

**«УТВЕРЖДАЮ»**  
Директор  
Федерального института  
педагогических измерений



А.Г. Ершов  
2011 г.

**«СОГЛАСОВАНО»**  
Председатель  
Научно-методического совета  
ФИПИ по информатике и ИКТ

В. Б. Бетелин  
« 8 » *август* 2011 г.

**Единый государственный экзамен по ИНФОРМАТИКЕ и ИКТ**

**Демонстрационный вариант  
контрольных измерительных материалов единого  
государственного экзамена 2012 года  
по информатике и ИКТ**

подготовлен Федеральным государственным научным учреждением  
**«ФЕДЕРАЛЬНЫЙ ИНСТИТУТ ПЕДАГОГИЧЕСКИХ ИЗМЕРЕНИЙ»**

**Единый государственный экзамен по ИНФОРМАТИКЕ и ИКТ**

**Пояснения к демонстрационному варианту контрольных  
измерительных материалов единого государственного экзамена  
2012 года по ИНФОРМАТИКЕ и ИКТ**

При ознакомлении с демонстрационным вариантом контрольных измерительных материалов ЕГЭ 2012 г. следует иметь в виду, что задания, в него включённые, не отражают всех вопросов содержания, которые будут проверяться с помощью вариантов КИМ в 2012 г. Полный перечень вопросов, которые могут контролироваться на едином государственном экзамене 2012 г., приведён в кодификаторе элементов содержания и требований к уровню подготовки выпускников общеобразовательных учреждений для единого государственного экзамена 2012 г. по информатике и ИКТ.

Назначение демонстрационного варианта заключается в том, чтобы дать возможность любому участнику ЕГЭ и широкой общественности составить представление о структуре будущих КИМ, количестве заданий, их форме, уровне сложности. Приведённые критерии оценки выполнения заданий с развёрнутым ответом, включённые в этот вариант, дают представление о требованиях к полноте и правильности записи развёрнутого ответа. Эти сведения позволят выпускникам выработать стратегию подготовки к ЕГЭ по информатике и ИКТ.

**Единый государственный экзамен по ИНФОРМАТИКЕ и ИКТ  
Демонстрационный вариант 2012 г.**

**Инструкция по выполнению работы**

На выполнение экзаменационной работы по информатике и ИКТ отводится 4 часа (240 минут). Экзаменационная работа состоит из 3 частей, содержащих 32 задания. На выполнение частей 1 и 2 работы рекомендуется отводить 1,5 часа (90 минут). На выполнение заданий части 3 – 2,5 часа (150 минут).

Часть 1 содержит 13 заданий с выбором ответа. К каждому заданию даётся 4 варианта ответа, из которых только один правильный.

Часть 2 состоит из 15 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.

Все бланки ЕГЭ заполняются яркими черными чернилами. Допускается использование гелевой, капиллярной или перьевой ручек.

При выполнении заданий Вы можете пользоваться черновиком. Обращаем Ваше внимание, что записи в черновике не будут учитываться при оценке работы.

Советуем выполнять задания в том порядке, в котором они даны. Для экономии времени пропускайте задание, которое не удаётся выполнить сразу, и переходите к следующему. Если после выполнения всей работы у Вас останется время, Вы сможете вернуться к пропущенным заданиям.

Баллы, полученные Вами за выполненные задания, суммируются. Постарайтесь выполнить как можно больше заданий и набрать наибольшее количество баллов.

**Желаем успеха!**

В экзаменационных заданиях используются следующие соглашения.

1. Обозначения для логических связок (операций):

- а) *отрицание* (инверсия, логическое НЕ) обозначается  $\neg$  (например,  $\neg A$ );
- б) *конъюнкция* (логическое умножение, логическое И) обозначается  $\wedge$  (например,  $A \wedge B$ ) либо  $\&$  (например,  $A \& B$ );
- в) *дизъюнкция* (логическое сложение, логическое ИЛИ) обозначается  $\vee$  (например,  $A \vee B$ );
- д) *следование* (импликация) обозначается  $\rightarrow$  (например,  $A \rightarrow B$ );
- е) *тождество* обозначается  $\equiv$  (например,  $A \equiv B$ ). Выражение  $A \equiv B$  истинно тогда и только тогда, когда значения  $A$  и  $B$  совпадают (либо они оба истинны, либо они оба ложны);
- ф) символ 1 используется для обозначения истины (истинного высказывания); символ 0 – для обозначения лжи (ложного высказывания).

