3 \cdot 3 + 3 \cdot 3 \cdot 3 \cdot 3 \cdot 1 + 3 \cdot 3 \cdot 3 \cdot 1 \cdot 1 + 3 \cdot 3 \cdot 1 \cdot 1 \cdot 1 + 3 \cdot 1 \cdot 1 \cdot 1 \cdot 1 + 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 = 243 + 81 + 27 + 9 + 3 + 1 = 364$

Часть 3

Задание	Комментарии
1	2
C1	Пример: $x = -3$, $y = -3$. При ложном значении условия в первом операторе ветвления программа не выводит никакого сообщения. Возможная доработка – заменить все операторы ветвления программы на предложенные:

Паскаль

```
if (y<=x) and (x<=0) or (y<=x) and (y<=-x*x+2) then
  writeln('принадлежит')
  else
  writeln('не принадлежит ');
```

Бейсик

```
IF y<=x AND x<=0 OR y<=x AND y<=-x*x+2 THEN
  PRINT "принадлежит"
  ELSE
  PRINT "не принадлежит"
ENDIF
```

Си

```
if (((y<=x)&&(x<=0))|| ((y<=x)&&(y<=-x*x+2)))
  printf("принадлежит");
  else
 printf("не принадлежит");
```

Продолжение таблицы

1	2

C2	**Паскаль**
	{Индексы максимального и минимального элементов}
	nmax:=1;
	nmin:=1;
	for i:=2 to n do
	begin
	if (a[i] > a[nmax]) then nmax:= i;
	{Нашли большее значение. Запоминаем номер элемента}
	if (a[i] < a[nmin]) then nmin:= i;
	{Нашли меньшее значение. Запоминаем номер элемента}
	end;
	{Меняем местами элементы}
	max := a[nmax];
	a[nmax]:= a[nmin];
	a[nmin]:= max;
	for i:=1 to n do write(' ', a[i]);
	writeln;
	Бейсик
	'Индексы максимального и минимального элементов
	NMIN = 1
	NMAX = 1
	FOR I = 2 TO N
	IF A(I) > A(NMAX) THEN NMAX = I

Продолжение таблицы

1	2

```
    'Нашли большее значение. Запоминаем номер элемента
  IF A(I) < A(NMIN) THEN NMIN = I
    'Нашли меньшее значение. Запоминаем номер элемента
  NEXT I
'Меняем местами элементы
MAX = A(NMAX)
A(NMAX) = A(NMIN)
A(NMIN) = MAX
FOR I = 1 TO N
    PRINT A(I), ' '
NEXT I
```

Си

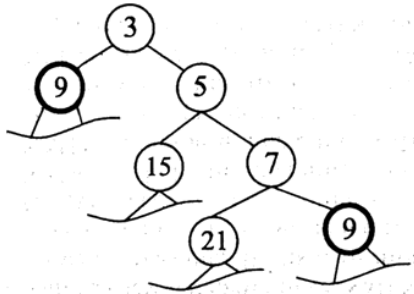
```
   // Индексы максимального и минимального элементов
   nmax = 0;
   nmin = 0;
   //Просмотр элементов массива
   for(i = 1; i < N; i++)
   {
       if(a[i] > a[nmax]) nmax = i;
//Нашли большее значение. Запоминаем номер элемента
     if(a[i] < a[nmin]) nmin = i;
//Нашли меньшее значение. Запоминаем номер элемента
   }
```

Продолжение таблицы

1	2

	`//Меняем местами элементы`
	`max = a[nmax];`
	`a[nmax] = a[nmin];`
	`a[nmin] = max;`
	`//Выводим массив после перестановки`
	`for(i = 0; i < N; i++) printf("%d",a[i]);`
	Естественный язык
	Записываем в переменные NMIN и NMAX начальное значение, равное 1 – индекс первого элемента массива. В цикле от второго до последнего элемента последовательно выполняем два сравнения. Если значение текущего элемента превышает значение элемента с индексом NMAX, заменяем значение позиции максимума NMAX на индекс текущего элемента NMAX = I. Если значение текущего элемента меньше значения элемента с индексом NMIN, заменяем значение NMIN на значение индекса текущего элемента NMIN = I. После завершения цикла меняем местами элементы с максимальным и минимальным значениями. Для этого в переменной MAX сохраняем значение элемента с индексом NMAX, элементу с индексом NMAX присваиваем значение элемента с индексом NMIN, элементу с индексом NMIN присваиваем значение переменной MAX.
	Выводим элементы массива.

C3	Пусть $F(n)$ – количество программ получения числа n из числа 3. Составим рекуррентные соотношения для вычисления значений функции по предыдущим значениям аргумента. После выполнения первой команды программы получим либо $3 + 2 = 5$, либо $3 \cdot 3 = 9$.

1	2
	Число 5 невозможно получить другой программой (последовательностью команд). Число 9 можно получить и второй программой (3 + 2 + 2 + 2). На рисунке показано начало дерева выполнения команд от исходного числа 3, левая ветвь соответствует команде «умножь на 3», правая – команде «прибавь 2».  Тогда $F(5) = 1$; $F(9) = 2$. Для $n = 3k$ (аргументов, кратных трём) функция равна: $F(3k) = F(k) + F(3k - 2)$. Для $n \neq 3k$ (аргументов, не кратных трём) можно записать $F(n) = F(n - 2)$.

Продолжение таблицы

1	2
	Итак, вычисления будем выполнять по рекуррентным формулам: $F(n) = F(n/3) + F(n-2)$ при $n = 3k,\ k > 3$; $F(n) = F(n-2)$ при $n \neq 3k,\ k > 3$; $F(5) = 1$; $F(9) = 2$; $F(n) = 0$ при $n < 5$. Определим количество программ получения числа 35 из числа 3, слагаемые будем записывать в порядке возрастания аргументов: $F(35) = F(33) = \lvert F(33) = F(31) = F(29) = F(27) \rvert =$ $= F(27) + F(11) =$ $= \lvert F(27) = F(25) = F(23) = F(21) \rvert =$ $= F(21) + 2 \cdot F(9) =$ $= F(15) + F(7) + 4 =$ $= F(9) + 2 \cdot F(5) + 4 = 2 + 2 + 4 = 8.$ Ответ: 8

Продолжение таблицы

1	2
C4	В программе используется одномерный массив, состоящий из 12 элементов, по количеству участников конкурса. Каждый элемент массива имеет тип «запись» (thea) и состоит из двух полей: строкового для хранения имени города name и числового для числа голосов count. 1. Программа считывает в переменную N количество голосов, которые ей предстоит обработать. 2. В переменную M записывается 0 – текущее количество городов в массиве. 3. В цикле от 1 до N обрабатываются строки входных данных, соответствующие городам, за которые отданы голоса. 3.1. В переменную строкового типа считывается очередная строка входных данных. 3.2. Во внутреннем цикле просматриваются элементы массива до тех пор, пока не встретится такая же строка в поле name и переменная цикла j меньше или равна M. 3.2.1. Если обнаружена такая же строка в поле name, то значение поля count соответствующего элемента массива увеличивается на 1 и выполнение внутреннего цикла прерывается. 3.2.2. Если переменная цикла j стала больше M, значение M увеличивается на 1, в поле name M-го элемента массива записывается строка, в поле count записывается 1. 4. Проводится сортировка полученных данных в порядке возрастания значения поля count по алгоритму сортировки «пузырьком». 5. Данные выводятся на экран.

Продолжение таблицы

1	2

	Код программы
	
	{ тип «запись» для обработки результатов голосования }
	type thea=record
	name:string[100]; { название города }
	count:integer; { число голосов }
	end;
	var
	N,i,j,M:integer;
	city:array[1..12]of thea; { массив записей типа thea }
	t:thea; { вспомогательная переменная }
	s:string[100]; { вводимая строка }
	begin
	for i:=1 to 12 do begin
	city[i].name:='';
	city[i].count:=0;
	end;
	readln(N); { считываем количество запросов }
	M:=0;
	{ цикл чтения строк с входными данными }
	for i:=1 to N do begin
	readln(s);
	j:=1;
	{ ищем город в массиве записей }
	while((j<M)and(city[j].name<>s))do

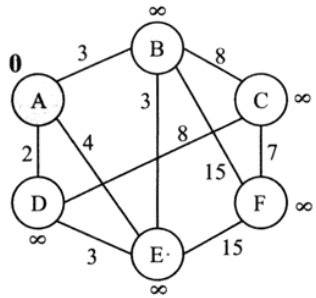
Окончание таблицы

1	2

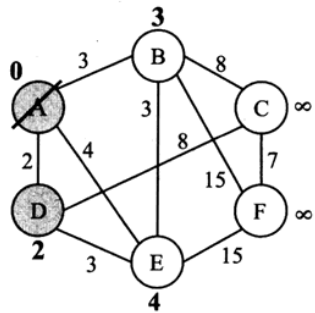
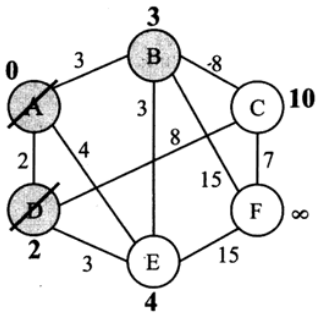
	`inc(j,1);`
	`if(city[j].name<>s)then begin`
	`{ города не найден - добавляем элемент в массив}`
	`inc(M,1);`
	`city[M].name:=s;`
	`city[M].count:=1;`
	`end`
	`else`
	`{ город найден в массиве, увеличиваем количество голосов }`
	`begin`
	`inc(city[j].count,1);`
	`end;`
	`end;`
	`{ сортируем записи в порядке возрастания }`
	`for i:=1 to M do`
	`for j:=1 to M-i do`
	`if(city[j+1].count< city[j].count) then begin`
	`t:= city[j];`
	`city[j]:= city[j+1];`
	`city[j+1]:=t;`
	`end;`
	`{ выводим список городов }`
	`for i:=1 to M do`
	`writeln(city[i].name,' ',city[i].count);`
	`end.`

Вариант 3

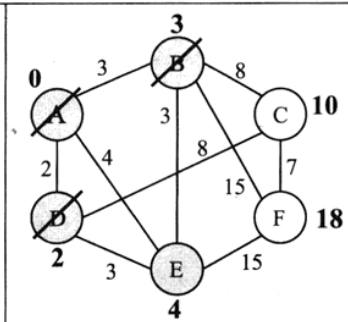
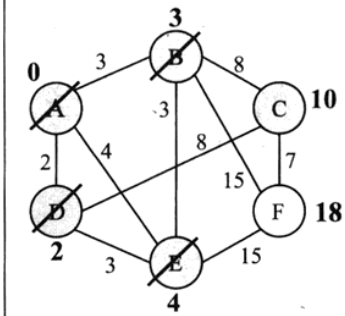
Часть 1

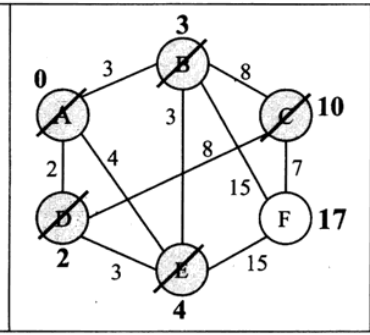
<table>
<tr><th>Задание</th><th>Номер ответа</th><th>Комментарии</th></tr>
<tr><td>1</td><td>2</td><td>3</td></tr>
<tr><td>A1</td><td>3</td><td>Переведём число 1088 в двоичную систему счисления. Выделим максимальные степени двойки:
$$1088 = 1024 + 64 = 2^{10} + 2^{6}.$$
Десятый и шестой разряды заполним единицами, остальные – нулями, получаем: $108810 = 10001000000$. Число значащих нулей равно 9</td></tr>
<tr><td>A2</td><td>2</td><td>
<table>
<tr><th>Шаг</th><th>Метки</th><th>Комментарии</th></tr>
<tr><td>0</td><td> 
</td><td>Начальные метки пунктов.
Минимальная метка в A</td></tr>
</table>
</td></tr>
</table>

Продолжение таблицы

1	2	3
		1-1  Для соседей A: AB = 3 AD = 2 AE = 4 Вершина A – посещённая. Минимальная метка – в D
		1-2  Для соседей D: AC = 2 + 8 = 10 AE = 2 + 3 = 5; 5 > 4. Вершины A и D – посещённые. Минимальная метка – в B

Продолжение таблицы

1	2	3
		1-3  Для соседей B: AE = 3 + 3 = 6; 6 > 4 AC = 3 + 8 = 11; 11 > 10; AF = 3 + 15 = 18. Вершины A, B, D – посещённые. Минимальная метка – в E
		1-4  Для соседей E: AF = 4 + 15 = 19; 19 > 18. Вершины A, B, D, E – посещённые. Минимальная метка – в C

1	2	3
		1-5  Для соседей C: AF = 10 + 7 = 17; 17 < 18. Вершины A, B, C, D, E – посещённые. Минимальная метка – в F, непосещённых вершин нет Длина пути 17
A3	4	Построим таблицы истинности для всех вариантов ответов

X	Y	Z	F	¬X ∨ ¬Y ∨ Z	X ∧ ¬Y ∧ ¬Z	¬X ∨ Y ∨ ¬Z	X ∨ ¬Y ∨ ¬Z
0	1	1	**0**	1	0	1	0
1	0	1	**1**	1	0	0	1
1	1	0	**1**	0	0	1	1

Значения выражения ответа № 4 совпадают с **F** на всех заданных наборах

A4	2	См. решение варианта 1

Продолжение таблицы

<table>
<tr><th>1</th><th>2</th><th>3</th></tr>
<tr><td>A5</td><td>1</td><td>Сумма цифр старших разрядов принадлежит отрезку [2, 18], остальных разрядов – отрезку [0, 18]. Итоговое число состоит из трёх частей. Вторая часть числа должна быть меньше первой. Этим условиям удовлетворяет число 1112 (можем разбить его на 3 части, принадлежащие отрезку [0, 18]: 11, 1 и 2, где 11 > 1)</td></tr>
<tr><td>A6</td><td>2</td><td>Выясняем по таблице «Предметы» ID_предмета для математики и английского языка:
<table>
<tr><th>ID_предмета</th><th>Наименование</th></tr>
<tr><td>1</td><td>Математика</td></tr>
<tr><td>5</td><td>Английский язык</td></tr>
</table>
Затем по таблице «Оценки» находим ID_ученика, у которых по математике или английскому языку больше 70 баллов:
<table>
<tr><th>ID_ученика</th><th>ID_предмета</th><th>Баллы</th></tr>
<tr><td>10146</td><td>5</td><td>96</td></tr>
<tr><td>10161</td><td>1</td><td>98</td></tr>
<tr><td>10128</td><td>1</td><td>77</td></tr>
<tr><td>10157</td><td>1</td><td>81</td></tr>
</table>
Затем по таблице «Ученики» смотрим, кто из выбранных учеников учится в 11-м классе:</td></tr>
</table>

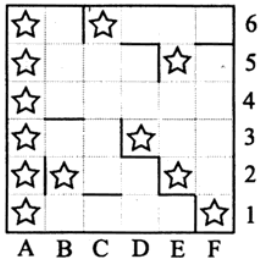
Продолжение таблицы

<table>
<tr><th>1</th><th>2</th><th>3</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>ID_ученика</th><th>Фамилия</th><th>Имя</th><th>Класс</th></tr>
<tr><td>10146</td><td>Иванов</td><td>Денис</td><td>11</td></tr>
<tr><td>10128</td><td>Фёдорова</td><td>Анастасия</td><td>9</td></tr>
<tr><td>10157</td><td>Сидоров</td><td>Илья</td><td>11</td></tr>
<tr><td>10161</td><td>Смирнов</td><td>Пётр</td><td>9</td></tr>
</table>
Таких учеников двое</td></tr>
<tr><td>A7</td><td>1</td><td>Следует оценить отношение количества затраченной ткани к общему количеству пошитых игрушек для каждого предприятия</td></tr>
<tr><td>A8</td><td>2</td><td>Для записи сигнала, имеющего $N = 256$ возможных значений (уровня дискретизации), требуется
$I = \log_2 N = \log_2 256 = 8$ бит = 1 байт.
Записывается 16000 сигналов в секунду, продолжительность записи $2 \cdot 60 + 8 = 128$ с. Учитывая, что в одном килобайте 2^{10} байт, определим размер файла в килобайтах:
$2^7 \cdot 125 \cdot 2^7 / 2^{10} = 125 \cdot 16 = 2000$ Кбайт</td></tr>
<tr><td>A9</td><td>3</td><td>Проверим, являются ли предложенные ответы началом заданных кодовых слов или заданные коды началом кодов, предложенных в ответах.
1) 00 – является началом кода символа Г (000).</td></tr>
</table>

Продолжение таблицы

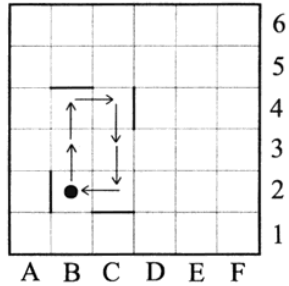
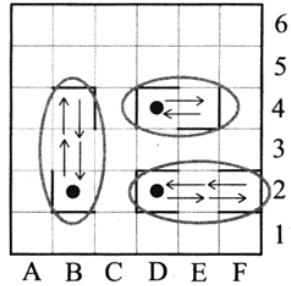
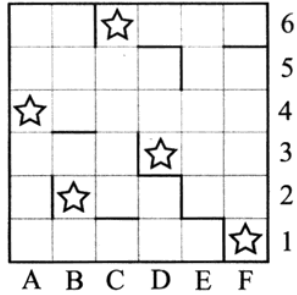
<table>
<tr><th>1</th><th>2</th><th>3</th></tr>
<tr><td></td><td></td><td>2) 011 – не является началом ни одного из заданных кодов символов, ни один из заданных кодов не является началом этого кода.
3) 01 – не является началом ни одного из заданных кодов символов, ни один из заданных кодов не является началом этого кода.
4) 101 – началом этого кода является код символа А (10).
Кодом буквы Д может быть 01 и 011. Кодовое слово минимальной длины 01</td></tr>
<tr><td>A10</td><td>1</td><td>Введём обозначения: А – первая буква гласная, В – четвёртая буква согласная.
Тогда мы получаем выражение $\neg (A \oplus B)$
Проще всего решить данную задачу с помощью построения таблицы истинности:
<table>
<tr><th></th><th>A</th><th>B</th><th>A ⊕ B</th><th>(¬ (A ⊕ B)</th></tr>
<tr><td>ЕЛЕНА</td><td>1</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>ВЛАДИМИР</td><td>0</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>КСЕНИЯ</td><td>0</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>АРСЕНИЙ</td><td>1</td><td>0</td><td>1</td><td>0</td></tr>
</table>
Правильный ответ – ЕЛЕНА</td></tr>
<tr><td>A11</td><td>3</td><td>На один символ потребуется $[\log_2 36] = 6$ бит, тогда на 9 символов потребуется $9 \cdot 6 = 54$ бит. Минимально возможное количество байт на 9 символов: $54 : 8 = 6{,}75$. Округляя до ближайшего большего целого, получим 7 байт. Под хранение индивидуальных кодов 16 самолётов потребуется: $7 \cdot 16 = 112$ байт</td></tr>
</table>