2. Два логических выражения, содержащих переменные, называются *равносильными* (эквивалентными), если значения этих выражений совпадают при любых значениях переменных. Так, выражения  $A \rightarrow B$  и  $(\neg A) \vee B$  равносильны, а  $A \vee B$  и  $A \wedge B$  – нет (значения выражений разные, например, при  $A = 1, B = 0$ ).

3. Приоритеты логических операций: инверсия (отрицание), конъюнкция (логическое умножение), дизъюнкция (логическое сложение), импликация (следование), 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  $(A \vee B) \vee C$ .

4. Обозначения Мбайт и Кбайт используются в традиционном для информатики смысле – как обозначения единиц измерения, чьё соотношение с единицей "байт" выражается степенью двойки.

**Часть 1**

При выполнении заданий этой части в бланке ответов № 1 под номером выполняемого Вами задания (A1–A13) поставьте знак «x» в клеточке, номер которой соответствует номеру выбранного Вами ответа.

**A1** Сколько единиц в двоичной записи числа 1025?

- 1) 1
- 2) 2
- 3) 10
- 4) 11

**A2** Между населёнными пунктами А, В, С, 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		7	
C	4	1		3	4	
D			3		3	
E		7	4	3		2
F					2	

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

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

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

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

Каким выражением может быть F?

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

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

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

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

В каталоге находятся пять файлов:

```
fort.docx
ford.docx
lord.doc
orsk.dat
port.doc
```

Определите, по какой из масок из них будет отображена указанная группа файлов:

```
fort.docx
ford.docx
lord.doc
port.doc
```

- 1) \*o?\*.\*d?\*
- 2) ?o\*?.d\*
- 3) \*or\*.\*doc?
- 4) ?or?.doc?

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

1. Вычисляются три числа – сумма старших разрядов заданных трехзначных чисел, сумма средних разрядов этих чисел, сумма младших разрядов.
2. Полученные три числа записываются друг за другом в порядке убывания (без разделителей).

*Пример. Исходные трехзначные числа: 835, 196. Поразрядные суммы: 9, 12, 11. Результат: 12119*

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

- 1) 151303
- 2) 161410
- 3) 191615
- 4) 121613

**A6** В фрагменте базы данных представлены сведения о родственных отношениях. Определите на основании приведенных данных фамилию и инициалы бабушки Ивановой А.И.

**Таблица 1**

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

**Таблица 2**

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

- 1) Петренко А.И. 2) Черных И.А. 3) Цейс Т.Н. 4) Петренко Н.Н.

**A7** В ячейке B4 электронной таблицы записана формула = \$C3\*2. Какой вид приобретет формула, после того как ячейку B4 скопируют в ячейку B6?  
Примечание: знак \$ используется для обозначения абсолютной адресации.

- 1) = \$C5 \*4 2) = \$C5 \*2 3) = \$C3 \*4 4) = \$C1 \*2

**A8** Производится одноканальная (моно) звукозапись с частотой дискретизации 16 кГц и 24-битным разрешением. Запись длится 1 минуту, ее результаты записываются в файл, сжатие данных не производится. Какая из приведенных ниже величин наиболее близка к размеру полученного файла?

- 1) 0.2 Мбайт 2) 2 Мбайт 3) 3 Мбайт 4) 4 Мбайт

**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 может быть закодирована буква Д. Код должен удовлетворять свойству однозначного декодирования.

- 1) 00 2) 01 3) 11 4) 010

**A10** Какое из приведённых имён удовлетворяет логическому условию: (первая буква согласная → вторая буква согласная) ∧ (предпоследняя буква гласная → последняя буква гласная)?

- 1) КРИСТИНА 2) МАКСИМ 3) СТЕПАН 4) МАРИЯ

**A11** Для регистрации на сайте некоторой страны пользователю требуется придумать пароль. Длина пароля – ровно 11 символов. В качестве символов используются десятичные цифры и 12 различных букв местного 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 занимает хранение 60 паролей.