Продолжение таблицы

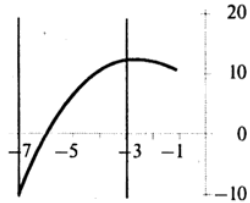
<table>
<tr><th>1</th><th>2</th><th>3</th></tr>
<tr><td>A12</td><td>2</td><td>
В задаче используются два оператора цикла с параметром. В первом цикле массив A заполняется числами от 0 до 10
<table>
<tr><td>i</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td></tr>
<tr><td>A[i]</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td></tr>
</table>
Во втором цикле параметр цикла i изменяется с шагом 1 и принимает значения 0, 1, 2, 3, 4. В теле цикла меняются местами значения переменных A[i] и A[2*i+2], т. е. значения A[0] и A[2], A[1] и A[4], A[2] и A[6], A[3] и A[8], A[4] и A[10].
Значения A[5]=5, A[7]=7, A[9]=9 остались без изменений. В результате получим:
<table>
<tr><td>i</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td></tr>
<tr><td>A[i]</td><td>2</td><td>4</td><td>6</td><td>8</td><td>10</td><td>5</td><td>0</td><td>7</td><td>3</td><td>9</td><td>1</td></tr>
</table>
</td></tr>
<tr><td>A13</td><td>1</td><td>
Максимальное количество клеток, требующих проверки:


</td></tr>
</table>

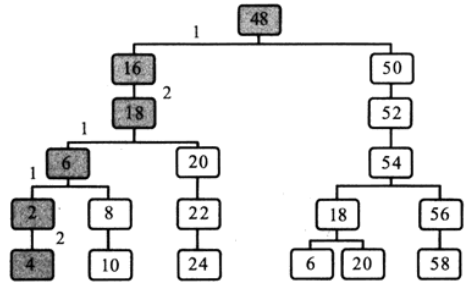
Продолжение таблицы

1	2	3
		Траектория движения РОБОТА и примеры вырожденных случаев:   Клетки, удовлетворяющие условию задачи: 

Окончание таблицы

1	2	3
A14	2	Переменные a, b, d используются для организации цикла. В программе используется функция пользователя $F(x)$, которая возвращает значение выражения $(1 - x)*(x + 6)$. Значение аргумента в теле функции не изменяется. В цикле вычисляется максимальное значение функции $F(t)$. В переменной R сохраняется значение функции $F(t)$, претендующее на максимальное. В переменной M сохраняется значение аргумента t, при котором функция принимает максимальное значение. Таким образом, в программе определяется значение аргумента функции F, при котором функция принимает максимальное значение, на отрезке $[-7; -3]$. Запишем функцию $F(x) = (1 - x)*(x + 6) = - x^2 - 5*x + 6$. Получили уравнение параболы $(px^2 + qx + k)$, ветви которой направлены вниз $(p<0)$, а вершина находится в точке $x_в = -q / 2p = 5 / (-2) = -2{,}5$, $x_в \notin [-7;-3]$. На отрезке $[-7;-3]$ функция возрастает и максимальное значение будет достигнуто на правой границе отрезка в точке -3.  Таким образом, на печать будет выведено число -3

Часть 2

Задание	Ответ	Комментарии
1	2	3
B1	960	Сообщение состоит из 120 символов. Первоначально оно имело объём 120 · 2 байт, после перекодирования 120 · 1 байт. Следовательно, объём сообщения уменьшился на 120 · 2 – 120 = 120 байт = 960 бит
B2	21121	Не все числа делятся на 3 без остатка, следовательно, количество ветвей и узлов обратного дерева решения (от 48 к 4) будет меньше, чем прямого (от 4 к 48). Построим дерево решения от результата 48 к исходному числу 4 с использованием обратных команд 1. делить на 3 2. прибавить 2  Запишем последовательность команд получения из числа 4 числа 48 (от листа к корню дерева)

Продолжение таблицы

<table>
<tr><th>1</th><th>2</th><th>3</th></tr>
<tr><td>B3</td><td>81</td><td>Построим трассировочную таблицу, контрольная точка – логическое выражение (a>b) или (b>a)
<table>
<tr><th>a</th><th>b</th><th>(a>b)</th><th>(b>a)</th></tr>
<tr><td>11</td><td>7</td><td>да</td><td></td></tr>
<tr><td>10</td><td>8</td><td>да</td><td>да</td></tr>
<tr><td>9</td><td>9</td><td>нет</td><td>да</td></tr>
</table>
a*a = $9^2 = 81$</td></tr>
<tr><td>B4</td><td>ИУУУУ</td><td>Буквы А, И, О, У имеют значения 0, 1, 2, 3 соответственно. Тогда записи слов соответствуют значениям неотрицательных целых чисел в четверичной системе счисления: 0000_4, 0001_4, 0002_4, 0003_4, 0010_4, 0011_4, ...
На 512-м месте находится число 511_{10}. Переведём число 511_{10} в четверичную систему счисления:
511 | 4
508 | 127 | 4
3 124 | 31 | 4
3 28 | 7 | 4
3 4 | 1
3
Получим $511_{10} = 13333_4$. Перейдём к исходным обозначениям, получим запись слова ИУУУУ</td></tr>
</table>

Продолжение таблицы

<table>
<tr><th>1</th><th>2</th><th>3</th></tr>
<tr><td>B5</td><td>4</td><td>Проведём возможные вычисления в ячейках таблицы, получим
<table>
<tr><th></th><th>A</th><th>B</th><th>C</th><th>D</th></tr>
<tr><td>1</td><td>4</td><td>4</td><td>2</td><td></td></tr>
<tr><td>2</td><td>8</td><td>6</td><td>=2+D1</td><td>=D1+4</td></tr>
</table>
На диаграмме две пары секторов имеют одинаковые размеры, из фрагмента таблицы следует, что пара разных секторов соответствует значениям 6 и 8. В ячейках C2 и D2 после вычислений должны находиться значения 6 и 8, следовательно,
C2 + D2 = 6 + 8;
A1 + C1 + 2 · D1 = 14;
4 + 2 + 2 · D1 = 14;
D1 = 8 / 2 = 4.
В ячейке D1 должно быть записано число 4</td></tr>
</table>

1	2	3
B6	102	Построим трассировочную таблицу:

№	Команда или условие (логическое выражение)	Вычисление правой части команды присваивания или условия (логического выражения)	a	b	c
1	a = 144	144	144		
2	b = 84	84		84	
3	a/3 > b - a/3	(48 > = 36) = да			
4	c = a - b/2	144 - 84/2 = 102			102
Результат			144	84	102

B7	82	**Шаг 1.** Исходные значения элементов массива R:

R[0]	R[1]	R[2]	R[3]	R[4]
2	2	1	3	2

Продолжение таблицы

1	2	3

Шаг 2. Трассировка вызова функции: `ArrProcessing_1()`:

i	R[0]	R[1]	R[2]	R[3]	R[4]	p
	2	2	1	3	2	2
0		R[0] * *p = 2 * 2 = 4				i+1=0+1=1
1			R[1] * *p = 4 * 1 = 4			i+1=1+1=2
2				R[2] * *p = 4 * 2 = 8		i+1=2+1=3
3					R[3] * *p = 8 * 3 = 24	i+1=3+1=4
Итог:	**2**	**4**	**4**	**8**	**24**	**4**

Шаг 3. Трассировка вызова функции: `MultCounter()`:

i	T	N
	1	0
1	R[0] = 2	N + T* (T – 1) = 0 + 2* 1 = 4
2	R[1] = 4	N + T* (T – 1) = 4 + 4* 3 = 16
3	R[2] = 4	N + T* (T – 1) = 16 + 4* 3 = 28
4	R[3] = 8	N + T* (T – 1) = 28 + 8* 7 = 82
Итог:	**8**	**82**

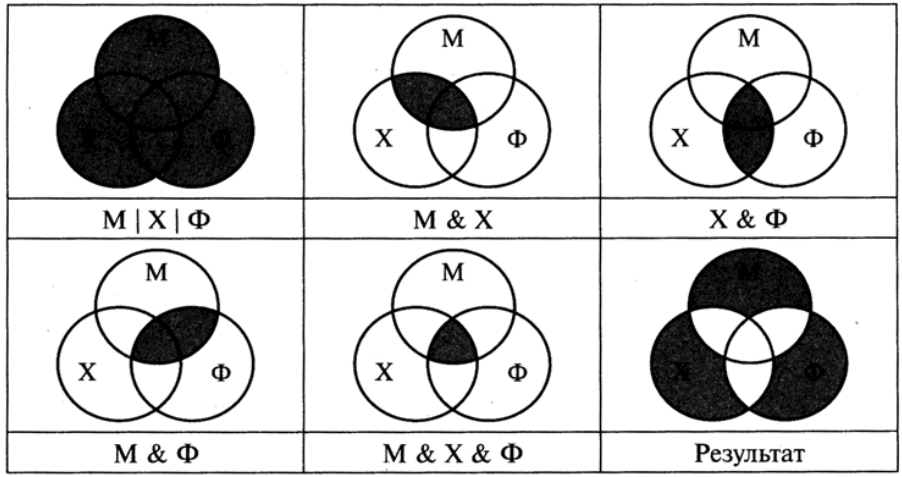
Продолжение таблицы

1	2	3
		Из последней таблицы видно, что в переменную N вызывающей программы попадёт значение 82, именно оно будет выведено на экран.
B8	2	Запишем число в N-ичной системе счисления $YZ1_N$, где Y, Z – цифры, которые могут принимать значения от 0 до N–1. Развёрнутая запись числа имеет вид: $Y \cdot N^2 + Z \cdot N + 1 = 91$ или $N \cdot (Y \cdot N + Z) = 90$. Так как N является одним из сомножителей в формуле определения значения числа, а $90 = 2 \cdot 3 \cdot 3 \cdot 5$, то N может принимать значения 2, 3, 5, 6 ($2 \cdot 3$), 9 ($3 \cdot 3$), 10, 15, ... Сразу можно отбросить 10 и числа больше 10 в качестве оснований систем счисления, так как запись числа в них будет выглядеть меньше, чем 91, и не будет содержать 3 цифры. Будем последовательно переводить 91 в двоичную, троичную и т. д. системы счисления. Получим $91_{10} = 1011011_2 = 10101_3 = 331_5 = 231_6 = 11_9$. В пятеричной и шестеричной системах счисления запись числа 91 имеет три цифры, причём последняя – единица
B9	11	$F(А) = F(Б) + F(В) + F(Ж) = \mid F(Б) = F(Ж) \mid =$ $= F(В) + 2 \cdot F(Ж) = \mid F(В) = F(Г) + F(Д) + F(Е) \mid =$ $= F(Г) + F(Д) + F(Е) + 2 \cdot F(Ж) + F(З) = \mid F(Г) = F(Д) \mid =$ $= 2 \cdot F(Д) + F(Е) + 2 \cdot F(Ж) + F(З) = \mid F(Д) = F(З) \mid =$ $= F(Е) + 2 \cdot F(Ж) + 3 \cdot F(З) = \mid F(Ж) = F(Е) + F(И) \mid =$ $= 3 \cdot F(Е) + 3 \cdot F(З) + 2 \cdot F(И) = \mid F(Е) = F(З) + F(И) \mid =$ $= 6 \cdot F(З) + 5 \cdot F(И) = \mid F(З) = F(И) \mid =$ $= 11 \cdot F(И) = 11$

Продолжение таблицы

1	2	3
B10	1296	2 Мб = $2 \cdot 2^{23}$ бит 10 Мб = $10 \cdot 2^{23}$ бит Вычислим время: $t_1 = 2 \cdot 2^{23}$ бит / 2^{20} бит/с = 16 с $t_2 = 10 \cdot 2^{23}$ бит / 2^{16} бит/с = $10 \cdot 2^7$ с = 1280 с $t_{общ}$ = 1296 с
B11	GDHF	& 10101000 11110000 10100000 10100000_2 = 160 IP-адрес сети: 111.115.238.160
B12	3330	Введём обозначения: М – множество страниц, полученных по запросу *Математика*, Х – множество страниц, полученных по запросу *Химия*, Ф – множество страниц, полученных по запросу *Физика*. |А| – мощность множества А, например: |М| – это количество страниц, найденных по запросу *Математика*. Чтобы получить количество страниц, которые отобразятся в результате запроса (М | Х | Ф) & – ((М & Х) | (М & Ф) | (Х & Ф)), используем круги Эйлера, каждый из которых будет соответствовать множеству (М, Х и Ф).

1	2	3
		Результат запроса показан ниже в таблице. Для получения результата вычтем из количества страниц, найденных по запросу М | Х | Ф, количество страниц по запросам М & Х, М & Ф и Х & Ф. Следует учесть, что пересечение трёх множеств вычиталось трижды, поэтому просуммируем полученное число с удвоенным количеством страниц запроса М & Х & Ф.



Продолжение таблицы

1	2	3
		Результат определяется по формуле: $\lvert(М \mid Х \mid Ф)\rvert - \lvert(М \,\&\, Х)\rvert - \lvert(Х \,\&\, Ф)\rvert - \lvert(М \,\&\, Ф)\rvert + 2 \cdot \lvert(М \,\&\, Х \,\&\, Ф)\rvert =$ $4500 - 630 - 400 - 750 + 2 \cdot 305 = 3330$
B13	11	Числа, полученные в результате выполнения программы, определяются по формуле $К = 7 + 2x - 3(10 - x) = 5x - 23, \quad x = 0, 1, 2,...,10,$ где x – количество команд 1 в программе
B14	64	Система уравнений: $a\bar{b} \vee \bar{a}b \vee cd \vee \bar{c}\bar{d} = 1$ $c\bar{d} \vee \bar{c}d \vee ef \vee \bar{e}\bar{f} = 1$ $e\bar{f} \vee \bar{e}f \vee gh \vee \bar{g}\bar{h} = 1$ $g\bar{h} \vee \bar{g}h \vee ij \vee \bar{i}\bar{j} = 1$ $i\bar{j} \vee \bar{i}j \vee ab \vee \bar{a}\bar{b} = 1$

Продолжение таблицы

1	2	3
241.		Введём обозначения: $K = a \oplus b$ $\quad \bar{K} = a \equiv b$ $L = c \oplus d$ $\quad \bar{L} = c \equiv d$ $M = e \oplus f$ $\quad \bar{M} = e \equiv f$ $N = g \oplus h$ $\quad \bar{N} = g \equiv h$ $O = i \oplus j$ $\quad \bar{O} = i \equiv j$ Переменные K, L, M, N, O взаимно независимы. Система уравнений примет вид: $K \vee \bar{L} = 1$ $L \vee \bar{M} = 1$ $M \vee \bar{N} = 1$ $N \vee \bar{O} = 1$ $O \vee \bar{K} = 1$ Так как все пять выражений должны быть одновременно истинны, объединим их операцией конъюнкции в одно выражение, получим: $(K \vee \bar{L})(L \vee \bar{M})(M \vee \bar{N})(N \vee \bar{O})(O \vee \bar{K}) = 1.$

Окончание таблицы

1	2	3
		При раскрытии скобок, получаем: $(KL \vee K\bar{M} \vee \cancel{\bar{L}L} \vee \bar{L}\bar{M})\,(MN \vee M\bar{O} \vee \cancel{\bar{N}N} \vee \bar{N}\bar{O})\,(O \vee \bar{K}) = 1$; $(KLMN \vee KLM\bar{O} \vee KL\bar{N}\bar{O} \vee \cancel{K\bar{M}MN} \vee \cancel{K\bar{M}M\bar{O}} \vee K\bar{M}\bar{N}\bar{O} \vee \cancel{\bar{L}\bar{M}MN}$ $\vee\, \cancel{\bar{L}\bar{M}M\bar{O}} \vee \bar{L}\bar{M}\bar{N}\bar{O})\,(O \vee \bar{K}) = 1$; $(KLMN \vee KLM\bar{O} \vee KL\bar{N}\bar{O} \vee K\bar{M}\bar{N}\bar{O} \vee \bar{L}\bar{M}\bar{N}\bar{O})\,(O \vee \bar{K}) = 1$; $KLMNO \vee \cancel{KLM\bar{O}O} \vee \cancel{KL\bar{N}\bar{O}O} \vee \cancel{K\bar{M}\bar{N}\bar{O}O} \vee \cancel{\bar{L}\bar{M}N\bar{O}O} \vee \cancel{KLMN\bar{K}} \vee$ $\vee\, \cancel{KLM\bar{O}\bar{K}} \vee \cancel{K\bar{M}N\bar{O}\bar{K}} \vee \bar{L}\bar{M}\bar{N}\bar{O}\bar{K} = 1$. СДНФ имеет вид: $KLMNO \vee \bar{K}\bar{L}\bar{M}\bar{N}\bar{O} = 1$. Количество решений равно: $2^5 + 2^5 = 64$

Часть 3

Задание	Комментарии
1	2
C1	1) Пример: $x = -1$, $y = 0$. Любая пара (x, y), для которой выполняется $(y <= x * x)$ и $(y * y + x * x <= 1)$ и $(y >= 0)$ и $(x <= 0)$. 2) При ложном значении условия в первых операторах ветвления программа не выводит никакого сообщения. Возможная доработка – заменить все операторы ветвления программы на предложенные:

Паскаль

```
if (y<=x*x) and (y*y+x*x<=1) and (y>=0) and (x>=0) then
   writeln('принадлежит')
   else
   writeln('не принадлежит');
```

Бейсик

```
IF y<=x*x AND y*y+x*x<=1 AND y>=0 AND x>=0 THEN
   PRINT "принадлежит"
   ELSE
   PRINT "не принадлежит"
END IF
```

Продолжение таблицы

1	2

	Си
	`if ((y<=x*x) && (y*y+x*x<=1) && (y>=0) && (x>=0))`
	`  printf("принадлежит");`
	`  else`
	`  printf("не принадлежит");`

C2	**Паскаль**
	`{ Обнуляем накопитель суммы и счётчик количества }`
	`  s:=0;`
	`  k:=0;`
	`  for i:=1 to n do`
	`  begin`
	`  if (a[i] mod 2 = 0) then { Нашли чётный элемент }`
	`  begin`
	`  s:=s+a[i]; { Накапливаем сумму }`
	`  k:=k+1; { увеличиваем счётчик }`
	`  end`
	`  end;`
	`{ Вычисляем среднее значение }`
	`  sr:= s/k;`
	`{Выводим количество элементов, значение}`
	`  {которых чётно и среднее значение этих элементов}`
	`  writeln(k,' ',sr);`

Продолжение таблицы

1	2

Бейсик

```
 'Обнуляем накопитель суммы и счётчик количества
S = 0
K = 0
FOR  I = 1 TO N
    IF A(I) mod 2 = 0 THEN 'Нашли чётный элемент
         S = S + A(I)      'Накапливаем сумму
         K = K + 1         'Увеличиваем счётчик
       END IF
NEXT I
'Вычисляем среднее значение
SR = S / K
'Выводим количество элементов, значение которых чётно
PRINT K
'Выводим среднее значение этих элементов
PRINT SR
```

Си

```
// Обнуляем накопитель суммы и счётчик количества
    s =0;
    k =0;
    for(i=0; i<N; i++){
         if (a[i] % 2 == 0) { // Нашли чётный элемент
```

Продолжение таблицы

1	2

	`s=s+a[i];  // Накапливаем сумму`
	`k=k+1;  // Увеличиваем счётчик`
	`}`
	`}`
	`// Вычисляем среднее значение`
	`sr = 1.0*s/k;`
	`//Выводим количество элементов, значение которых чётно`
	`//и среднее значение этих элементов`
	`printf("%d%.2f", k, sr);`
	Естественный язык
	Записываем в переменные S и K начальные значения, равные 0. В цикле от первого до последнего элемента ищем чётные элементы массива. Если значение текущего элемента массива чётно, добавляем его в сумму S = S + A[I], значение переменной K увеличиваем на единицу. После завершения цикла вычисляем среднее значение чётных элементов SR = S / K и затем выводим количество чётных элементов (K) и среднее значение этих элементов (SR).