- 1) 540 байт 2) 600 байт 3) 660 байт 4) 720 байт

**A12** В программе используется одномерный целочисленный массив А с индексами от 0 до 9. Ниже представлен фрагмент программы, записанный на разных языках программирования, в котором значения элементов сначала задаются, а затем меняются.

Бейсик	Паскаль
<pre>FOR i=0 TO 9   A(i) = 9-i NEXT i FOR i = 0 TO 4   k = A(i)   A(i) = A(9-i)   A(9-i) = k NEXT i</pre>	<pre>for i:=0 to 9 do   A[i] := 9-i; for i:=0 to 4 do begin   k := A[i];   A[i] := A[9-i];   A[9-i] := k; end;</pre>
Си	Алгоритмический язык
<pre>for (i=0;i&lt;=9;i++)   A[i] = 9-i; for (i=0;i&lt;=4;i++) {   k = A[i];   A[i] = A[9-i];   A[9-i] = k; }</pre>	<pre>нц для i от 0 до 9   A[i] := 9-i кц нц для i от 0 до 4   k := A[i]   A[i] := A[9-i]   A[9-i] := k кц</pre>

Чему будут равны элементы этого массива после выполнения фрагмента программы?

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

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

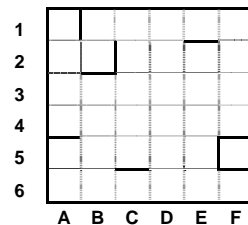
<b>вверх</b>	<b>вниз</b>	<b>влево</b>	<b>вправо</b>
--------------	-------------	--------------	---------------

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:

<b>сверху свободно</b>	<b>снизу свободно</b>	<b>слева свободно</b>	<b>справа свободно</b>
------------------------	-----------------------	-----------------------	------------------------

Цикл  
ПОКА < условие > команда  
выполняется, пока условие истинно, иначе происходит переход на следующую строку.  
Если РОБОТ начнёт движение в сторону находящейся рядом с ним стены, то он разрушится и программа прервётся.  
Сколько клеток лабиринта соответствуют 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) 1
- 2) 3
- 3) 5
- 4) 7

**Часть 2**

*Ответом к заданиям этой части (B1–B15) 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 в 20 символов, первоначально записанного в 2-байтном коде Unicode, в 8-битную кодировку КОИ-8. На сколько бит уменьшилась длина сообщения? В ответе запишите только число.

Ответ: \_\_\_\_\_.

**B2** У исполнителя Утроитель две команды, которым присвоены номера:  
**1. прибавь 1,**  
**2. умножь на 3.**  
Первая из них увеличивает число на экране на 1, вторая – утраивает его. Запишите порядок команд в программе преобразования числа 1 в число 22, содержащей не более 5 команд, указывая лишь номера команд. (Например, **21211** – это программа

**умножь на 3**  
**прибавь 1**  
**умножь на 3**  
**прибавь 1**  
**прибавь 1,**

которая преобразует число 1 в 14.)  
(Если таких программ более одной, то запишите любую из них.)

Ответ: \_\_\_\_\_.

**В3** Определите, что будет напечатано в результате работы следующего фрагмента программы:

Бейсик	Паскаль
<pre>DIM k, s AS INTEGER s = 0 k = 0 WHILE s &lt; 1024   s = s + 10   k = k + 1 WEND PRINT k</pre>	<pre>Var k, s : integer; BEGIN   s := 0;   k := 0;   while s&lt;1024 do     begin       s := s+10;       k := k+1;     end;   write(k); END.</pre>
Си	Алгоритмический язык
<pre>{   int k, s;   s = 0;   k = 0;   while (s&lt;1024) {     s = s+10;     k = k+1;   }   printf("%d", k); }</pre>	<pre>нач   цел k, s   s := 0   k := 0   нц пока s &lt; 1024     s := s+10     k := k+1   кц   вывод k кон</pre>

Ответ: \_\_\_\_\_.

**В4** Все 5-буквенные слова, составленные из букв А, О, У, записаны в алфавитном порядке.

Вот начало списка:

1. ААААА
2. ААААО
3. ААААУ
4. АААОА

.....