C3	Пусть $F(n)$ – количество программ получения числа n из числа 1. Составим рекуррентные соотношения для вычисления значений функции по предыдущим значениям аргумента. После выполнения первой команды программы получим либо $1 + 3 = 4$, либо $1 \cdot 2 = 2$. Число 2 невозможно получить другой программой (последовательностью команд). Число 4 можно получить и второй программой ($1 \cdot 2 \cdot 2$). Тогда $F(2) = 1$; $F(4) = 2$.

Продолжение таблицы

1	2
	Для чётных аргументов $n = 2k$ функция равна: $F(2k) = F(k) + F(2k - 3)$. Нечётные аргументы $n = 2k + 1$ можно получить только командой «прибавь 3», поэтому $F(2k + 1) = F(2k + 1 - 3) = F(2k - 2)$. Итак, вычисления будем выполнять по рекуррентным формулам: $F(2k) = F(k) + F(2k - 3)$ при $k > 2$; $F(2k + 1) = F(2k - 2)$ при $k > 2$; $F(2) = 1$; $F(4) = 2$; $F(n) = 0$ при $n < 2$. Определим количество программ получения числа 28 из числа 1, слагаемые будем записывать в порядке возрастания аргументов: $F(28) = F(25) + F(14) = F(22) + F(14) = F(19) + F(14) + F(11) =$ $= F(16) + F(11) + F(8) + F(7) =$ $= F(13) + F(8) + F(8) + F(8) + F(7) =$ $= F(10) + 3 \cdot F(8) + F(7) =$ $= 3 \cdot F(8) + 2 \cdot F(7) + F(5) =$ $= 4 \cdot F(5) + 3 \cdot F(4) + 2 \cdot F(4) =$ $= 5 \cdot F(4) + 4 \cdot F(2) =$ $= 5 \cdot 2 + 4 = 14$. Ответ: 14

Продолжение таблицы

1	2
C4	В программе используется одномерный массив, состоящий из 10 элементов, по количеству фирм. Каждый элемент массива имеет тип «запись» (firm) и состоит из двух полей: строкового для представления имени фирмы name и числового для числа запросов count. 1. Программа считывает в переменную N количество запросов, которые ей предстоит обработать. 2. В переменную M записывается 0 – текущее количество фирм в массиве. 3. В цикле от 1 до N обрабатываются строки входных данных, соответствующие именам фирм, отправивших запросы. 3.1. В переменную строкового типа считывается очередная строка входных данных. 3.2. Во внутреннем цикле просматриваются элементы массива до тех пор, пока не встретится такая же строка в поле name и переменная цикла j меньше или равна M. 3.2.1. Если обнаружена такая же строка в поле name, то значение поля count соответствующего элемента массива увеличивается на 1 и выполнение внутреннего цикла прерывается. 3.2.2. Если переменная цикла j стала больше M, значение M увеличивается на 1, в поле name M-го элемента массива записывается строка, в поле count записывается 1. 4. Проводится сортировка полученных данных в порядке возрастания значения поля count по алгоритму сортировки «пузырьком». 5. Данные выводятся на экран.

Продолжение таблицы

1	2

	Код программы
	
	{ информация о фирме }
	type firm=record
	name:string[100]; { название фирмы }
	count:integer; { число запросов }
	end;
	var
	N,i,j,M:integer;
	firms:array[1..10]of firm; { статистика по запросам }
	t:firm; { вспомогательная переменная }
	s:string[100]; { вводимая строка }
	begin
	for i:=1 to 10 do begin
	firms[i].name:='';
	firms[i].count:=0;
	end;
	readln(N); { считываем длину списка запросов }
	M:=0;
	{ цикл обработки запросов }
	for i:=1 to N do begin
	readln(s);
	j:=1;
	{ ищем фирму в массиве }
	while((j<M)and(firms[j].name<>s))do
	inc(j,1);

Окончание таблицы

1	2

	`if(firms[j].name<>s)then begin`
	`{ фирмы в массиве нет - заводим новый элемент массива}`
	`inc(M,1);`
	`firms[M].name:=s;`
	`firms[M].count:=1;`
	`end`
	`else`
	`{Если фирма найдена, увеличиваем количество голосов на 1}`
	`begin`
	`inc(firms[j].count,1);`
	`end;`
	`end;`
	`{ сортируем записи в порядке возрастания }`
	`for i:=1 to M do`
	`for j:=1 to M-i do`
	`if(firms [j+1].count< firms [j].count) then begin`
	`t:= firms [j];`
	`firms [j]:= firms [j+1];`
	`firms [j+1]:=t;`
	`end;`
	`{ выводим записи }`
	`for i:=1 to M do`
	`writeln(firms[i].name,' ',firms[i].count);`
	`end.`

Вариант 4

Часть 1

<table>
<tr><th>Задание</th><th>Номер ответа</th><th>Комментарии</th></tr>
<tr><td>1</td><td>2</td><td>3</td></tr>
<tr><td>A1</td><td>1</td><td>Переведём число 272 в двоичную систему счисления. Выделим максимальные степени двойки: $272 = 256 + 16 = 2^8 + 2^4$. В восьмой и четвёртый разряды запишем единицы, в остальные — нули, получим: $272_{10} = 100010000_2$. Количество значащих нулей равно 7</td></tr>
<tr><td>A2</td><td>2</td><td>Вычеркнем все авиарейсы, которые прилетают в аэропорт ЮЖНЫЙ, вылетают из аэропорта ВОСТОЧНЫЙ или вылетают из ЮЖНОГО до 13:00.
<table>
<tr><th>Аэропорт вылета</th><th>Аэропорт прилёта</th><th>Время вылета</th><th>Время прилёта</th></tr>
<tr><td>ЗАПАДНЫЙ</td><td>СЕВЕРНЫЙ</td><td>13:10</td><td>14:20</td></tr>
<tr><td>ЮЖНЫЙ</td><td>СЕВЕРНЫЙ</td><td>13:30</td><td>15:00</td></tr>
<tr><td>ЮЖНЫЙ</td><td>ЗАПАДНЫЙ</td><td>14:20</td><td>17:55</td></tr>
<tr><td>~~ВОСТОЧНЫЙ~~</td><td>~~ЮЖНЫЙ~~</td><td>~~14:45~~</td><td>~~16:55~~</td></tr>
<tr><td>~~ВОСТОЧНЫЙ~~</td><td>~~СЕВЕРНЫЙ~~</td><td>~~15:30~~</td><td>~~16:20~~</td></tr>
<tr><td>СЕВЕРНЫЙ</td><td>ЗАПАДНЫЙ</td><td>16:05</td><td>17:45</td></tr>
<tr><td>ЮЖНЫЙ</td><td>ЗАПАДНЫЙ</td><td>16:15</td><td>18:00</td></tr>
<tr><td>СЕВЕРНЫЙ</td><td>ВОСТОЧНЫЙ</td><td>17:10</td><td>19:50</td></tr>
<tr><td>ЗАПАДНЫЙ</td><td>ВОСТОЧНЫЙ</td><td>18:20</td><td>19:05</td></tr>
</table></td></tr>
</table>

Продолжение таблицы

1	2	3

ЗАПАДНЫЙ	ВОСТОЧНЫЙ	18:45	19:25
ЮЖНЫЙ	ВОСТОЧНЫЙ	19:20	20:05

Рассмотрим рейсы, прибывающие в ВОСТОЧНЫЙ (выделено цветом), в порядке возрастания времени прилёта.

1) Прибытие в 19:05. На этот рейс путешественник не успевает, так как не хватает времени на пересадку.

ЗАПАДНЫЙ	ВОСТОЧНЫЙ	18:20	19:05
СЕВЕРНЫЙ	ЗАПАДНЫЙ	16:05	17:45
ЮЖНЫЙ	ЗАПАДНЫЙ	16:15	18:00
ЮЖНЫЙ	СЕВЕРНЫЙ	13:30	15:00

2) Прибытие в 19:25. На этот рейс путешественник может успеть только с двумя пересадками: ЮЖНЫЙ — СЕВЕРНЫЙ — ЗАПАДНЫЙ — ВОСТОЧНЫЙ.

ЮЖНЫЙ	СЕВЕРНЫЙ	13:30	15:00
ЮЖНЫЙ	ЗАПАДНЫЙ	14:20	17:55
СЕВЕРНЫЙ	ЗАПАДНЫЙ	16:05	17:45
ЮЖНЫЙ	ЗАПАДНЫЙ	16:05	18:00
ЗАПАДНЫЙ	ВОСТОЧНЫЙ	18:45	19:25

Остальные рейсы не рассматриваем

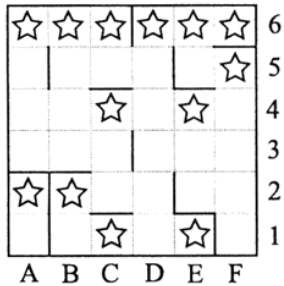
<table>
<tr><th>1</th><th>2</th><th>3</th></tr>
<tr><td>A3</td><td>4</td><td>
<table>
<tr><th>X</th><th>Y</th><th>Z</th><th>F</th><th>¬X ∨ ¬Y ∨ Z</th><th>¬X ∧ Y ∧ Z</th><th>X ∨ Y ∨ ¬Z</th><th>¬X ∨ ¬Y ∨ Z</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td></tr>
</table>
Значения выражения ответа № 4 совпадают с F на всех заданных наборах
</td></tr>
<tr><td>A4</td><td>2</td><td>См. решение варианта 1</td></tr>
<tr><td>A5</td><td>2</td><td>Сумма цифр старших разрядов принадлежит отрезку [2, 18], вторых и младших разрядов — отрезку [0, 18]. Итоговое число состоит из трёх частей. По заданию можно определить, что вторая часть числа должна быть больше третьей. Этим условиям удовлетворяет число 16127 (можем разбить его на три части, принадлежащие отрезку: 16, 12 и 7, где 12 > 7)</td></tr>
<tr><td>A6</td><td>1</td><td>По таблице «Направления» определим код направления «Программная инженерия» — 231000 и проходной балл — 249.
По таблице «Город» определим код Саратова — 65.
По таблице «Абитуриент» определим абитуриентов из Саратова:
<table>
<tr><th>ID_abit</th><th>ID_City</th><th>Mat</th><th>Inf</th><th>Russ</th><th>Ball</th></tr>
<tr><td>1046</td><td>65</td><td>78</td><td>95</td><td>76</td><td>249</td></tr>
<tr><td>1067</td><td>65</td><td>96</td><td>86</td><td>95</td><td>277</td></tr>
</table>
</td></tr>
</table>

Продолжение таблицы

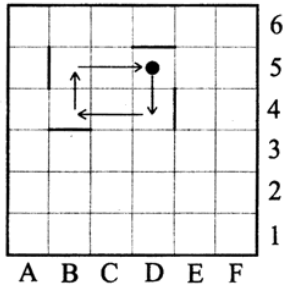
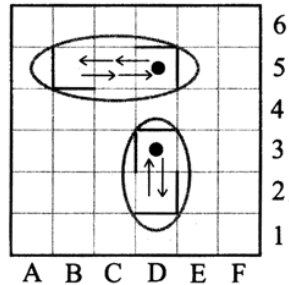
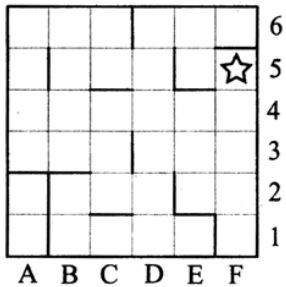
<table>
<tr><th>1</th><th>2</th><th>3</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><td>1094</td><td>65</td><td>84</td><td>87</td><td>75</td><td>246</td></tr>
<tr><td>1105</td><td>65</td><td>82</td><td>78</td><td>75</td><td>235</td></tr>
</table>
По таблице «Заявления» определим, подавали ли заявление на направление 231000 абитуриенты 1046 и 1067, набравшие достаточно баллов для поступления</td></tr>
<tr><td>A7</td><td>4</td><td>Следует оценить отношение количества проданного товара (всего) к количеству поставленного (всего) для каждой компании</td></tr>
<tr><td>A8</td><td>3</td><td>Решение: измерение звука производится 16000 раз в секунду в течение 128 секунд. Для записи одного сигнала требуется
$4000 \cdot 2^{10} / 16000 / 128 = 2^{10} / 4 / 2^{7} = 2$ байт = 16 бит</td></tr>
<tr><td>A9</td><td>4</td><td>Проверим, являются ли предложенные ответы началом заданных кодовых слов или заданные коды началом кодов, предложенных в ответах.
1) 00 — является началом кодов символов Б (000) и В (001).
2) 111 — началом этого кода является код символа Г (11).
3) 011 — началом этого кода является код символа А (01).
4) 10 — не является началом ни одного из заданных кодов символов, ни один из заданных кодов не является началом этого кода.
Кодом буквы Д может быть только код 10</td></tr>
<tr><td>A10</td><td>3</td><td>Введём обозначения логических высказываний:
А — вторая буква гласная;
В — первая буква гласная;
С — последняя буква гласная.</td></tr>
</table>

<table>
<tr><th>1</th><th>2</th><th>3</th></tr>
<tr><td></td><td></td><td>Получим выражение (A → B) ∧ C.
Запишем значения логических высказываний A, B, C и построим таблицу истинности выражения (A → B) ∧ C:
<table>
<tr><th></th><th>A</th><th>B</th><th>C</th><th>(A → B)</th><th>(A → B) ∧ C</th></tr>
<tr><td>ВЯТКА</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>ДНЕПР</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>ЯУЗА</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>ВОЛГА</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td></tr>
</table>
Правильный ответ — ЯУЗА</td></tr>
<tr><td>A11</td><td>1</td><td>Всего символов 33 · 2 + 10 = 76. На один символ потребуется [$\log_2 76$] = 7 бит, тогда на семнадцать символов потребуется 17 · 7 = 119 бит. Минимально возможное количество байтов на семнадцать символов: 119 : 8 ≈ 14,8. Округляя до ближайшего большего целого, получим 15 байт. Под хранение паролей потребуется 15 · 60 = 900 байт</td></tr>
<tr><td>A12</td><td>4</td><td>В задаче используются два оператора цикла с параметром. В первом цикле массив A заполняется числами от 0 до 10:
<table>
<tr><td>i</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td></tr>
<tr><td>A[i]</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td></tr>
</table></td></tr>
</table>

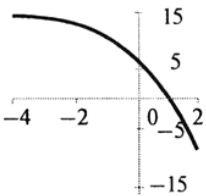
Продолжение таблицы

<table>
<tr><th>1</th><th>2</th><th>3</th></tr>
<tr><td></td><td></td><td>Во втором цикле параметр цикла i изменяется с шагом 2 и принимает значения 0, 2, 4. В теле цикла меняются местами значения переменных A[i] и A[10-i], т. е. значения A[0] и A[10], A[2] и A[8], A[4] и A[6].
Значения A[1]=1, A[3]=3, A[5]=5, A[7]=7, A[9]=9 остались без изменений. В результате получим:
<table>
<tr><td>i</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td></tr>
<tr><td>A[i]</td><td>10</td><td>1</td><td>8</td><td>3</td><td>6</td><td>5</td><td>4</td><td>7</td><td>2</td><td>9</td><td>0</td></tr>
</table></td></tr>
<tr><td>A13</td><td>2</td><td>Максимальное количество клеток, требующих проверки:
 
</td></tr>
</table>

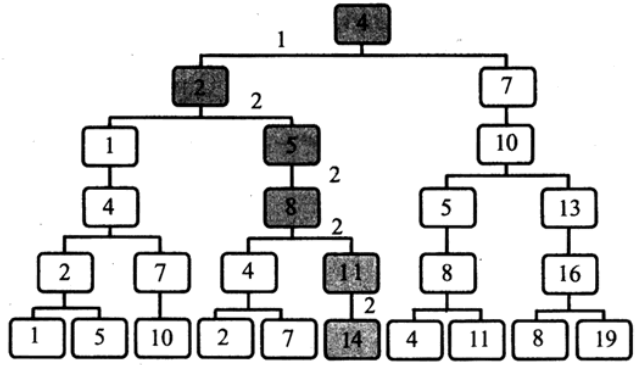
Продолжение таблицы

1	2	3
		Траектории движения РОБОТА:   Клетки, удовлетворяющие условию задачи: 

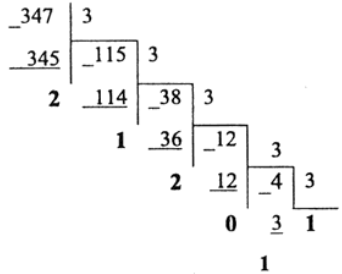
Окончание таблицы

1	2	3
A14	4	Переменные a, b, d используются для организации цикла. В программе используется функция пользователя $F(x)$, которая возвращает значение выражения $(1 - x) * (x + 6)$. Значение аргумента в теле функции не изменяется. В цикле вычисляется минимальное значение функции $F(t)$. В переменной R сохраняется значение функции $F(t)$, претендующее на минимальное. В переменной M сохраняется значение аргумента t, при котором функция принимает минимальное значение. Таким образом, в программе определяется значение аргумента функции F, при котором функция принимает минимальное значение, на отрезке [–3; 2]. Запишем функцию $F(x) = (1 - x) * (x + 6) = -x^2 - 5 * x + 6$. Получили уравнение параболы $(px^2 + qx + k)$, ветви которой направлены вниз ($p < 0$), а вершина находится в точке $x_в = -q / 2p = 5/ (-2) = -2{,}5$, $x_в \notin [-3; 2]$. На отрезке [–3; 2] функция убывает, и минимальное значение будет достигнуто на правой границе отрезка в точке 2.  Таким образом, на печать будет выведено число 2

Часть 2

Задание	Ответ	Комментарии
1	2	3
B1	448	Определим длину сообщения, учитывая все символы и не забывая о пробелах. Сообщение состоит из 56 символов. Первоначально оно имело объём 56 · 2 байт, после перекодирования — 56 · 1 байт. Следовательно, объём сообщения уменьшился на 56 · 8 = 448 байт
B2	12222	Не все числа делятся на 2 без остатка, следовательно, количество ветвей и узлов дерева решения от 4 к 14 будет меньше, чем обратного (от 14 к 4). Построим дерево решения от исходного числа 4 к результату 14 с использованием заданных команд. Решение выделено цветом, ветви помечены номерами команд. Запишем номера команд получения из числа 4 числа 14 (от корня к листу): 

Продолжение таблицы

<table>
<tr><th>1</th><th>2</th><th>3</th></tr>
<tr><td>B3</td><td>38</td><td>Составим трассировочную таблицу, контрольная точка — логическое выражение 5 * a – b <= 0:
<table>
<tr><th>a</th><th>b</th><th>5 * a – b <= 0</th><th>a > b</th></tr>
<tr><td>2</td><td>20</td><td>да</td><td></td></tr>
<tr><td>4</td><td></td><td>да</td><td>нет</td></tr>
<tr><td>6</td><td></td><td>нет</td><td>нет</td></tr>
</table>
a*a – 3*a + b = $6^2 - 3 * 6 + 20 = 36 - 18 + 20 = 38$</td></tr>
<tr><td>B4</td><td>ООА
УОУ</td><td>Три буквы — А, О, У — можно заменить цифрами 0, 1 и 2 соответственно (А = 0, О = 1, У = 2). Список слов соответствует значениям неотрицательных целых чисел в троичной системе счисления:
1) $00001_3 = 1_{10}$ 2) $00002_3 = 2_{10}$ 3) $00010_3 = 3_{10}$ 4) $00011_3 = 4_{10}$
На 347-м месте находится число 347_{10}. Переведём его в троичную систему счисления:


Результат – запись остатков от деления – равен 110212, что соответствует записи слова с использованием заданных букв ООАУОУ</td></tr>
</table>

<table>
<tr><th>1</th><th>2</th><th>3</th></tr>
<tr><td>B5</td><td>2</td><td>
Выполним копирование ячеек, учитывая, что относительные ссылки в формулах меняются при копировании:
<table>
<tr><th></th><th>A</th><th>B</th><th>C</th><th>D</th></tr>
<tr><td>19</td><td>3</td><td>4</td><td>1</td><td></td></tr>
<tr><td>20</td><td>=A19-1</td><td>=B19+1</td><td>=C19</td><td>=D19*2</td></tr>
<tr><td>21</td><td>=A20-1</td><td>=B20+1</td><td>=C20</td><td>=D20*2</td></tr>
</table>
Проведём возможные вычисления в ячейках таблицы, получим:
<table>
<tr><th></th><th>A</th><th>B</th><th>C</th><th>D</th></tr>
<tr><td>19</td><td>3</td><td>4</td><td>1</td><td></td></tr>
<tr><td>20</td><td>2</td><td>5</td><td>1</td><td>=D19*2</td></tr>
<tr><td>21</td><td>1</td><td>6</td><td>1</td><td>=D20*2</td></tr>
</table>
На диаграмме три сектора имеют значения из диапазона A21:C21, а именно 1, 6 и 1. Четвёртый сектор равен сумме трёх первых, следовательно, значение ячейки D21 равно 8: D21 = D20 * 2 = 8. Подставим формулу из D20: D19 * 4 = 8.
В ячейке D19 должно быть записано число 2
</td></tr>
</table>

Продолжение таблицы

<table>
<tr><th>1</th><th>2</th><th>3</th></tr>
<tr><td>B6</td><td>36</td><td>Построим трассировочную таблицу первого вида:
<table>
<tr><th>№</th><th>Команда или условие (логическое выражение)</th><th>Вычисление правой части команды присваивания или условия (логического выражения)</th><th>a</th><th>b</th><th>c</th></tr>
<tr><td>1</td><td>a = 48</td><td>48</td><td>48</td><td></td><td></td></tr>
<tr><td>2</td><td>b = 24</td><td>24</td><td></td><td>24</td><td></td></tr>
<tr><td>3</td><td>b > a/2 - 1</td><td>(24 > 23) = да</td><td></td><td></td><td></td></tr>
<tr><td>4</td><td>c = b + a/4</td><td>24 + 48/4 = 36</td><td></td><td></td><td>36</td></tr>
<tr><td colspan="3">Результат</td><td>48</td><td></td><td></td></tr>
</table></td></tr>
<tr><td>B7</td><td>72</td><td>Шаг 1. Исходные значения элементов массива A:
<table>
<tr><th>i</th><th>A[0]</th><th>A[1]</th><th>p</th></tr>
<tr><td></td><td></td><td></td><td>15</td></tr>
<tr><td>1</td><td>p/i=15/1=15</td><td></td><td></td></tr>
<tr><td>2</td><td></td><td>p/i=15/2=7</td><td></td></tr>
<tr><td></td><td>15</td><td>7</td><td></td></tr>
</table></td></tr>
</table>

<table>
<tr><th>1</th><th>2</th><th>3</th></tr>
<tr><td></td><td></td><td>

Шаг 2. Трассировка вызова функции `ArrCr(L, A, B, &p)`. В функции получают значения элементы массива `B`:

<table>
<tr><th>i</th><th>B[0]</th><th>B[1]</th><th>p</th></tr>
<tr><td></td><td></td><td></td><td>1</td></tr>
<tr><td>0</td><td>A[0]+*p = 15+1=16</td><td></td><td>*p+i=1+0=1</td></tr>
<tr><td>1</td><td></td><td>A[1]+*p = 7 + 1 = 8</td><td>*p+i=1+1=2</td></tr>
<tr><td>Итог</td><td>16</td><td>8</td><td>11</td></tr>
</table>

Шаг 3. Трассировка вызова функции `Combine(L, A, B)`:

<table>
<tr><th>i</th><th>N</th></tr>
<tr><td></td><td>0</td></tr>
<tr><td>0</td><td>N +A[0]+2*B[1] = 0+15+2*8=31</td></tr>
<tr><td>1</td><td>N +A[1]+2*B[0] = 31+7+2*16=70</td></tr>
<tr><td></td><td>N +i = 70+2</td></tr>
<tr><td>Итог</td><td>72</td></tr>
</table>

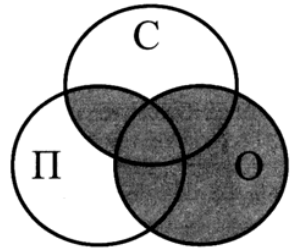
Из последней таблицы видно, что в переменную `N` вызывающей программы попадёт значение 72, именно оно будет выведено на экран

</td></tr>
<tr><td>B8</td><td>57</td><td>Запишем число в N-ичной системе счисления $YZ2_N$, где Y, Z — цифры, которые могут принимать значения от 0 до N–1.</td></tr>
</table>

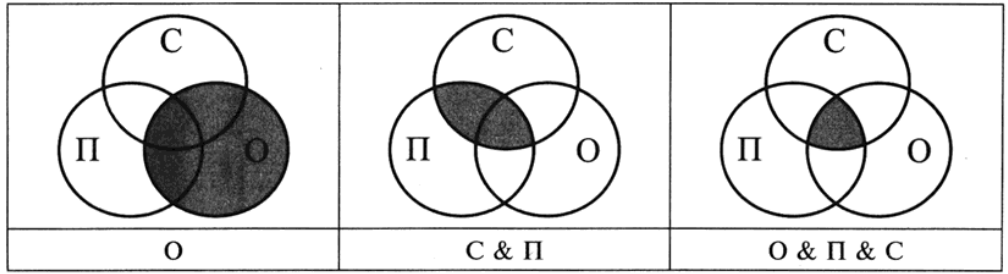
Продолжение таблицы

1	2	3
		Развёрнутая запись числа имеет вид: $Y \cdot N^2 + Z \cdot N + 2 = 107$ или $N \cdot (Y \cdot N + Z) = 105$. Так как N — один из сомножителей в формуле определения значения числа, а $105 = 3 \cdot 5 \cdot 7$, то N может принимать значения 3, 5, 7, 15 ($3 \cdot 5$), 21, 35, 105. Сразу можно отбросить числа больше 10 в качестве оснований систем счисления, так как запись числа в них будет выглядеть меньше 107, и не будет содержать три цифры. Последовательно переведём 107 в троичную, пятеричную и т. д. системы счисления. Получим: $107_{10} = 10222_3 = 412_5 = 212_7$. В пятеричной и семеричной системах счисления запись числа 107 имеет три цифры, причём последняя — 2
B9	9	$F(А) = F(Б) + F(В) + F(Е) = \mid F(Б) = F(Е) + F(Ж) \mid =$ $= F(В) + 2 \cdot F(Е) + F(Ж) = \mid F(В) = F(Г) \mid =$ $= F(Г) + 2 \cdot F(Е) + F(Ж) = \mid F(Г) = F(Е) + F(Д) \mid =$ $= F(Д) + 3 \cdot F(Е) + F(Ж) = \mid F(Д) = F(З) + F(И) \mid =$ $= 3 \cdot F(Е) + F(Ж) + F(З) + F(И) = \mid F(Е) = F(Ж) + F(З) \mid =$ $= 4 \cdot F(Ж) + 4 \cdot F(З) + F(И) = \mid F(Ж) = F(И);\ F(З) = F(И) \mid = 9 \cdot F(И) = 9$
B10	3616	1 Мб = 2^{23} бит 7 Мб = $7 \cdot 2^{23}$ бит Вычислим время: $t_1 = 2^{23}$ бит / 2^{18} бит/с = 32 с $t_2 = 7 \cdot 2^{23}$ бит / 2^{14} бит/с = 3584 с $t_{общ} = 3616$ с

Продолжение таблицы

1	2	3
B11	BD AF	& 00100100 11110000 00100000 $00100000_2 = 32$ IP-адрес сети: 30.115.32.0
B12	5900	Введём обозначения: С — множество страниц, полученных по запросу *Салат*, П – множество страниц, полученных по запросу *Помидор*, О — множество страниц, полученных по запросу *Огурец*. Чтобы получить число страниц, которые отобразятся в результате запроса *(Салат & Помидор) | Огурец,* используем круги Эйлера, каждый из которых будет соответствовать множеству (С, П и О). Результатом требуемого запроса будет являться закрашенная область. Для получения результата просуммируем количество страниц, найденных по запросу *Огурец,* с количеством страниц, найденных по запросу *Салат & Помидор*. Следует учесть, что пересечение трёх множеств было включено дважды, поэтому вычтем из полученного числа количество страниц, полученных для запроса *Салат & Помидор & Огурец*. 

Продолжение таблицы

1	2	3
		 Результат определяется по формуле: |О| + |(С & П)| – |(С & П & О)| = 4000 + 3800 – 1900 = 5900
B13	11	Числа, полученные в результате выполнения программы, определяются по формуле К = 7 – 2x + 3(10 – x) = 37 – 5x, x = 0, 1, 2, ..., 10, где x — количество команд 1 в программе
B14	9	$((A \vee \bar{B} \vee C) \to \overline{D \to \bar{E}})\ (DE \to (A \vee \bar{B} \vee C))\ (D \vee E \vee \bar{B})$ Убираем импликацию: $((A \vee \bar{B} \vee C) \to \overline{\bar{D} \vee \bar{E}})\ (DE \to (A \vee \bar{B} \vee C))\ (D \vee E \vee \bar{B})$. Снимаем отрицание по закону де Моргана и двойного отрицания: $((A \vee \bar{B} \vee C) \to DE))\ (DE \to (A \vee \bar{B} \vee C))\ (D \vee E \vee \bar{B})$. Убираем импликации: $(\overline{A \vee \bar{B} \vee C} \vee DE))\ (\overline{DE} \vee A \vee \bar{B} \vee C))\ (D \vee E \vee \bar{B})$.

1	2	3
		Снимаем отрицания (закон де Моргана и двойного отрицания): $(\bar{A}B\bar{C} \vee DE))\ (\bar{D} \vee \bar{E} \vee A \vee \bar{B} \vee C))\ (D \vee E \vee \bar{B})$. Раскрываем скобки, используем закон непротиворечия: $(\bar{A}B\bar{C}\bar{D} \vee \bar{A}B\bar{C}\bar{E} \vee \cancel{\bar{A}AB\bar{C}} \vee \cancel{\bar{A}B\bar{B}\bar{C}} \vee \cancel{\bar{A}B\bar{C}C} \vee \cancel{D\bar{D}E} \vee$ $\vee \cancel{DE\bar{E}} \vee ADE \vee \bar{B}DE \vee CDE)\ (D \vee E \vee \bar{B})$. Раскрываем скобки, используем закон непротиворечия: $(\cancel{\bar{A}B\bar{C}\bar{D}D} \vee \bar{A}B\bar{C}\bar{D}E \vee \cancel{\bar{A}B\bar{B}\bar{C}\bar{D}} \vee \bar{A}B\bar{C}D\bar{E} \vee \cancel{\bar{A}B\bar{C}\bar{E}E} \vee \cancel{\bar{A}B\bar{B}\bar{C}\bar{E}} \vee$ $\vee ADE \vee A\bar{B}DE \vee \bar{B}DE \vee CDE \vee \bar{B}CDE$. Используем закон поглощения: $\bar{A}B\bar{C}\bar{D}E \vee \bar{A}B\bar{C}D\bar{E} \vee ADE \cancel{\vee A\bar{B}DE} \vee \bar{B}DE \vee CDE \cancel{\vee \bar{B}CDE}$. Для перехода к СДНФ домножим неполные конъюнкции и используем закон идемпотентности (тавтологии): $\bar{A}B\bar{C}\bar{D}E \vee \bar{A}B\bar{C}D\bar{E} \vee ADE(\bar{B} \vee B)\ (\bar{C} \vee C) \vee \bar{B}DE(\bar{A} \vee A)\ (\bar{C} \vee C) \vee CDE(\bar{A} \vee A)\ (\bar{B} \vee B) =$ $= \bar{A}B\bar{C}\bar{D}E \vee \bar{A}B\bar{C}D\bar{E} \vee A\bar{B}\bar{C}DE \vee AB\bar{C}DE \vee A\bar{B}CDE \vee ABCDE \vee \bar{A}\bar{B}\bar{C}DE \vee \cancel{\bar{A}\bar{B}\bar{C}DE} \vee$ $\vee \cancel{\bar{A}\bar{B}\bar{C}DE} \vee \cancel{A\bar{B}\bar{C}DE} \vee \bar{A}\bar{B}CDE \vee \cancel{A\bar{B}\bar{C}DE} \vee \bar{A}BCDE \vee \cancel{ABCDE}$. Получили СДНФ, содержащую 9 слагаемых: $\bar{A}B\bar{C}\bar{D}E \vee \bar{A}B\bar{C}D\bar{E} \vee A\bar{B}\bar{C}DE \vee AB\bar{C}DE \vee A\bar{B}CDE \vee ABCDE \vee \bar{A}\bar{B}\bar{C}DE \vee \bar{A}\bar{B}CDE \vee$ $\vee \bar{A}BCDE = 1$.

Часть 3

Задание	Комментарии
1	2
C1	Пример 1: $x = -0{,}5$, $y = 0{,}25$. Будет выведено ошибочное сообщение «не принадлежит». Пример 2: $x = 0{,}5$, $y = 0{,}25$. При ложном значении условия в любом из первых двух операторов ветвления программа не выводит никакого сообщения. Возможная доработка — заменить все операторы ветвления программы на предложенные:

Паскаль

```
if (y<=0) and (x>=0) and (y*y+x*x<=1) or
   (x<=0) and (y<=x*x) and (y*y+x*x<=1) then
  writeln('принадлежит')
  else
  writeln('не принадлежит');
```

Бейсик

```
IF y<=0 AND x>=0 AND y*y+x*x<=1 OR x<=0 AND y<=x*x AND y*y+x*x<=1
THEN
  PRINT "принадлежит"
  ELSE
  PRINT "не принадлежит"
END IF
```

Продолжение таблицы

1	2

	Си
	`if (((y<=0) && (x>=0) && (y*y+x*x<=1))||`
	`((x<=0) && (y<=x*x) && (y*y+x*x<=1)))`
	`printf("принадлежит");`
	`else`
	`printf("не принадлежит");`

C2	Фрагмент программы с решением.
	Паскаль
	`k1:= 0; k2:= 0; k3:= 0;`
	`for i:=1 to n do`
	`if (a[i] = 1) then k1 := k1 + 1 {Нашли 1}`
	`else if (a[i] = 2) then k2 := k2 + 1 {Нашли 2}`
	`else k3 := k3 + 1; {Нашли 3}`
	`{Заносим в начало массива K3 троек}`
	`for i:=1 to k3 do a[i] := 3;`
	`{ Заносим за тройками K2 двоек}`
	`for i:=1 to k2 do a[i + k3] := 2;`
	`{ Заносим за двойками K1 единиц }`
	`for i:=1 to k1 do a[i + k2 + k3] := 1;`
	`{ Выводим массив после сортировки }`
	`for i:=1 to n do write(' ', a[i]);`
	`writeln;`

Продолжение таблицы

1	2

Бейсик

```
     'Обнуляем счётчики
  K1 = 0: K2 = 0: K3 = 0
 'Перебираем все элементы
  FOR I = 1 TO N
     IF A(I) = 1 THEN        'Нашли 1
        K1 = K1 + 1
      ELSE
        IF A(I) = 2 THEN     'Нашли 2
           K2 = K2 + 1
        ELSE
           K3 = K3 + 1   'Нашли 3
        END IF
     END IF
  NEXT I
 'Заносим в начало массива K3 троек
  FOR I = 1 TO K3
      A(I) = 3
  NEXT I
 'Заносим за тройками K2 двоек
  FOR I = 1 TO K2
      A(I + K3) = 2
  NEXT I
```

Продолжение таблицы

1	2
	'Заносим за двойками K1 единиц FOR I = 1 TO K1 A(I + K2 + K3) = 1 NEXT I 'Выводим массив после сортировки FOR I = 1 TO N PRINT A(I) NEXT I
	Си

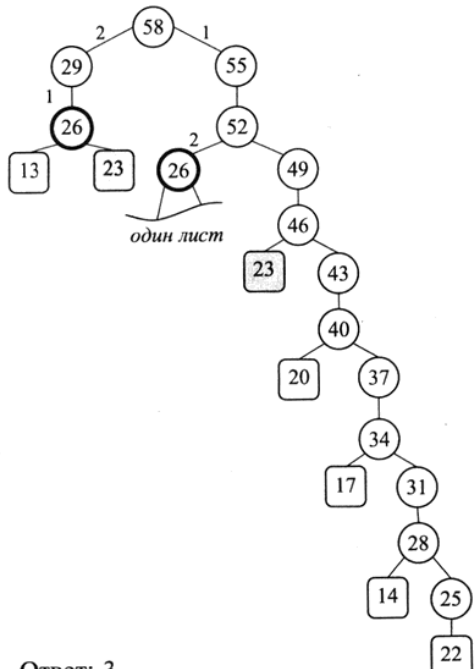
	// Обнуляем счётчики
	k1 = 0; k2 = 0; k3 = 0;
	for(i=0; i<N; i++)
	if(a[i] == 1) k1 = k1 + 1; // Нашли 1
	else
	if(a[i] == 2) k2 = k2 + 1; // Нашли 2
	else
	k3 = k3 + 1; // Нашли 3
	// Заносим в начало массива K3 троек
	for(i=0; i<k3; i++) a[i] = 3;
	// Заносим за тройками K2 двоек
	for(i=0; i<k2; i++) a[i+k3] = 2;
	// Заносим за двойками K1 единиц
	for(i=0; i<k1; i++) a[i + k2 + k3] = 1;
	for(i=0; i<N; i++) printf("%d",a[i]);

Продолжение таблицы

1	2
	Естественный язык Записываем в переменные K1, K2, K3 начальное значение, равное 0. В цикле от первого элемента до последнего элемента сравниваем элементы. Если значение текущего элемента равно 1, то увеличиваем значение переменной K1 на 1, если значение текущего элемента равно 2, то увеличиваем значение переменной K2 на 1, если значение текущего элемента равно 3, то увеличиваем значение переменной K3 на 1. Затем в цикле от 1 до K3 заполняем массив тройками, начиная с первого элемента. Затем в цикле от 1 до K2 заполняем массив двойками, начиная с элемента с индексом I + K3. Затем в цикле от 1 до K1 заполняем массив единицами, начиная с элемента с индексом I + K2 + K3. После завершения последнего цикла выводим элементы массива.
C3	Пусть $F(n)$ — количество программ получения числа n из числа 23. Составим рекуррентные соотношения для вычисления значений функции по предыдущим значениям аргумента. После выполнения первой команды программы получим либо $23 + 3 = 26$, либо $23 \cdot 2 = 46$. Число 26 невозможно получить другой программой из 23. Число 46 также невозможно получить другой программой. Другой программой могла бы быть программа с последовательностью команд `прибавь 3`. Но $46 - 23 = 23 \neq 3 \cdot i$ (23 не кратно трём). Тогда $F(26) = 1$; $F(46) = 1$. Для чётных аргументов $n = 2k$ функция равна: $F(2k) = F(k) + F(2k - 3)$.