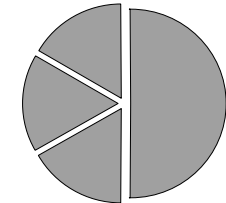
Запишите слово, которое стоит на **240-м месте** от начала списка.

Ответ: \_\_\_\_\_.

**В5** Дан фрагмент электронной таблицы:

	A	B	C	D
1	3		3	2
2	= $(C1+A1)/2$	= $C1-D1$	= $A1-D1$	= $B1/2$

Какое число должно быть записано в ячейке В1, чтобы построенная после выполнения вычислений диаграмма по значениям диапазона ячеек А2:D2 соответствовала рисунку?



Ответ: \_\_\_\_\_.

**В6** Определите значение переменной **c** после выполнения следующего фрагмента программы (записанного ниже на разных языках программирования).

Бейсик	Паскаль
<pre>a = 40 b = 80 b = - a - 2 * b IF a &lt; b THEN   c = b - a ELSE   c = a - 2 * b END IF</pre>	<pre>a := 40; b := 80; b := - a - 2 * b; if a &lt; b then   c := b - a else   c := a - 2 * b;</pre>
Си	Алгоритмический язык
<pre>a = 40; b = 80; b = - a - 2 * b; if (a &lt; b)   c = b - a; else   c = a - 2 * b;</pre>	<pre>а := 40 b := 80 b := - a - 2 * b если a &lt; b   то c := b - a иначе c := a - 2 * b все</pre>

Ответ: \_\_\_\_\_.

**В7**

Ниже на 4-х языках записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает два числа  $L$  и  $M$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 7.

Бейсик	Паскаль
<pre> DIM X, L, M AS INTEGER INPUT X L=0: M=0 WHILE X &gt; 0   L = L+1   IF M &lt; (X MOD 10) THEN     M = X MOD 10   ENDIF   X = X \ 10 WEND PRINT L PRINT M                 </pre>	<pre> var x, L, M: integer; begin   readln(x);   L := 0; M := 0;   while x&gt;0 do     begin       L := L+1;       if M &lt; (x mod 10) then         begin           M := x mod 10;         end;       x := x div 10;     end;   writeln(L); write(M); end.                 </pre>
Си	Алгоритмический язык
<pre> #include&lt;stdio.h&gt; void main() {   int x, L, M;   scanf("%d", &amp;x);   L=0; M=0;   while (x&gt;0){     L=L+1;     if M &lt; x % 10 {       M = x % 10     }     x= x/10;   }   printf("%d\n%d", L, M); }                 </pre>	<pre> алг нач   цел x, L, M   ввод x   L := 0; M := 0   нц пока x&gt;0     L := L+1     если M &lt; mod(x,10)       то         M := mod(x,10)     все     x := div(x,10)   кц   вывод L, M кон                 </pre>

Ответ: \_\_\_\_\_.

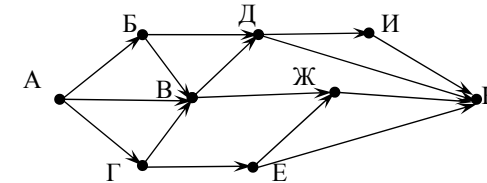
**В8**

Запись числа  $67_{10}$  в системе счисления с основанием  $N$  оканчивается на 1 и содержит 4 цифры. Чему равно основание этой системы счисления  $N$ ?

Ответ: \_\_\_\_\_.

**В9**

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



Ответ: \_\_\_\_\_.

**В10**

У Кати есть доступ в Интернет по высокоскоростному одностороннему радиоканалу, обеспечивающему скорость получения информации  $2^{20}$  бит в секунду. У Сергея нет скоростного доступа в Интернет, но есть возможность получать информацию от Кати по телефонному каналу со средней скоростью  $2^{13}$  бит в секунду. Сергей договорился с Катей, что она скачает для него данные объёмом 9 Мбайт по высокоскоростному каналу и ретранслирует их Сергею по низкоскоростному каналу.

Компьютер Кати может начать ретрансляцию данных не раньше, чем им будут получены первые 1024 Кбайт этих данных. Каков минимально возможный промежуток времени (в секундах) с момента начала скачивания Катей данных до полного их получения Сергеем?