1	2
	Нечётные аргументы $n = 2k + 1$ можно получить только командой «прибавь 3», поэтому $F(2k + 1) = F(2k + 1 - 3) = F(2k - 2)$. Итак, вычисления будем выполнять по рекуррентным формулам $F(2k) = F(k) + F(2k - 3)$ при $k > 13$; $F(2k + 1) = F(2k - 2)$ при $k > 13$; $F(26) = 1$; $F(46) = 1$; $F(n) = 0$ при $n < 26$. Определим количество программ получения числа 58 из числа 23, слагаемые будем записывать в порядке возрастания аргументов: $F(58) = F(55) + F(29) = F(52) + F(29) =$ $= F(46) + F(26) + F(26) =$ $= 1 + 2 \cdot F(26) =$ $= 1 + 2 = 3$. Проверим решение — построим дерево решений, используя обратные команды 1. вычти 3 2. раздели на 2 от результата 58 к исходному числу 23. Правая ветвь (и ветвь вниз) соответствует команде 1, левая – команде 2, листьями будут вершины со значениями 23 и меньше. Вершины с одинаковыми значениями, из которых есть путь к листьям 23, выделим толщиной контура. Листья обозначим прямоугольниками, а листья со значением 23 выделим цветом.

Продолжение таблицы

1	2
	 Ответ: 3

1	2
C4	В программе используется одномерный массив, состоящий из 12 элементов, по количеству сортов кофе. Каждый элемент массива имеет тип «запись» (`coffee`) и состоит из двух полей: строкового для представления названия сорта кофе `name` и числового для суммы продаж по сорту `summ`. 1. Программа считывает в переменную `N` количество покупок, которые ей предстоит обработать. 2. В переменную `M` записывается 0 — текущее количество сортов кофе в массиве. 3. В цикле от 1 до `N` обрабатываются строки входных данных, соответствующие отдельным покупкам. 3.1. В переменную строкового типа последовательно считываются символы до достижения пробела. 3.2. В переменную числового типа считывается сумма заказа. 3.3. Во внутреннем цикле просматриваются элементы массива до тех пор, пока не встретится такая же строка в поле `name` и переменная цикла `j` меньше или равна `M`. 3.3.1. Если обнаружена такая же строка в поле `name`, то значение поля `summ` соответствующего элемента массива увеличивается на значение считанного числа. 3.3.2. Если переменная цикла `j` стала больше `M`, значение `M` увеличивается на 1, в поле `name` `M`-го элемента массива записывается строка, в поле `summ` — считанное число. 4. Проводится сортировка полученных данных в порядке возрастания значения поля `summ` по алгоритму сортировки «пузырьком». 5. Данные выводятся на экран.

Продолжение таблицы

1	2
	Код программы

```
{ информация о сорте кофе }
type coffee=record
         name:string[100]; { название сорта }
         summ:integer; { суммарная стоимость }
        end;
var
    N,i,M,j,price:integer;
    coffees:array[1..12]of coffee; { отчёт по сортам кофе }
    t:coffee; { вспомогательная переменная }
    ch:char; { символ для посимвольного считывания }
    s:string[100]; { буферная переменная }
begin
       for i:=1 to 10 do begin
              coffees[i].name:='';
              coffees[i].summ:=0;
       end;
       readln(N); { считываем количество покупок }
       M:=0;
       for i:=1 to N do begin
             s:='';
             { считываем сорт кофе }
             read(ch);
```

Продолжение таблицы

1	2

	while(ch<>' ')do begin
	s:=s+ch;
	read(ch);
	end;
	{ считываем стоимость покупки }
	readln(price);
	j:=1;
	{ ищем сорт кофе в массиве }
	while((j<M)and(coffees[j].name<>s))do
	inc(j,1);
	if(coffees[j].name<>s)then begin
	{ сорт не найден - добавляем новый элемент в массив}
	inc(M,1);
	coffees[M].name:=s;
	coffees[M].summ:=price;
	end
	else
	{ сорт кофе найден, увеличиваем сумму }
	begin
	coffees[j].summ:=coffees[j].summ+price;
	end;
	end;
	{ сортируем массив записей в порядке возрастания }
	for i:=1 to M do

Окончание таблицы

1	2

	for j:=1 to M-i do
	if(coffees [j+1].summ< coffees [j].summ) then begin
	t:= coffees [j];
	coffees [j]:= coffees [j+1];
	coffees [j+1]:=t;
	end;
	{ выводим элементы массива }
	for i:=1 to M do
	writeln(coffees[i].name,' ',coffees[i].summ);
	end.

Вариант 5

Часть 1

<table>
<tr><th>Задание</th><th>Номер ответа</th><th>Комментарии</th></tr>
<tr><td>1</td><td>2</td><td>3</td></tr>
<tr><td>A1</td><td>1</td><td>Переведём число 258 в двоичную систему счисления. Для того чтобы сделать это быстро, выделим максимальные степени двойки: 258 – 256 = 2. Итак, 258 = 256 + + 2 = $2^8 + 2^1$. Восьмой и первый разряды заполним единицами, остальные — нулями, получаем: $258_{10} = 100000010_2$. Число значащих нулей равно 7</td></tr>
<tr><td>A2</td><td>2</td><td>Вычеркнем все рейсы, которые прибывают на станцию ЛИСЬЕ или выезжают из пункта ЗАЙЦЕВО.
<table>
<tr><th>Пункт отправления</th><th>Пункт прибытия</th><th>Время отправления</th><th>Время прибытия</th></tr>
<tr><td>ЛИСЬЕ</td><td>ЕЖОВО</td><td>9:10</td><td>10:15</td></tr>
<tr><td>ЛИСЬЕ</td><td>СОБОЛЕВО</td><td>10:25</td><td>11:20</td></tr>
<tr><td>ЕЖОВО</td><td>СОБОЛЕВО</td><td>10:40</td><td>11:25</td></tr>
<tr><td>ЛИСЬЕ</td><td>ВОЛКОВО</td><td>10:55</td><td>12:50</td></tr>
<tr><td>~~СОБОЛЕВО~~</td><td>~~ЛИСЬЕ~~</td><td>~~10:20~~</td><td>~~11:10~~</td></tr>
<tr><td>СОБОЛЕВО</td><td>ЗАЙЦЕВО</td><td>11:30</td><td>12:20</td></tr>
<tr><td>ЕЖОВО</td><td>ЗАЙЦЕВО</td><td>11:35</td><td>13:30</td></tr>
</table></td></tr>
</table>

Продолжение таблицы

1	2	3

СОБОЛЕВО	ЕЖОВО	11:40	12:25
СОБОЛЕВО	ВОЛКОВО	11:45	12:35
ЕЖОВО	ЗАЙЦЕВО	12:30	12:55
ВОЛКОВО	ЗАЙЦЕВО	12:40	13:25

Определим рейсы, прибывающие в ЗАЙЦЕВО (выделены в таблице цветом). Проверим, есть ли у путешественника возможность добраться этими рейсами, начиная с самого раннего времени.

1) На рейс, прибывающий в 12:20, путешественник попасть не может.

СОБОЛЕВО	ЗАЙЦЕВО	11:30	12:20
ЛИСЬЕ	ЕЖОВО	9:10	10:15
ЛИСЬЕ	СОБОЛЕВО	10:25	11:20
ЕЖОВО	СОБОЛЕВО	10:40	11:25

2) На рейс, прибывающий в 12:55 в ЗАЙЦЕВО, путешественник успевает, ему придётся сделать пересадку в ЕЖОВО.

ЕЖОВО	ЗАЙЦЕВО	12:30	12:55
ЛИСЬЕ	ЕЖОВО	9:10	10:15

Остальные рейсы прибывают позже, их можно не проверять

Продолжение таблицы

<table>
<tr><th>1</th><th>2</th><th>3</th></tr>
<tr><td>A3</td><td>2</td><td>
<table>
<tr><th>X</th><th>Y</th><th>Z</th><th>F</th><th>X ∨ ¬Y ∨ Z</th><th>¬X ∨ Y ∨ Z</th><th>¬X ∨ Y ∨ ¬Z</th><th>¬X ∧ ¬Y ∧ ¬Z</th></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr>
</table>
Значения выражения ответа № 2 совпадают с F на всех заданных наборах переменных
</td></tr>
<tr><td>A4</td><td>2</td><td>См. решение варианта 1</td></tr>
<tr><td>A5</td><td>4</td><td>Сумма цифр старших разрядов принадлежит отрезку [2, 18], остальных разрядов — отрезку [0, 18]. Итоговое число состоит из трёх частей. По заданию можно определить, что все части расположены в порядке возрастания. Этим условиям удовлетворяет число 91317 (можем разбить его на три части, принадлежащие отрезку: 9, 13 и 17, где 9 < 13 < 17)</td></tr>
<tr><td>A6</td><td>3</td><td>Выясняем по таблице «Товары» ID_товара для конфет «Мишка косолапый» и их цену за 1 кг:
<table>
<tr><th>ID_товара</th><th>Наименование</th><th>Цена за 1 кг, руб.</th></tr>
<tr><td>65</td><td>Конфеты «Мишка косолапый»</td><td>500,00</td></tr>
</table>
Затем по таблице «Покупатели» находим ID_покупателя для кафе «Шоколадница»:</td></tr>
</table>

Продолжение таблицы

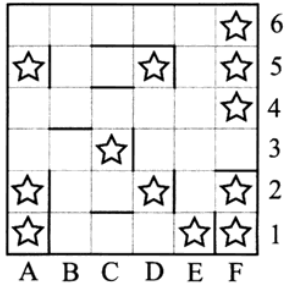
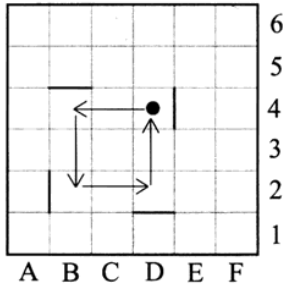
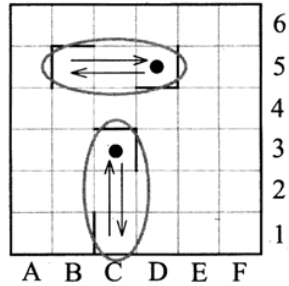
<table>
<tr><th>1</th><th>2</th><th>3</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>ID_покупателя</th><th>Наименование</th></tr>
<tr><td>2148</td><td>Кафе «Шоколадница»</td></tr>
</table>
Затем по таблице «Заказы» находим заказ, где ID_товара = 65, а ID_покупателя = = 2148:
<table>
<tr><th>ID_заказа</th><th>ID_ покупателя</th><th>ID_ товара</th><th>Количество (кг)</th><th>Дата заказа</th><th>Дата доставки</th></tr>
<tr><td>11112</td><td>2148</td><td>65</td><td>36,6</td><td>22.06.2011</td><td>30.06.2011</td></tr>
<tr><td>11172</td><td>2148</td><td>65</td><td>17</td><td>04.07.2011</td><td>09.07.2011</td></tr>
</table>
Тогда кафе «Шоколадница» потратит на покупку конфет «Мишка косолапый» $36{,}6 \cdot 500 + 17 \cdot 500 = 26800$ рублей
</td></tr>
<tr><td>A7</td><td>4</td><td>Следует оценить отношение общего количества поступивших по ЕГЭ к общему количеству студентов для каждого вуза</td></tr>
<tr><td>A8</td><td>3</td><td>Решение: измерение звука производится в течение 64 секунд. Для сохранения результата каждого измерения требуется 24 бит = 3 байт. Определим, сколько раз в секунду производятся измерения:
$3000 \cdot 2^{10} /3/ 64= 125 \cdot 2^{13} / 2^{6} = 125 \cdot 2^{7} = 125 \cdot 128 = 16000$ Гц = 16 кГц</td></tr>
<tr><td>A9</td><td>3</td><td>Проверим, являются ли предложенные ответы началом заданных кодовых слов или заданные коды началом кодов, предложенных в ответах.</td></tr>
</table>

Продолжение таблицы

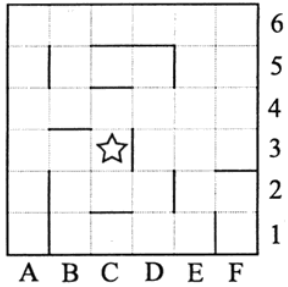
<table>
<tr><th>1</th><th>2</th><th>3</th></tr>
<tr><td></td><td></td><td>1) 00 — началом этого кода является код символа А (0).
2) 10 — является началом кода символов Б (100) и В (101).
3) 111 — не является началом ни одного из заданных кодов символов, ни один из заданных кодов не является началом этого кода.
4) 1111 — не является началом ни одного из заданных кодов символов, ни один из заданных кодов не является началом этого кода.
Кодом буквы Д может быть 111 и 1111. Кодовое слово минимальной длины 111</td></tr>
<tr><td>А10</td><td>2</td><td>Введём обозначения логических высказываний:
А — первая буква согласная;
В — вторая буква согласная;
С — последняя буква гласная.
Получим выражение $(A \to B) \wedge \neg C$.
Запишем значения высказываний А, В, С и построим таблицу истинности:
<table>
<tr><th></th><th>A</th><th>B</th><th>C</th><th>¬ C</th><th>A → B</th><th>(A → B) ∧ ¬ C</th></tr>
<tr><td>НОВГОРОД</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>МЦЕНСК</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>ВОРОНЕЖ</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>РЕВДА</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
</table>
Правильный ответ — МЦЕНСК</td></tr>
</table>

Продолжение таблицы

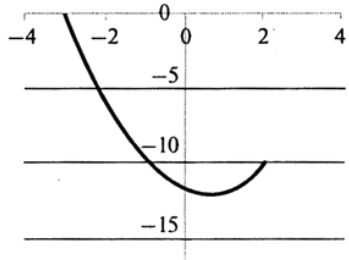
<table>
<tr><th>1</th><th>2</th><th>3</th></tr>
<tr><td>A11</td><td>3</td><td>На один символ потребуется [$\log_2 35$] = 6 бит, тогда на тринадцать символов потребуется 13 · 6 = 78 бит. Минимально возможное количество байтов на тринадцать символов: 78 : 8 = 9,75. Округляя до ближайшего большего целого, получим 10 байт. Под хранение индивидуальных кодов 23 команд потребуется 10 · 23 = 230 байт</td></tr>
<tr><td>A12</td><td>3</td><td>В задаче используются два оператора цикла с параметром. В первом цикле массив A заполняется числами от 0 до 10:
<table>
<tr><td>i</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td></tr>
<tr><td>A[i]</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td></tr>
</table>
Во втором цикле параметр цикла i изменяется с шагом 1 и принимает значения 0, 1, ... 10. Результатом операций «/» в Си, «div» в Паскале и «\» в Бейсике является целая часть частного. В теле цикла меняются местами значения переменных A[i] и A[(10-i)/2], т. е. значения A[0] и A[5], A[1] и A[4], ... A[10] и A[0].
В результате получим:
<table>
<tr><td>i</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td></tr>
<tr><td>A[i]</td><td>10</td><td>8</td><td>6</td><td>2</td><td>3</td><td>1</td><td>0</td><td>4</td><td>7</td><td>5</td><td>9</td></tr>
</table></td></tr>
</table>

1	2	3
A13	1	Максимальное количество клеток, требующих проверки:  Траектории движения РОБОТА:  

Продолжение таблицы

1	2	3
		Клетки, удовлетворяющие условию задачи: 
A14	3	Переменные a, b, d используются для организации цикла. В программе используется функция пользователя $F(x)$, которая возвращает значение выражения $(x + 3)*(x - 4)$. Значение аргумента в теле функции не изменяется. В цикле вычисляется минимальное значение функции $F(t)$. В переменной R сохраняется значение функции $F(t)$, претендующее на минимальное. В переменной M сохраняется значение аргумента t, при котором функция принимает минимальное значение. Таким образом, в программе определяется значение аргумента функции F, при котором функция принимает минимальное значение, на отрезке $[-3; 2]$. Запишем функцию $F(x) = (x + 3)*(x - 4) = x^2 - x - 12$.

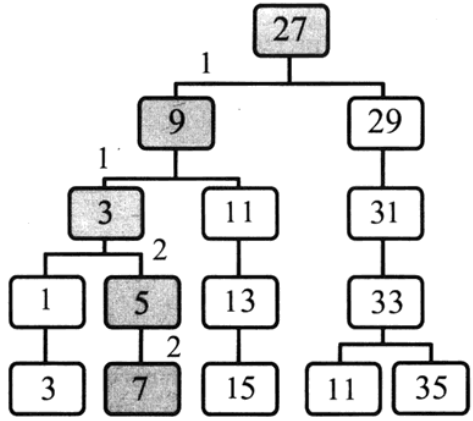
Окончание таблицы

1	2	3
		Получили уравнение параболы ($px^2 + qx + k$), ветви которой направлены вверх ($p>0$), а вершина находится в точке $x_в = -q / 2p = 1/2 = 0{,}5$, $x_в \in [-3; 2]$.  Таким образом, на печать будет выведено число 0,5

Часть 2

Задание	Ответ	Комментарии
1	2	3
B1	128	Объём сообщения увеличился в 2 раза, следовательно, первоначальный файл имел размер 1 Кбит, что равно 2^{10} бит, или $2^{10} / 2^3 = 2^7 = 128$ байт. На каждый символ отводится 8 бит, или 1 байт, значит длина сообщения 128 символов
B2	2211	Не все числа делятся на 3 без остатка, следовательно, количество ветвей и узлов обратного дерева решения (от 27 к 7) будет меньше, чем прямого (от 7 к 27).

Продолжение таблицы

1	2	3
		Построим дерево решения от результата 27 к исходному числу 7 с использованием обратных команд 1. делить на 3 2. прибавить 2  Последовательность команд (выделено цветом) запишем от листа 7 к корню 27

Продолжение таблицы

<table>
<tr><th>1</th><th>2</th><th>3</th></tr>
<tr><td>B3</td><td>6</td><td>
<table>
<tr><th>Команда или условие (логическое выражение)</th><th>a</th><th>b</th></tr>
<tr><td>a := 8</td><td>8</td><td></td></tr>
<tr><td>b := -4</td><td></td><td>–4</td></tr>
<tr><td>не(a = 2) = да</td><td></td><td></td></tr>
<tr><td>(b > 0) = нет</td><td></td><td></td></tr>
<tr><td>a := a div 2</td><td>4</td><td></td></tr>
<tr><td>b := -b</td><td></td><td>4</td></tr>
<tr><td>не (a = 2) = да</td><td></td><td></td></tr>
<tr><td>(b > 0) = да</td><td></td><td></td></tr>
<tr><td>a := a div 2</td><td>2</td><td>4</td></tr>
<tr><td>(a = 2) = да</td><td></td><td></td></tr>
<tr><td>a := -a +2*b = -2 + 8 = 6</td><td>6</td><td></td></tr>
</table>
</td></tr>
<tr><td>B4</td><td>5</td><td>$3^1 + 3^2 + ... + 3^n >= 122$.
Выделим в числе 122 слагаемые — степени тройки, начиная с единицы.
$122 = (3 + 9 + 27 + 81) + 2 = 3^1 + 3^2 + 3^3 + 3^4 + 2$.
Таким образом, $n = 5$</td></tr>
</table>

Продолжение таблицы

<table>
<tr><th>1</th><th>2</th><th>3</th></tr>
<tr><td>B5</td><td>1</td><td>Выполним копирование ячеек, учитывая, что относительные ссылки в формулах меняются при копировании:
<table>
<tr><th></th><th>A</th><th>B</th><th>C</th><th>D</th></tr>
<tr><td>1</td><td>2</td><td>–1</td><td></td><td>–2</td></tr>
<tr><td>2</td><td>= –A1</td><td>=B1*C1</td><td>=C1*D1</td><td>=D1*A1</td></tr>
<tr><td>3</td><td>= –A2</td><td>=B2*C2</td><td>=C2*D2</td><td>=D2*A2</td></tr>
</table>
Проведём возможные вычисления в ячейках таблицы, получим:
<table>
<tr><th></th><th>A</th><th>B</th><th>C</th><th>D</th></tr>
<tr><td>1</td><td>2</td><td>–1</td><td></td><td>–2</td></tr>
<tr><td>2</td><td>–2</td><td>=B1*C1</td><td>=C1*D1</td><td>–4</td></tr>
<tr><td>3</td><td>2</td><td>=B2*C2</td><td>=C2*D2</td><td>8</td></tr>
</table>
Преобразуем формулы:
<table>
<tr><th></th><th>A</th><th>B</th><th>C</th><th>D</th></tr>
<tr><td>1</td><td>2</td><td>–1</td><td></td><td>–2</td></tr>
<tr><td>2</td><td>–2</td><td>= –C1</td><td>=C1*(–2)</td><td>–4</td></tr>
<tr><td>3</td><td>2</td><td>=2*C1*C1</td><td>=C1*8</td><td>8</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>1</th><th>2</th><th>3</th></tr>
<tr><td></td><td></td><td>Из диаграммы следует, что две пары секторов имеют одинаковые значения, равные 2 и 8. Следовательно, сумма значений ячеек B3 и C3 равна 8 + 2 = 10.
Определим C1 из квадратного уравнения
2*C1*C1 + C1*8 – 10 = 0.
Корни уравнения равны 1 и –5. При C1=1 получим B3 = 2 и C3 = 8. При C1 = –5 получим B3 = 50 и C3 = –40.
В ячейке C1 должно быть записано число 1</td></tr>
<tr><td>B6</td><td>–33</td><td>Построим трассировочную таблицу:
<table>
<tr><th>№</th><th>Команда или условие (логическое выражение)</th><th>Вычисление правой части команды присваивания или условия (логического выражения)</th><th>a</th><th>b</th><th>c</th></tr>
<tr><td>1</td><td>a = 18</td><td>18</td><td>18</td><td></td><td></td></tr>
<tr><td>2</td><td>b = 7</td><td>7</td><td></td><td>7</td><td></td></tr>
<tr><td>3</td><td>b > a / 3 – 1</td><td>(7 > 5) = да</td><td></td><td></td><td></td></tr>
<tr><td>4</td><td>c = (b – a) · 3</td><td>(7 – 18) · 3 = –33</td><td></td><td></td><td>–33</td></tr>
<tr><td colspan="3">Результат</td><td>18</td><td>7</td><td>–33</td></tr>
</table></td></tr>
</table>

Продолжение таблицы

1	2	3
B7	52	**Шаг 1.** Исходные значения элементов массива `A`:

i	A[0]	A[1]	A[2]	p
				2
1	p+i=2+1=3			
2		p+i=2+2=4		
3			p+i=2+3=5	
	3	**4**	**5**	**2**

Шаг 2. Трассировка вызова функции `ArrCr(L, A, B, &p)`. В функции получают значения элементы массива `B`:

i	B[0]	B[1]	B[2]	B[3]	B[4]	B[5]	p
							0
0	2*A[0]+p= =2*3+0=6						0
1		2*A[1]+p= =2*4+0=8					1

1	2	3

i	B[0]	B[1]	B[2]	B[3]	B[4]	B[5]	p
2			2*A[0]+p= =2*3+1=7				2
3				2*A[1]+p= =2*4+2=10			3
4					2*A[0]+p= =2*3+3=9		4
5						2*A[1]+p= =2*4+4=12	5
Итог	**6**	**8**	**7**	**10**	**9**	**12**	**5**

Шаг 3. Трассировка вызова функции `Combine(2*L, B)`. Обратите внимание, что в функции для обращения к массиву используется локальная ссылочная переменная `A`. При вызове с этой локальной переменной будет связан массив `B` из вызывающей программы.

Продолжение таблицы

<table>
<tr><th>1</th><th>2</th><th>3</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>i</th><th>N</th></tr>
<tr><td></td><td>0</td></tr>
<tr><td>0</td><td>N +A[0]=0+6=6</td></tr>
<tr><td>1</td><td>N +A[1]=6+8=14</td></tr>
<tr><td>2</td><td>N +A[2]=14+7=21</td></tr>
<tr><td>3</td><td>N +A[3]=21+10=31</td></tr>
<tr><td>4</td><td>N +A[4]=31+9=40</td></tr>
<tr><td>5</td><td>N +A[5]=40+12=52</td></tr>
<tr><td>Итог</td><td>52</td></tr>
</table>
Подробная трассировка не потребуется, если заметить, что в переменной N накапливается сумма элементов массива.
Из последней таблицы видно, что в переменную N вызывающей программы попадёт значение 52, именно оно будет выведено на экран</td></tr>
<tr><td>B8</td><td>679</td><td>Запишем число в N-ичной системе счисления $YZ2_N$, где Y, Z — цифры, которые могут принимать значения от 0 до N – 1.
Развёрнутая запись числа имеет вид:
$Y \cdot N^2 + Z \cdot N + 2 = 128$ или
$N \cdot (Y \cdot N + Z) = 126.$</td></tr>
</table>

1	2	3
		Так как N — один из сомножителей в формуле определения значения числа, а $126 = 2 \cdot 3 \cdot 3 \cdot 7$, то N может принимать значения 2, 3, 6, 7, 9, 14 и т. д. Сразу можно отбросить 2 в качестве оснований систем счисления, так как в записи числа есть цифра 2. Запись минимального трёхзначного числа в любой системе счисления имеет вид 100_N, его значение равно N^2. Для определения максимального значения трёхзначного числа $YZ2_N$ из максимально возможного трёхзначного числа $(N^3 - 1)$ вычтем максимально возможную последнюю цифру $(N - 1)$ и добавим 2, так как младший разряд по условию задачи равен 2: $(N^3 - 1) - (N - 1 + 2) = N^3 - N - 2$. Должно выполняться неравенство $N^2 \leqslant 128 \leqslant N^3 - N - 2$. Проверим выполнение неравенства для возможных значений N (2, 3, 5, 6):

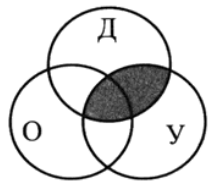
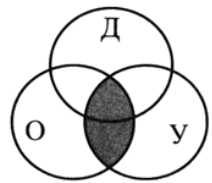
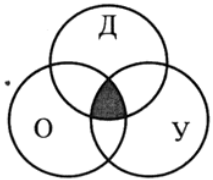
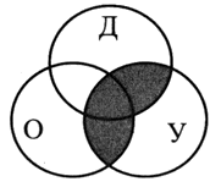
N	$N^2 \leqslant 128$	$128 \leqslant N^3 - N - 2$
3	$27 \leqslant 128$ да	$128 \leqslant 27 - 3 - 2$ нет
6	$36 \leqslant 128$ да	$128 \leqslant 216 - 6 - 2$ да
7	$49 \leqslant 128$ да	$128 \leqslant 343 - 6 - 2$ да

Продолжение таблицы

<table>
<tr><th>1</th><th>2</th><th>3</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>N</th><th>$N^2 \leqslant 128$</th><th>$128 \leqslant N^3 - N - 2$</th></tr>
<tr><td>9</td><td>$81 \leqslant 128$ да</td><td>$128 \leqslant 719 - 6 - 2$ да</td></tr>
<tr><td>14</td><td>$196 \leqslant 128$ нет</td><td>–</td></tr>
</table>
Оба неравенства выполняются для N, равного 6, 7, 9
</td></tr>
<tr><td>B9</td><td>15</td><td>$F(B) = F(2) + F(3) = | F(2) = F(4)+F(5) | =$
$= F(3) + F(4) + F(5) = | F(3) = F(5) | =$
$= F(4) + 2 \cdot F(5) = | F(4) = F(7) + F(8) | =$
$= 2 \cdot F(5) + F(7) + F(8) = | F(5) = F(6) + F(7) | =$
$= 2 \cdot F(6) + 3 \cdot F(7) + F(8) = | F(6) = F(9) | =$
$= 3 \cdot F(7) + F(8) + 2 \cdot F(9) = | F(7) = F(8) + F(9) + F(11) + F(X) | =$
$= 4 \cdot F(8) + 5 \cdot F(9) + 3 \cdot F(11) + 3 \cdot F(X) = | F(8) = F(10);\ \ F(9) = F(11) | =$
$= 4 \cdot F(10) + 8 \cdot F(11) + 3 \cdot F(X) = | F(10) = F(X);\ \ F(11) = F(X) | =$
$= 4 \cdot F(X) + 8 \cdot F(X) + 3 \cdot F(X) = 15$</td></tr>
<tr><td>B10</td><td>8448</td><td>128 Кб = 2^{20} бит
2 Мб = $2 \cdot 2^{23}$ бит
Вычислим время:
$t_1 = 2^{20}$ бит / 2^{12} бит/с = 256 с
$t_2 = 2 \cdot 2^{23}$ бит / 2^{11} бит/с = 8192 с
$t_{общ} = 8448$ с</td></tr>
</table>

1	2	3
B11	GBAD	& 01100000 11000000 01000000 $01000000_2 = 64$ IP-адрес сети: 105.37.27.64
B12	4700	Введём обозначения: Д — множество страниц, полученных по запросу *Диетолог*, О — множество страниц, полученных по запросу *Окулист*, У — множество страниц, полученных по запросу *Уролог*. Чтобы получить число страниц, которые отобразятся в результате запроса (Д | О) & У используем круги Эйлера, каждый из которых будет соответствовать множеству (Д, О и У). Результатом требуемого запроса будет являться закрашенная область. Для получения результата просуммируем количество страниц, найденных по запросу Д & У, с количеством по запросу О & У. Учтём, что пересечение трёх множеств было включено дважды, поэтому вычтем из полученного числа количество страниц, полученных по запросу Д & О & У.

Продолжение таблицы

1	2	3
		 Д & У  О & У  Д & О & У  Результат Результат определяется по формуле: $\lvert$(Д & У)$\rvert$ + $\lvert$(О & У)$\rvert$ – $\lvert$(Д & О & У)$\rvert$ = 4000 + 2300 – 1600 = 4700
B13	8	Числа, полученные в результате выполнения программы, определяются по формуле $K = 4 - 5x + 3(7 - x) = 25 - 8x, \quad x = 0, 1, 2,..., 7,$ где x — количество команд 1 в программе

Продолжение таблицы

1	2	3
B14	6	Для решения логических уравнений вида $F(A, B, C, D, E) = 0$ следует переходить к СКНФ, но если добавить отрицание к обеим частям уравнения, получим $\neg F(A, B, C, D, E) = 1$. Решение подобных уравнений рассмотрено в учебно-справочных материалах и использует переход к СДНФ. $\overline{\overline{(A \to B) \to CDE} \vee \overline{(A\bar{B}) \to \overline{CDE}}} \vee CA = 1$ Используя закон де Моргана, избавимся от отрицания, затем уберём следование: $(\overline{\bar{A} \vee B} \vee CDE)(\overline{A\bar{B}} \vee \overline{CDE})\,\overline{CA} =$ $= (A\bar{B} \vee CDE)(\bar{A} \vee B \vee \bar{C} \vee \bar{D} \vee \bar{E})(\bar{A} \vee \bar{C}) =$ $= (A\bar{B} \vee CDE)(\bar{A} \vee \bar{C})(\bar{A} \vee B \vee \bar{C} \vee \bar{D} \vee \bar{E}).$ Раскроем скобки и применим закон непротиворечия: $(\cancel{A\bar{A}\bar{B}} \vee A\bar{B}\bar{C} \vee \bar{A}CDE \vee \cancel{C\bar{C}DE})(\bar{A} \vee B \vee \bar{C} \vee \bar{D} \vee \bar{E}) =$ $= (A\bar{B}\bar{C} \vee \bar{A}CDE)(\bar{A} \vee B \vee \bar{C} \vee \bar{D} \vee \bar{E}).$ Раскроем скобки и применим закон непротиворечия: $\cancel{A\bar{A}\bar{B}\bar{C}} \vee \bar{A}CDE \vee \cancel{AB\bar{B}\bar{C}} \vee \bar{A}BCDE \vee A\bar{B}\bar{C} \vee \cancel{\bar{A}C\bar{C}DE} \vee A\bar{B}\bar{C}\bar{D} \vee \cancel{\bar{A}CD\bar{D}E} \vee$ $A\bar{B}\bar{C}\bar{E} \vee \cancel{\bar{A}CDE\bar{E}} = \bar{A}CDE \vee \bar{A}BCDE \vee A\bar{B}\bar{C} \vee A\bar{B}\bar{C}\bar{D} \vee A\bar{B}\bar{C}\bar{E}.$

Окончание таблицы

1	2	3
		Применим закон поглощения: $\bar{A}CDE \vee \cancel{\bar{A}BCDE} \vee A\bar{B}\bar{C} \vee \cancel{A\bar{B}\bar{C}\bar{D}} \vee \cancel{A\bar{B}\bar{C}\bar{E}}$. Домножаем неполные конъюнкции для приведения к СДНФ: $\bar{A}CDE(\bar{B} \vee B) \vee A\bar{B}\bar{C}(\bar{D} \vee D)(\bar{E} \vee E) =$ $= \bar{A}\bar{B}CDE \vee \bar{A}BCDE \vee A\bar{B}\bar{C}\bar{D}\bar{E} \vee A\bar{B}\bar{C}\bar{D}E \vee A\bar{B}\bar{C}D\bar{E} \vee A\bar{B}\bar{C}DE$. Количество слагаемых в СДНФ равно 6

Часть 3

Задание	Комментарии
1	2
C1	Пример: $x = -1$, $y = 0{,}5$. При ложном значении условия в любом из первых двух операторов ветвления программа не выводит никакого сообщения. Возможная доработка — заменить все операторы ветвления программы на предложенные:

Продолжение таблицы

1	2

Паскаль

```
if (y>=0) and (x<=0) and (y<= -x*x+4) or
   (y>=x*x) and (y<= -x*x+4) then
   writeln('принадлежит')
   else
   writeln('не принадлежит ');
```

Бейсик

```
IF y>=0 AND x<=0 AND y<= -x*x+4 OR y>=x*x AND y<= -x*x+4 THEN
   PRINT "принадлежит"
   ELSE
   PRINT "не принадлежит"
END IF
```

Си

```
if (((y>=0) && (x<=0) && (y<= -x*x+4)) ||
    ((y>=x*x) && (y<= -x*x+4)))
 printf("принадлежит");
else
 printf("не принадлежит");
```

Продолжение таблицы

1	2
C2	Фрагмент программы с решением.

Паскаль

```
k:=0;
   for i:=1 to n-1 do
            if (a[i] * a[i+1] < 0)   then k := k + 1;
{Умножение чисел с разным знаком - отрицательное число}
{Выводим результат}
writeln(' ',   k);
```

Бейсик

```
K = 0
FOR I = 1 TO N - 1
    IF (A(I) * A(I + 1) < 0) THEN K = K + 1
'Умножение чисел с разным знаком - отрицательное число
NEXT I
'Выводим результат
PRINT K
```

1	2

	Си
	`//Обнуляем счётчик перемен знака`
	`k=0;`
	`//Перебираем пары соседних элементов`
	`for(i=0; i<N-1; i++)`
	`if(a[i] * a[i+1] < 0) k++;`
	`//Умножение чисел с разным знаком - отрицательное число`
	`//Выводим результат`
	`printf("%d",k);`
	
	Естественный язык
	Записываем в переменную K начальное значение, равное 0. В цикле от первого элемента до предпоследнего элемента просматриваем элементы. Если произведение значения текущего элемента на значение следующего элемента меньше 0, то увеличиваем значение переменной K на 1. После завершения цикла выводим K.

C3	Введём функцию $F(k)$, значением которой является число программ Калькулятора, преобразующих число 2 к аргументу функции k. Заметим, что обе команды увеличивают обрабатываемое число. Рассмотрим результат применения каждой команды к исходному числу 2.

Продолжение таблицы

1	2
	После выполнения команды `1. прибавь 1` получим число 3. Никакой другой программой невозможно получить из числа 2 число 3. Следовательно, $F(3) = 1$. Аналогично после выполнения команды `2. умножь на 4` получим число 8. Можно получить из числа 2 число 8 и другой программой — шесть раз выполнить команду 1. Следовательно, $F(8) = 2$ и $F(n) = 1$ при $2 < n < 8$. Составим рекуррентные соотношения, учитывая при этом, что не кратные четырём числа получаются только командой добавления единицы, следовательно, существует одинаковое количество программ получения чисел: $F(4k - 1) = F(4k - 2) = F(4k - 3) = F(4k - 4)$ при $k > 2$. Числа, кратные четырём, могут быть получены двумя заданными командами, следовательно, следует сложить количества программ для двух предыдущих значений аргументов: $F(4k) = F(k) + F(4k - 1) = F(k) + F(4k - 4)$ при $k > 2$; $F(n) = F(n - 1)$ при $n \neq 4k$, $k > 2$. Рекуррентные соотношения имеют вид: $F(n) = F(n - 4)$ при $n \neq 4k$, $k > 2$; $F(n) = F(n/4) + F(n - 4)$ при $n = 4k$, $k > 2$; $F(8) = 2$; $F(n) = 1$ при $2 < n < 8$; $F(n) = 0$ при $n < 3$; $F(23) = F(20) = F(16) + F(5) = F(12) + F(4) + 1 = F(8) + F(3) + 2 = 2 + 1 + 2 = 5$. Ответ: 5

1	2
C4	В программе используется одномерный массив, состоящий из 10 элементов, по количеству компьютеров в сети. Каждый элемент массива имеет тип «запись» (`host`) и состоит из двух полей: строкового для представления имени компьютера `name` и числового для количества переданных компьютером пакетов `summ`. 1. Программа считывает в переменную `N` количество сетевых запросов, которые ей предстоит обработать. 2. В переменную `M` записывается 0 — текущее количество компьютеров в массиве. 3. В цикле от 1 до `N` обрабатываются строки входных данных, соответствующие отдельным сетевым запросам. 3.1. В переменную строкового типа последовательно считываются символы до достижения пробела. 3.2. В переменную числового типа считывается количество пакетов. 3.3. Во внутреннем цикле просматриваются элементы массива до тех пор, пока не встретится такая же строка в поле `name` и переменная цикла `j` меньше или равна `M`. 3.3.1. Если обнаружена такая же строка в поле `name`, то значение поля `summ` соответствующего элемента массива увеличивается на величину считанного числа. 3.3.2. Если переменная цикла `j` стала больше `M`, значение `M` увеличивается на 1, в поле `name` `M`-го элемента массива записывается строка, в поле `summ` — считанное число. 4. Проводится сортировка полученных данных в порядке возрастания значения поля `summ` по алгоритму сортировки «пузырьком». 5. Данные выводятся на экран.