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

Ответ: \_\_\_\_\_.

**В11** 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: 217.233.232.3

Маска: 255.255.252.0

При записи ответа выберите из приведенных в таблице чисел четыре элемента IP-адреса и запишите в нужном порядке соответствующие им буквы. Точки писать не нужно.

A	B	C	D	E	F	G	H
0	3	217	233	232	244	252	255

Пример.

Пусть искомым IP-адрес 192.168.128.0, и дана таблица

A	B	C	D	E	F	G	H
128	168	255	8	127	0	17	192

В этом случае правильный ответ будет записан в виде: HBAF

Ответ: \_\_\_\_\_.

**В12** 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)
Шахматы   Теннис	7770
Теннис	5500
Шахматы & Теннис	1000

Какое количество страниц (в тысячах) будет найдено по запросу Шахматы?

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

Ответ: \_\_\_\_\_.

**В13** У исполнителя Кузнечик 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 Кузнечика – это последовательность команд. Сколько различных чисел можно получить из числа 1 с помощью программы, которая содержит ровно 5 команд?

Ответ: \_\_\_\_\_.

**В14** Определите, какое число будет напечатано в результате выполнения следующего алгоритма (для Вашего удобства алгоритм представлен на четырех языках):

Бейсик	Паскаль
<pre> DIM A, B, T, M, R AS INTEGER A = -20: B = 20 M = A: R = F(A) FOR T = A TO B     IF F(T) &lt; R THEN         M = T         R = F(T)     END IF NEXT T PRINT M  FUNCTION F (x)     F = 4 * (x - 1) * (x - 3) END FUNCTION                     </pre>	<pre> var a,b,t,M,R :integer; Function F(x:integer):integer; begin     F := 4*(x-1)*(x-3); end; BEGIN a := -20; b := 20; M := a; R := F(a); for t := a to b do     begin         if (F(t)&lt;R) then begin             M := t;             R := F(t);         end;     end; write(M); END.                     </pre>
Си	Алгоритмический язык
<pre> int F(int x) {     return 4*(x-1)*(x-3); } void main() {     int a, b, t, M, R;     a = -20; b = 20;     M = a; R = F(a);     for (t=a; t&lt;=b; t++){         if ( F(t)&lt;R ) {             M = t; R = F(t);         }     }     printf("%d", M); }                     </pre>	<pre> алг нач цел a, b, t, M, R a := -20; b := 20 M := a; R:= F(a) нц для t от a до b если F(t)&lt; R то M := t; R := F(t) все кц вывод M кон  алг цел F(цел x) нач знач := 4*(x-1)*(x-3) кон                     </pre>

Ответ: \_\_\_\_\_.



**B15** Сколько существует различных наборов значений логических переменных  $x_1, x_2, \dots, x_9, x_{10}$ , которые удовлетворяют всем перечисленным ниже условиям?

$$\begin{aligned} &((x_1 \equiv x_2) \vee (x_3 \equiv x_4)) \wedge (\neg(x_1 \equiv x_2) \vee \neg(x_3 \equiv x_4)) = 1 \\ &((x_3 \equiv x_4) \vee (x_5 \equiv x_6)) \wedge (\neg(x_3 \equiv x_4) \vee \neg(x_5 \equiv x_6)) = 1 \\ &((x_5 \equiv x_6) \vee (x_7 \equiv x_8)) \wedge (\neg(x_5 \equiv x_6) \vee \neg(x_7 \equiv x_8)) = 1 \\ &((x_7 \equiv x_8) \vee (x_9 \equiv x_{10})) \wedge (\neg(x_7 \equiv x_8) \vee \neg(x_9 \equiv x_{10})) = 1 \end{aligned}$$

В ответе **не нужно** перечислять все различные наборы значений  $x_1, x_2, \dots, x_9, x_{10}$ , при которых выполнена данная система равенств. В качестве ответа вам нужно указать количество таких наборов.

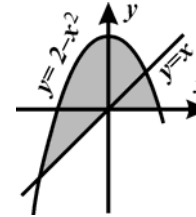
Ответ: \_\_\_\_\_.