Продолжение таблицы

1	2
	Код программы

	`{ информация о компьютере и пакетах }`
	`type host=record`
	`  name:string[100]; {имя компьютера}`
	`  summ:integer; {суммарное количество пакетов}`
	`  end;`
	`var`
	`  N,i,M,j,packages:integer;`
	`  hosts:array[1..10]of host; {массив записей о компьютерах}`
	`  t:host;`
	`  ch:char;`
	`  s:string[100];{строка для считывания наименования компьютера}`
	`begin`
	`  readln(N); {считываем длину списка запросов}`
	`  M:=0;`
	`  for i:=1 to N do begin`
	`  s:='';`
	`  {считываем имя компьютера}`
	`  read(ch);`
	`  while(ch<>' ')do begin`
	`  s:=s+ch;`
	`  read(ch);`
	`  end;`
	`  {считываем количество пакетов}`
	`  readln(packages);`
	`  j:=1;`

Окончание таблицы

1	2

	{ищем компьютер в списке}
	while((j<M)and(hosts[j].name<>s))do
	inc(j, 1);
	if(hosts[j].name<>s)then begin
	{компьютер не найден - добавляем данные в массив}
	inc(M,1);
	hosts[M].name:=s;
	hosts[M].summ:=packages;
	end
	else
	{компьютер найден - увеличиваем суммарное количество пакетов}
	begin
	hosts[j].summ:=hosts[j].summ+packages;
	end;
	end;
	{сортируем записи в порядке убывания}
	for i:=1 to M do
	for j:=1 to M-i do
	if(hosts [j+1].summ> hosts [j].summ) then begin
	t:= hosts [j];
	hosts [j]:= hosts [j+1];
	hosts [j+1]:=t;
	end
	{выводим список на экран}
	for i:=1 to M do
	writeln(hosts[i].name,' ',hosts[i].summ);
	end.

Вариант 6

Часть 1

Задание	Номер ответа	Комментарии
1	2	3
A1	4	Переведём число $254{,}8_{16}$ в двоичную систему счисления. Каждую шестнадцатеричную цифру заменяем на двоичную тетраду: $001001010100{,}1000_2$. Обратите внимание на то, что нули в начале записи числа и последние нули в дробной части не являются значащими, это число может быть записано в следующем виде: $1001010100{,}1_2$. Число значащих нулей равно 6
A2	3	Вычеркнем все авиарейсы, которые прилетают в аэропорт MSK, вылетают из аэропорта NSB, на которые путешественник не успевает, и те, что прибывают в NSB позже 20:00.

Аэропорт вылета	Аэропорт прилёта	Время вылета	Время прилёта
~~MSK~~	~~SPB~~	~~9:05~~	~~9:40~~
~~SPB~~	~~NSB~~	~~9:35~~	~~12:50~~
MSK	SPB	10:55	14:30
MSK	EKG	11:20	14:45
~~NSB~~	~~MSK~~	~~14:05~~	~~17:00~~
SPB	NSB	14:45	18:05
~~EKG~~	~~MSK~~	~~15:55~~	~~18:50~~

<table>
<tr><th>1</th><th>2</th><th>3</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>Аэропорт вылета</th><th>Аэропорт прилёта</th><th>Время вылета</th><th>Время прилёта</th></tr>
<tr><td>~~MSK~~</td><td>~~NSB~~</td><td>~~17:05~~</td><td>~~20:15~~</td></tr>
<tr><td>EKG</td><td>NSB</td><td>17:35</td><td>18:40</td></tr>
<tr><td>SPB</td><td>NSB</td><td>18:20</td><td>19:15</td></tr>
<tr><td>~~NSB~~</td><td>~~EKG~~</td><td>~~19:05~~</td><td>~~20:20~~</td></tr>
</table>
Определим возможное время вылета из MSK: 10:55 и 11:20. Проверим возможность долететь в NSB, начиная с самого позднего времени вылета.
<table>
<tr><td>MSK</td><td>EKG</td><td>11:20</td><td>14:45</td></tr>
<tr><td>EKG</td><td>NSB</td><td>17:35</td><td>18:40</td></tr>
</table>
</td></tr>
<tr><td>A3</td><td>1</td><td>
<table>
<tr><th>X</th><th>Y</th><th>Z</th><th>F</th><th>$\neg X \wedge Y \wedge Z$</th><th>$X \wedge Y \wedge Z$</th><th>$X \vee \neg Y \vee \neg Z$</th><th>$X \wedge \neg Y \wedge \neg Z$</th></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr>
</table>
Значения выражения ответа № 1 совпадают с **F** на всех заданных наборах переменных
</td></tr>
</table>

Продолжение таблицы

<table>
<tr><th>1</th><th>2</th><th>3</th></tr>
<tr><td>А4</td><td>3</td><td>См. решение варианта 1</td></tr>
<tr><td>А5</td><td>1</td><td>Сумма цифр старших разрядов принадлежит отрезку [2, 18], остальных разрядов — отрезку [0, 18]. Итоговое число состоит из трёх частей. Все части числа расположены в порядке убывания. Этим условиям удовлетворяет число 171210 (можем разбить его на три части, принадлежащие отрезку: 17, 12 и 10, где 17 > 12 > 10)</td></tr>
<tr><td>А6</td><td>2</td><td>По таблице «Товары» определим ID_товара для тетрадей: 1075, 1090, 1189. По таблице «Поступление товаров» определим тетради, поступившие до 01.07.2011:
<table>
<tr><th>ID_товара</th><th>Количество</th><th>Цена за ед., руб.</th><th>Дата</th></tr>
<tr><td>1075</td><td>700</td><td>2,20</td><td>27.06.2011</td></tr>
<tr><td>1189</td><td>1900</td><td>3,80</td><td>29.06.2011</td></tr>
</table>
По таблице «Отгрузка товара» определим тетради, отгруженные до 01.07.2011:
<table>
<tr><th>ID_товара</th><th>Количество</th><th>Цена за ед., руб.</th><th>Дата</th></tr>
<tr><td>1075</td><td>700</td><td>4,20</td><td>30.06.2011</td></tr>
<tr><td>1189</td><td>300</td><td>5,60</td><td>30.06.2011</td></tr>
</table>
Поступление тетрадей до 1 июля: 700 + 1900 = 2600. Продажи тетрадей до 1 июля: 700 + 300 = 1000. Остаток на складе: 2600 – 1000 = 1600</td></tr>
</table>

Продолжение таблицы

1	2	3
A7	2	Следует оценить отношение суммарной заработной платы к общему количеству сотрудников для каждой компании
A8	4	Решение: измерение звука производится 16000 раз в секунду в течение 64 секунд. Для сохранения результата каждого измерения требуется 16 бит = 2 байт. Для стереозвука записываются два сигнала. Вычислим размер файла в мегабайтах: $16000 \cdot 64 \cdot 2 \cdot 2 / 2^{20} = 2^7 \cdot 125 \cdot 2^8 / 2^{20} = 125 / 32 \approx 4$
A9	2	Проверим, являются ли предложенные ответы началом заданных кодовых слов или заданные коды началом кодов, предложенных в ответах. 1) 01 — является началом кода символа Б (011). 2) 11 — не является началом ни одного из заданных кодов символов, ни один из заданных кодов не является началом этого кода. 3) 000 — не является началом ни одного из заданных кодов символов, ни один из заданных кодов не является началом этого кода. 4) 010 — не является началом ни одного из заданных кодов символов, ни один из заданных кодов не является началом этого кода. Таким образом, кодом буквы Д может быть 11, 000 и 010. Кодовое слово минимальной длины 11
A10	2	Введём обозначения логических высказываний: А — есть буква И; В — вторая буква согласная;

Продолжение таблицы

<table>
<tr><th>1</th><th>2</th><th>3</th></tr>
<tr><td></td><td></td><td>С — первая буква согласная.
Получим выражение А (В → С).
Запишем значения логических высказываний для всех ответов и построим таблицу истинности выражения:
<table>
<tr><th></th><th>А</th><th>В</th><th>С</th><th>В → С</th><th>А ∧ (В → С)</th></tr>
<tr><td>ИНДЮК</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>АИСТ</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>ЦАПЛЯ</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>УТКА</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
</table>
Правильный ответ — АИСТ</td></tr>
<tr><td>А11</td><td>2</td><td>Всего символов 28 · 2 + 10 + 1 = 67. На один символ потребуется $\lceil \log_2 67 \rceil = 7$ бит, тогда на десять символов потребуется 10 · 7 = 70 бит. Минимально возможное количество байтов на десять символов: 70 : 8 = 8,75. Округляя до ближайшего большего целого, получим 9 байт. Под хранение 90 номеров студенческих билетов потребуется 9 · 90 = 810 байт</td></tr>
<tr><td>А12</td><td>2</td><td>В задаче используются два оператора цикла с параметром. В первом цикле массив А заполняется числами от 0 до 10:</td></tr>
</table>

1	2	3

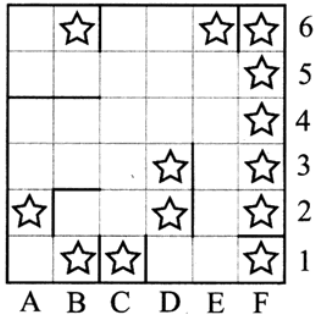
i	0	1	2	3	4	5	6	7	8	9	10
A[i]	0	1	2	3	4	5	6	7	8	9	10

Во втором цикле параметр цикла `i` изменяется с шагом 1 и принимает значения 0, 1, 2, 3, 4. В теле цикла меняются местами значения переменных `A[2*i]` и `A[2*i+1]`, т. е. значения A[0] и A[1], A[2] и A[3], ... A[8] и A[9].
Значение A[10]=10 осталось без изменений. В результате получим:

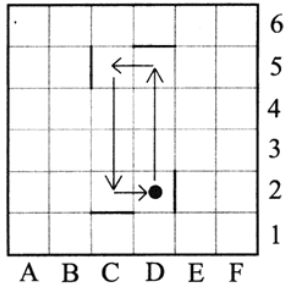
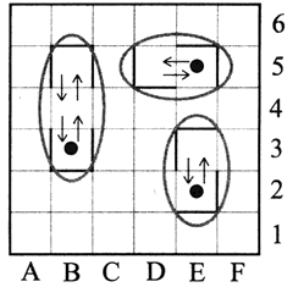
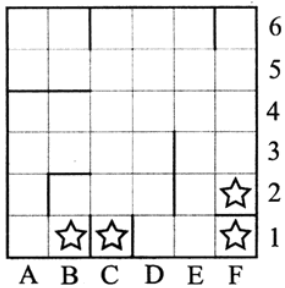
i	0	1	2	3	4	5	6	7	8	9	10
A[i]	1	0	3	2	5	4	7	6	9	8	10

A13 — **4**

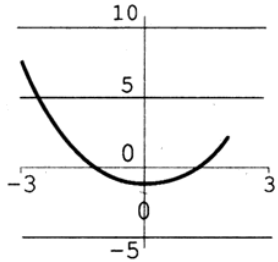
Максимальное количество клеток, требующих проверки:



Продолжение таблицы

1	2	3
		Траектории движения РОБОТА:   Клетки, удовлетворяющие условию задачи: 

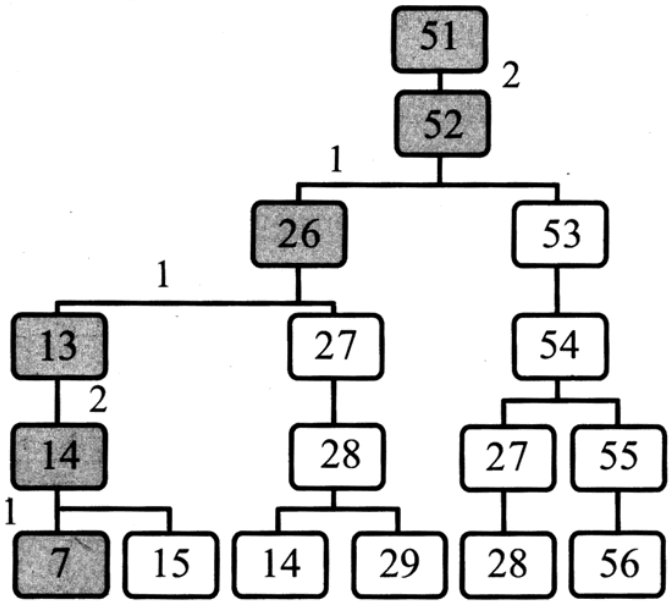
Окончание таблицы

1	2	3
A14	3	Переменные a, b, d используются для организации цикла. В программе используется функция пользователя $F(x)$, которая возвращает значение выражения $(x+1)*(x-1)$. Значение аргумента в теле функции не изменяется. В цикле вычисляется минимальное значение функции $F(t)$. В переменной R сохраняется значение функции $F(t)$, претендующее на минимальное. В переменной M сохраняется значение аргумента t, при котором функция принимает минимальное значение. Таким образом, в программе определяется значение аргумента функции F, при котором функция принимает минимальное значение, на отрезке [–3; 2]. Запишем функцию $F(x) = (x+1)*(x-1) = x^2 - 1$. Получили уравнение параболы ($px^2 + qx + k$), ветви которой направлены вверх ($p > 0$), а вершина находится в точке $x_в = -q / 2p = 0/2 = 0$, $x_в \in [-3; 2]$.  Таким образом, на печать будет выведено число 0

Часть 2

Задание	Ответ	Комментарии
1	2	3
B1	548	Сообщение состоит из 36 символов. Первоначально оно имело объём 36 · 1 байт, после перекодирования — 36 · 2 = 72 байт. К нему прибавили сообщение объёмом 4 Кбит = 4 · 2^{10} бит, или 2^{12} / 2^{3} = 2^{9} байт. 72 + 512 – 36 = 548 байт

B2	12112	Обратные команды
		`1. делить на 2`
		`2. прибавить 1`
		Построим дерево решения от результата 51 к исходному числу 7 с использованием обратных команд. Решение выделено цветом.

1	2	3
		 Запишем последовательность команд получения из числа 7 числа 51 (от листа к корню дерева)

Продолжение таблицы

<table>
<tr><th>1</th><th>2</th><th>3</th></tr>
<tr><td>B3</td><td>11</td><td>Построим трассировочную таблицу, контрольная точка – проверка условия не (a - b < 0):
<table>
<tr><th>a</th><th>b</th><th>не(a - b < 0)</th><th>a > b</th></tr>
<tr><td>8</td><td>5</td><td>да</td><td></td></tr>
<tr><td>7</td><td></td><td>да</td><td>да</td></tr>
<tr><td>6</td><td></td><td>да</td><td>да</td></tr>
<tr><td>5</td><td></td><td>да</td><td>да</td></tr>
<tr><td></td><td>6</td><td>нет</td><td>нет</td></tr>
</table>
a + b = 5 + 6 = 11</td></tr>
<tr><td>B4</td><td>117</td><td>$3^2 + 3^3 + 3^4 = 9 + 27 + 81 = 117$</td></tr>
<tr><td>B5</td><td>–1</td><td>Выполним копирование ячеек, учитывая, что относительные ссылки в формулах меняются при копировании:
<table>
<tr><th></th><th>C</th><th>D</th><th>E</th><th>F</th></tr>
<tr><td>9</td><td>1</td><td>–0,5</td><td>2</td><td></td></tr>
<tr><td>10</td><td>=C9*D9</td><td>=D9*E9</td><td>= –E9</td><td>=F9*C9</td></tr>
<tr><td>11</td><td>=C10*D10</td><td>=D10*E10</td><td>= –E10</td><td>=F10*C10</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>1</th><th>2</th><th>3</th></tr>
<tr><td></td><td></td><td>Проведём возможные вычисления в ячейках таблицы, получим:
<table>
<tr><th></th><th>C</th><th>D</th><th>E</th><th>F</th></tr>
<tr><td>9</td><td>1</td><td>–0,5</td><td>2</td><td></td></tr>
<tr><td>10</td><td>–0,5</td><td>–1</td><td>–2</td><td>=F9</td></tr>
<tr><td>11</td><td>0,5</td><td>2</td><td>2</td><td>= –0,5*F9</td></tr>
</table>
Из диаграммы следует, что в диапазоне C11:F11 есть две пары одинаковых значений, равных 0,5 и 2. Значения ячеек D11 и E11 совпадают и равны 2. Следовательно, значение в ячейке F11 должно быть равно значению в ячейке C11, равному 0,5. Тогда в ячейке F11 должно находиться число –1</td></tr>
<tr><td>B6</td><td>24</td><td>Построим трассировочную таблицу первого вида:
<table>
<tr><th>№</th><th>Команда или условие (логическое выражение)</th><th>Вычисление правой части команды присваивания или условия (логического выражения)</th></tr>
<tr><td>1</td><td>a = 5</td><td>5</td></tr>
<tr><td>2</td><td>b = a + 2 * 2</td><td>9</td></tr>
<tr><td>3</td><td>b >= a - 1</td><td>(9 >= 4) = да</td></tr>
</table></td></tr>
</table>

Продолжение таблицы

<table>
<tr><th>1</th><th>2</th><th>3</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>№</th><th>Команда или условие (логическое выражение)</th><th>Вычисление правой части команды присваивания или условия (логического выражения)</th></tr>
<tr><td>4</td><td>b = b - a</td><td>4</td></tr>
<tr><td>5</td><td>a = a + 1</td><td>6</td></tr>
<tr><td>6</td><td>b >= a - 1</td><td>(4 >= 5) = нет</td></tr>
<tr><td colspan="3">Результат</td></tr>
</table>
6 * 4 = 24
</td></tr>
<tr><td>B7</td><td>79</td><td>
Шаг 1. Определить исходные значения элементов массива A
<table>
<tr><th>i</th><th>A[0]</th><th>A[1]</th><th>A[2]</th><th>A[3]</th><th>p</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>2</td></tr>
<tr><td>0</td><td>2*i+1=
2*0+1=1</td><td></td><td></td><td></td><td></td></tr>
<tr><td>1</td><td></td><td>2*i+1=
2*1+1=3</td><td></td><td></td><td></td></tr>
</table>
</td></tr>
</table>

1	2	3

i	A[0]	A[1]	A[2]	A[3]	p
2			2*i+1= 2*2+1=5		
3				2*i+1= 2*3+1=7	
	1	**3**	**5**	**7**	

Шаг 2. Трассировка вызова функции: `ArrCr(L, A, &p)`

i	A[0]	A[1]	A[2]	A[3]	p
	1	3	5	7	7
1		A[0]+3**p= 1+3*7=22			i%4=1%4=1
2			A[1]+3**p= 22+3*1=25		i%4=2%4=2

Продолжение таблицы

1	2	3

i	A[0]	A[1]	A[2]	A[3]	p
3				A[2]+3**p= 25+3*2=31	i%4=3%4=3
Итог:	**1**	**22**	**25**	**31**	**3**

Шаг 3. Трассировка вызова функции: `Count(L, A)`

i	N
	0
0	N+A[0]=0+1=1
1	N+A[1]=1+22=23
2	N+A[2]=23+25=48
3	N+A[3]=48+31=79
Итог:	**79**

Подробная трассировка не потребуется, если заметить, что в переменной `N` накапливается сумма элементов массива.