*Не забудьте перенести все ответы в бланк ответов № 1.*

**Часть 3**

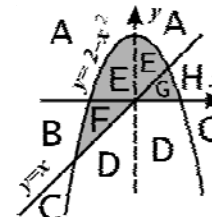
*Для записи ответов на задания этой части (C1–C4) используйте бланк ответов № 2. Запишите сначала номер задания (C1, C2 и т. д.), а затем полное решение. Ответы записывайте чётко и разборчиво.*

**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.

Бейсик	Паскаль
<pre>INPUT x, y IF y&gt;=x THEN   IF y&gt;=0 THEN     IF y&lt;=2-x*x THEN       PRINT "принадлежит"     ELSE       PRINT "не принадлежит"     ENDIF   ENDIF ENDIF END</pre>	<pre>var x,y: real; begin   readln(x,y);   if y&gt;=x then     if y&gt;=0 then       if y&lt;=2-x*x then         write('принадлежит')       else         write('не принадлежит')       end. end.</pre>
Си	Алгоритмический язык
<pre>void main(void) {   float x,y;   scanf("%f%f",&amp;x,&amp;y);   if (y&gt;=x)     if (y&gt;=0)       if (y&lt;=2-x*x)         printf("принадлежит");       else         printf("не принадлежит"); } }</pre>	<pre>алг нач   вещ x,y   ввод x,y   если y&gt;=x то     если y&gt;=0 то       если y&lt;=2-x*x то         вывод 'принадлежит'       иначе         вывод 'не принадлежит'     все   все все кон</pre>



Последовательно выполните следующее.  
 1. Перерисуйте и заполните таблицу, которая показывает, как работает программа при аргументах, принадлежащих различным областям (A, B, C, D, E, F, G и H). Точки, лежащие на границах областей, отдельно не рассматривать.

Область	Условие 1 ( $y >= x$ )	Условие 2 ( $y >= 0$ )	Условие 3 ( $y <= 2 - x * x$ )	Программа выведет	Область обрабатывается верно
A					
B					
C					
D					
E					
F					
G					
H					

В столбцах условий укажите "да", если условие выполняется, "нет" если условие не выполняется, "—" (прочерк), если условие не будет проверяться, «не изв.», если программа ведет себя по-разному для разных значений, принадлежащих данной области. В столбце "Программа выведет" укажите, что программа выведет на экран. Если программа ничего не выводит, напишите "—" (прочерк). Если для разных значений, принадлежащих области, будут выведены разные тексты, напишите «не изв.». В последнем столбце укажите "да" или "нет".

2. Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, достаточно указать любой способ доработки исходной программы.)

**C2**

Дан целочисленный массив из 20 элементов. Элементы массива могут принимать целые значения от 0 до 1000. Опишите на русском 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 три. Гарантируется, что в исходном массиве есть хотя бы один элемент, значение которого чётно и не кратно трем.

Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но использовать все описанные переменные не обязательно.

Паскаль	Алгоритмический язык
<pre>const   N = 20; var   a: array [1..N] of integer;   i, j, min: integer; begin   for i := 1 to N do     readln(a[i]);   ... end.</pre>	<pre>алг нач   цел N = 20   целтаб a[1:N]   цел i, j, MIN   нц для i от 1 до N     ввод a[i]   кц   ... кон</pre>

Бейсик	СИ
<pre>N = 20 DIM A(N) AS INTEGER DIM I, J, MIN AS INTEGER FOR I = 1 TO N   INPUT A(I) NEXT I ... END</pre>	<pre>#include &lt;stdio.h&gt; #define N 20 void main(void){   int a[N];   int i, j, min;   for (i=0; i&lt;N; i++)     scanf("%d", &amp;a[i]);   ... }</pre>
<b>Русский (естественный) язык</b>	
<p>Объявляем массив A из 20 элементов.          Объявляем целочисленные переменные I, J, MIN.          В цикле от 1 до 20 вводим элементы массива A с 1-го по 20-й.          ...</p>	

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 те же самые исходные данные и переменные, какие были предложены в условии (например, в образце, записанном на естественном языке).

**C3**

У исполнителя Утроитель две команды, которым присвоены номера:

1. прибавь 1,
2. умножь на 3.