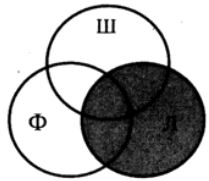
Из последней таблицы видно, что в переменную `N` вызывающей программы попадёт значение 79, именно оно будет выведено на экран

1	2	3
B8	5	Запишем число в N-ичной системе счисления XYZ0N, где X, Y, Z – цифры, которые могут принимать значения от 0 до N–1. Развёрнутая запись числа имеет вид: $X \cdot N^3 + Y \cdot N^2 + Z \cdot N + 2 = 212$ или $N \cdot (X \cdot N^2 + Y \cdot N + Z) = 210.$ N является сомножителем в разложении числа 210 на простые множители ($210 = 2 \cdot 3 \cdot 5 \cdot 7$). Таким образом, N может быть равно 2, 3, 5, 6, 7, 10, 14, ... Основание системы счисления N не может быть равно 2, так как в записи числа есть цифра 2. Также N не может быть равно 10 и более, так как при этом запись числа не может содержать более трёх цифр. Надо проверить, подходят ли варианты решения 3, 5, 6, 7. Для этого следует либо перевести число 212 в системы счисления с основанием 3, 5, 6, 7, либо решить неравенства, как в вар. 1 и вар. 5. $212_{10} = 1322_5$
B9	31	$F(B) = F(2) + F(3) + F(5) = \mid F(2) = F(4) + F(5) \mid =$ $= F(3) + F(4) + 2 \cdot F(5) = \mid F(3) = F(5) + F(6) \mid =$ $= F(4) + 3 \cdot F(5) + F(6) = \mid F(4) = F(5) + F(7) + F(8) \mid =$ $= 4 \cdot F(5) + F(6) + F(7) + F(8) = \mid F(5) = F(6) + F(7) \mid =$ $= 5 \cdot F(6) + 5 \cdot F(7) + F(8) = \mid F(6) = F(9) \mid =$ $= 5 \cdot F(7) + F(8) + 5 \cdot F(9) = \mid F(7) = F(9) + F(10) \mid =$ $= 6 \cdot F(8) + 10 \cdot F(9) + 5 \cdot F(10) = \mid F(8) = F(10) \mid =$ $= 10 \cdot F(9) + 11 \cdot F(10) = \mid F(9) = F(11) + F(X) \mid =$

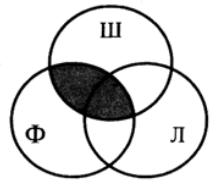
Продолжение таблицы

1	2	3
		$= 11 \cdot F(10) + 10 \cdot F(11) + 10 \cdot F(X) = \lvert F(10) = F(X);\ \ F(11) = F(X) \rvert =$ $= 31 \cdot F(X) = 31$
B10	481	2 Мб = $2 \cdot 2^{23}$ бит 15 Мб = $15 \cdot 2^{23}$ бит Вычислим время: $t_1 = 2 \cdot 2^{23}$ бит / 2^{24} бит/с = 1 с $t_2 = 15 \cdot 2^{23}$ бит / 2^{18} бит/с = 480 с $t_{общ}$ = 481 с
B11	HCDF	& 10101010 11100000 10100000 10100000_2=160 IP-адрес сети: 122.79.115.160
B12	3297	Введём обозначения: Ф – множество страниц, полученных по запросу *Финляндия*, Ш – множество страниц, полученных по запросу *Швеция*, Л – множество страниц, полученных по запросу *Литва*. Для получения результата просуммируем количество страниц, найденных по запросу *Литва*, с количеством страниц по запросу *Швеция & Финляндия*.

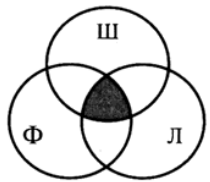
1	2	3
		Следует учесть, что пересечение трёх множеств было включено дважды, поэтому вычтем из полученного числа количество страниц, полученных по запросу ***Финляндия & Швеция & Литва***.



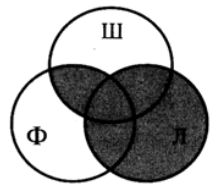
Л



Ш & Ф



Ф & Ш & Л



Результат

Результат определяется по формуле:
|Л| + |(Ш & Ф)| – |(Ф & Ш & Л)| = 2600 + 850 – 153 = 3297

Продолжение таблицы

1	2	3
B13	7	Числа, полученные в результате выполнения программы, определяются по формуле: $k = 7 - 2x + 2(6 - x) = 19 - 4x, \quad x = 0, 1, 2,..., 6,$ где x – количество команд 1 в программе. Таким образом, можно получить 7 разных чисел
B14	8	Выполним преобразования левой части уравнения, перейдём к нормальной форме логического выражения. Убираем импликацию: $((\bar{A} \vee B) \to CDE)\ (CDE \to (\bar{A} \vee B))\ (C \to A) =$ $= (\overline{\bar{A} \vee B} \vee CDE)\ (\bar{C} \vee \bar{D} \vee \bar{E} \vee \bar{A} \vee B)\ (\bar{C} \vee A).$ Убираем отрицание (используем законы де Моргана и двойного отрицания): $(A\bar{B} \vee CDE)\ (\bar{C} \vee \bar{D} \vee \bar{E} \vee \bar{A} \vee B)\ (\bar{C} \vee A).$ Раскроем скобки, используем закон непротиворечия: $(A\bar{B}\bar{C} \vee A\bar{B}\bar{D} \vee A\bar{B}\bar{E} \vee \cancel{A\bar{A}B} \vee \cancel{AB\bar{B}} \vee \cancel{C\bar{C}DE} \vee \cancel{CD\bar{D}E} \vee \cancel{CDE\bar{E}} \vee \bar{A}CDE \vee BCDE) =$ $= (\bar{C} \vee A) = (A\bar{B}\bar{C} \vee A\bar{B}\bar{D} \vee A\bar{B}\bar{E} \vee \bar{A}CDE \vee BCDE)\ (\bar{C} \vee A) =$ $= A\bar{B}\bar{C} \vee \cancel{A\bar{B}\bar{C}} \vee A\bar{B}\bar{C}\bar{D} \vee A\bar{B}\bar{D} \vee A\bar{B}\bar{C}\bar{E} \vee A\bar{B}\bar{E} \vee \cancel{\bar{A}C\bar{C}DE} \vee \cancel{A\bar{A}CDE} \vee \cancel{BC\bar{C}DE} \vee$ $\vee ABCDE = A\bar{B}\bar{C} \vee A\bar{B}\bar{C}\bar{D} \vee A\bar{B}\bar{D} \vee A\bar{B}\bar{C}\bar{E} \vee A\bar{B}\bar{E} \vee ABCDE.$

Окончание таблицы

1	2	3
		Используем закон поглощения: $A\bar{B}\bar{C} \vee \cancel{A\bar{B}\bar{C}\bar{D}} \vee A\bar{B}\bar{D} \vee \cancel{A\bar{B}\bar{C}\bar{E}} \vee A\bar{B}\bar{E} \vee ABCDE =$ $= A\bar{B}\bar{C} \vee A\bar{B}\bar{D} \vee A\bar{B}\bar{E} \vee ABCDE.$ Перейдём к СДНФ: $A\bar{B}\bar{C}\,(\bar{D} \vee D)\,(\bar{E} \vee E) \vee A\bar{B}\bar{D}\,(\bar{C} \vee C)\,(\bar{E} \vee E) \vee A\bar{B}\bar{E}\,(\bar{C} \vee C)\,(\bar{D} \vee D) \vee ABCDE =$ $= A\bar{B}\bar{C}\bar{D}\bar{E} \vee A\bar{B}\bar{C}\bar{D}E \vee A\bar{B}\bar{C}D\bar{E} \vee A\bar{B}\bar{C}DE \vee \cancel{A\bar{B}\bar{C}\bar{D}\bar{E}} \vee \cancel{A\bar{B}\bar{C}\bar{D}E} \vee A\bar{B}C\bar{D}\bar{E} \vee$ $\vee A\bar{B}C\bar{D}E \vee \cancel{A\bar{B}\bar{C}\bar{D}\bar{E}} \vee \cancel{A\bar{B}\bar{C}D\bar{E}} \vee \cancel{A\bar{B}C\bar{D}\bar{E}} \vee A\bar{B}CD\bar{E} \vee ABCDE =$ $= A\bar{B}\bar{C}\bar{D}\bar{E} \vee A\bar{B}\bar{C}\bar{D}E \vee A\bar{B}\bar{C}D\bar{E} \vee A\bar{B}\bar{C}DE \vee A\bar{B}C\bar{D}\bar{E} \vee A\bar{B}C\bar{D}E \vee A\bar{B}CD\bar{E} \vee$ $\vee ABCDE = 1.$ Количество слагаемых в СДНФ (и корней уравнения) равно 8

Часть 3

Задание	Комментарии
1	2
C1	1) Пример: $x = 1$, $y = 0$. При ложном значении условия в любом из первых двух операторов ветвления программа не выводит никакого сообщения. 2) Возможная доработка — заменить все операторы ветвления программы на предложенные:

Паскаль

```
if (y>=0) and (x>=0) and (y<=2-x*x) or
    (x<=0) and (y>=x) and (y<=2-x*x) then
  writeln('принадлежит')
  else
  writeln('не принадлежит');
```

Бейсик

```
IF y>=0 AND x>=0 AND y<=2-x*x OR
   x<=0 AND y >= x AND y<=2-x*x THEN
   PRINT "принадлежит"
   ELSE
   PRINT "не принадлежит"
END IF
```

Продолжение таблицы

1	2

	Си
	`if (((y>=0)&&(x>=0)&&(y<=2-x*x)))||`
	`((x<=0)&&(y>=x)&&(y<=2-x*x)))`
	`printf("принадлежит");`
	`else`
	`printf("не принадлежит");`

C2	**Паскаль**

```
min:=498;
{Перебираем элементы массива}
for i:=1 to N do
    if (a[i] mod 2 = 0) and not (a[i] mod 5 = 0) then
          {Нашли чётный элемент, который не делится на 5}
       if (min > a[i]) then min := a[i];
          {Выводим минимальное значение среди элементов}
        {массива, которые имеют чётное значение и
                                  не делятся на пять}
writeln(' ',  min)
```

Продолжение таблицы

1	2
	Бейсик

	`MIN = 498`
	`FOR I = 1 TO N`
	`  IF (A(I) MOD 2 = 0 AND NOT (A(I) MOD 5 = 0)) THEN`
	`'Нашли чётный элемент, который не делится на 5`
	`  IF(MIN > A(I)) MIN = A(I)`
	`NEXT I`
	`PRINT MIN`

	Си

	`  min = 498;`
	`  for(i=0; i<N; i++)`
	`  if((a[i] % 2 == 0) && (a[i] % 5 != 0))`
	`  // Нашли чётный элемент, который не делится на 5`
	`  if(a[i] < min) min=a[i];`
	`  printf("%d ", min);`

Продолжение таблицы

1	2
	Естественный язык
	Записываем в переменную MIN начальное значение, равное 498. В цикле от начального до последнего элемента последовательно выполняем два сравнения. Если значение текущего элемента чётно и не делится на 5, то сравниваем значение текущего элемента с MIN, если MIN больше значения текущего элемента, заменяем значение MIN на значение текущего элемента. После завершения цикла выводим значение переменной MIN
C3	Введём функцию $F(n)$, значением которой является количество программ Вычислителя, преобразующих число 1 к аргументу функции n. Следует определить, как вычислять функцию для чётных и нечётных аргументов, а также некоторые начальные значения функции. Выполним обе команды для исходного числа: 1) $1 \cdot 2 = 2$; 2) $1 + 1 = 2$. Следовательно, $F(2) = 2$. Нечётные числа можно получить только одной командой — «прибавь 1», следовательно, количество программ получения нечётного числа и ближайшего к нему меньшего чётного одинаковое: $F(2k - 1) = F(2k - 2)$. Чётные числа можно получить как командой 1, так и командой 2:

Продолжение таблицы

1	2
	$F(2k) = F(2k) + F(2k-1) = F(2k) + F(2k-2)$. Таким образом, получены рекуррентные соотношения: $F(2k) = F(2k) + F(2k-2), \quad k > 1;$ $F(2k-1) = F(2k-2), \quad k > 1;$ $F(2) = 2;$ $F(n) = 0, \quad n < 2,$ по которым и будем проводить вычисления: $F(25) = F(24) = F(22) + F(12) =$ $= F(20) + F(12) + F(11) =$ $= F(18) + F(12) + 2 \cdot F(10) =$ $= F(16) + F(12) + 2 \cdot F(10) + F(9) =$ $= F(14) + F(12) + 2 \cdot F(10) + 2 \cdot F(8) =$ $= 2 \cdot F(12) + 2 \cdot F(10) + 2 \cdot F(8) + F(7) =$ $= 4 \cdot F(10) + 2 \cdot F(8) + 3 \cdot F(6) =$ $= 6 \cdot F(8) + 3 \cdot F(6) + 4 \cdot F(5) =$ $= 9 \cdot F(6) + 10 \cdot F(4) =$ $= 19 \cdot F(4) + 9 \cdot F(3) =$ $= 19 \cdot F(2) + 19 \cdot F(2) + 9 \cdot F(2) = 47 \cdot 2 = 94.$ Ответ: 94

Продолжение таблицы

1	2
C4	В программе используются два одномерных массива, состоящих из 12 элементов каждый, по количеству счетов в системе. Один массив (list) состоит из элементов строкового типа для хранения имени счёта, второй массив (Summ) состоит из элементов числового вещественного типа для хранения суммы на счёте. 1. Программа считывает в переменную N количество сетевых запросов, которые ей предстоит обработать. 2. В переменную k записывается 0 – текущее количество счетов в массиве. 3. В цикле от 1 до N обрабатываются строки входных данных, соответствующие отдельным запросам. 3.1. Считывается входная строка в переменную строкового типа buf. 3.2. Из переменной строкового типа buf выделяется имя счёта и записывается в переменную ShetName. 3.3. Из переменной buf удаляется имя сети и запятая. 3.4. Преобразуется оставшаяся часть строки и записывается в числовую переменную c. 3.5. Во внутреннем цикле просматриваются элементы массива до тех пор, пока не встретится такая же строка в массиве list и переменная цикла j меньше или равна k. 3.5.1. Если обнаружена такая же строка в массиве list, то значение соответствующего элемента массива Summ увеличивается на значение переменной c. 3.5.2. Если подобная строка не была найдена, значение k увеличивается на 1, в поле k-го элемента массива list записывается строка ShetName, а в соответствующее поле массива list – значение переменной c.

Продолжение таблицы

1	2
	3.6. После ввода всей входной информации проводится сортировка полученных данных в порядке возрастания значений массива `Summ` по алгоритму сортировки «пузырьком». Одновременно сортируется массив `list`. 4. После сортировки массивов данные выводятся на экран.
	Код программы

	`const M = 12; {Максимальное количество счетов}`
	`var`
	`list:array[1..M] of string[100]; {Список имён счетов}`
	`Summ:array [1..M] of real; {Список сумм остатков средств на каждом из счетов}`
	`buf: string;{Очередная прочитанная строка}`
	`ShetName:string[100];{Имя счёта}`
	`c:real; {буферная переменная - сумма средств}`
	`i,j,`
	`N, {Количество строк исходных данных}`
	`k, {Количество различных наименований}`
	`p, {Номер позиции запятой в строке}`
	`code: integer;`
	`{Признак успешного преобразования строки в число}`
	`flag:boolean;`

Продолжение таблицы

1	2

	{Будем использовать для определения наличия счёта в списке счетов и для определения упорядоченности списка}
	begin
	for i:= 1 to M do begin
	Summ[i]:= 0; {Очистка массива сумм}
	List[i]:=''; {Очистка массива наименований счетов}
	end;
	readln(N); {Считывание количества строк исходных данных}
	k :=0;
	{Обработка строки исходных данных}
	for i:= 1 to N do begin
	readln(buf);
	{Выделение наименования счёта и суммы}
	p:= pos(',', buf);
	{Выделение наименования счёта}
	ShetName:=copy(buf,1,p-1);
	{Удаление из входной строки наименования счёта и пробела}
	delete(buf,1,p);
	{Преобразование оставшейся строки в число}
	val(buf,c,code);
	{Проверка наличия имени счёта в массиве}
	flag := false;
	for j:= 1 to k do begin

Продолжение таблицы

1	2

	`if ShetName = list[j] then begin`
	`flag:= true;`
	`break`
	`end;`
	`end;`
	`if not flag then {Этого имени счёта в списке ещё нет}`
	`begin`
	`k:= k+1;`
	`list[k]:= ShetName;`
	`Summ[k]:= c;`
	`end`
	`else`
	`Summ[j]:= Summ[j] + c;`
	`end; {конец обработки строк исходных данных}`
	`{сортировка списка наименований по возрастанию стоимости}`
	`{используется метод "пузырька" с проверкой упорядоченности на каждом просмотре}`
	`N:=k;`
	`repeat begin`
	`N:= N-1;`
	`flag:= true;`
	`for i:= 1 to N do`
	`if Summ[i] < Summ[i+1] then begin`

Окончание таблицы

1	2

	{если сумма на счёте в следующей позиции больше, чем в текущей, меняем их местами вместе с соответствующими суммами}
	c:=Summ[i];
	Summ[i]:=Summ[i+1];
	Summ[i+1]:=c;
	ShetName:= list[i];
	list[i]:= list[i+1];
	list[i+1]:=ShetName;
	flag:=false;
	end;
	end;
	until flag;
	{Выводим на экран}
	for i:=1 to k do
	writeln(list[i],' ',Summ[i]*0.95:15:2,Summ[i]*0.05:15:2);
	readln;
	end.

СОДЕРЖАНИЕ

Предисловие 3
Инструкция по выполнению работы 5

КОНТРОЛЬНЫЕ ТРЕНИРОВОЧНЫЕ ЗАДАНИЯ 7
Вариант 1 (с решениями) ..... –
Вариант 2 69
Вариант 3 93
Вариант 4 117
Вариант 5 142
Вариант 6 166

ОТВЕТЫ И КОММЕНТАРИИ 190
Вариант 2 ..... –
Вариант 3 222
Вариант 4 251
Вариант 5 279
Вариант 6 308

Тесты для подготовки к ЕГЭ

Авдошин Сергей Михайлович,
Ахметсафина Римма Закиевна,
Максименкова Ольга Вениаминовна

ИНФОРМАТИКА и ИКТ

ЕГЭ 2012

Контрольные тренировочные материалы
с ответами и комментариями

Редактор *Л. И. Кружкова*
Художественный редактор *Л. Г. Епифанов*
Корректоры *Е. Н. Александрова, В. В. Багрий, А. А. Сазонова*

Компьютерная вёрстка *ООО «Аргус»*

Налоговая льгота — Общероссийский классификатор продукции
ОК 005-93-953000. Подписано в печать с оригинал-макета 10.11.2011.
Формат 60×90 $^1/_{16}$. Бумага газетная. Гарнитура Newton, Courier, Рубленая.
Офсетная печать. Усл. печ. 21,5. Уч.-изд. л. 12,31. Тираж 5000 экз. Заказ 1294

Санкт-Петербургский филиал
Открытого акционерного общества
«Издательство «Просвещение».
191014, Санкт-Петербург, Литейный пр., 37-39.

Отпечатано с готовых диапозитивов
в ГУП «Типография «Наука».
199034, Санкт-Петербург, 9-я линия, 12.