Первая из них увеличивает число на экране на 1, вторая – утраивает его.

Программа для Утроителя – это последовательность команд.

Сколько есть программ, которые число 1 преобразуют в число 29?

Ответ обоснуйте.

**C4**

В командных олимпиадах по программированию для решения предлагается не больше 11 задач. Команда может решать предложенные задачи в любом порядке. Подготовленные решения команда посылает в единую проверяющую систему соревнований. Вам предлагается написать эффективную, в том числе по используемой памяти, программу, которая будет статистически обрабатывать пришедшие запросы, чтобы определить наиболее популярные задачи. Следует учитывать, что количество запросов в списке может быть очень велико, так как многие соревнования проходят с использованием Интернет.

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 задачи в виде текстовой строки. Длина строки не превосходит 100 символов, название может содержать буквы, цифры, пробелы и знаки препинания.

*Пример входных данных:*

6  
 A+B  
 Крестики-Нолики  
 Прямоугольник  
 Простой делитель  
 A+B  
 Простой делитель

Программа должна вывести список из трёх наиболее популярных задач с указанием количества запросов по ним. Если в запросах упоминаются менее трех задач, то выведите информацию об имеющихся задачах. Если несколько задач имеют ту же частоту встречаемости, что и третья по частоте встречаемости задача, их тоже нужно вывести.

*Пример выходных данных для приведённого выше примера входных данных:*

A+B 2  
 Простой делитель 2  
 Крестики-Нолики 1  
 Прямоугольник 1

### Система оценивания экзаменационной работы по информатике и ИКТ

#### Ответы к заданиям с выбором ответа

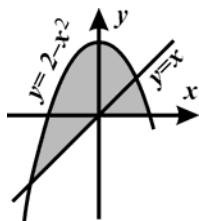
№ задания	Ответ
A1	2
A2	1
A3	1
A4	2
A5	2
A6	3
A7	2
A8	3
A9	4
A10	1
A11	1
A12	2
A13	2

#### Ответы к заданиям с кратким ответом

№ задания	Ответ
B1	160
B2	12121
B3	103
B4	УУУОУ
B5	2
B6	440
B7	777
B8	3
B9	13
B10	9224
B11	CDEA
B12	3270
B13	6
B14	2
B15	64

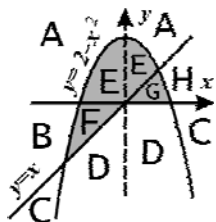
**Критерии оценивания заданий с развернутым ответом**

**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.

Бейсик	Паскаль
<pre>INPUT x, y IF y&gt;=x THEN   IF y&gt;=0 THEN     IF y&lt;=2-x*x THEN       PRINT "принадлежит"     ELSE       PRINT "не принадлежит"     ENDIF   ENDIF ENDIF END</pre>	<pre>var x,y: real; begin   readln(x,y);   if y&gt;=x then     if y&gt;=0 then       if y&lt;=2-x*x then         write('принадлежит')       else         write('не принадлежит')       end. end.</pre>
C++	Алгоритмический язык
<pre>void main(void) {   float x,y;   scanf("%f%f",&amp;x,&amp;y);   if (y&gt;=x)     if (y&gt;=0)       if (y&lt;=2-x*x)         printf("принадлежит");       else         printf("не принадлежит"); }</pre>	<pre>алг нач   вещ x,y   ввод x,y   если y&gt;=x то     если y&gt;=0 то       если y&lt;=2-x*x то         вывод 'принадлежит'       иначе         вывод 'не принадлежит'     все   все кон</pre>



Последовательно выполните следующее.  
 1. Перерисуйте и заполните таблицу, которая показывает, как работает программа при аргументах, принадлежащих различным областям (A, B, C, D, E, F, G и H). Точки, лежащие на границах областей, отдельно не рассматривать.

Область	Условие 1 ( $y \geq x$ )	Условие 2 ( $y \geq 0$ )	Условие 3 ( $y \leq 2 - x^2$ )	Программа выведет	Область обрабатывается верно
A					
B					
C					
D					
E					
F					
G					
H					

В столбцах условий укажите "да", если условие выполнится, "нет" если условие не выполнится, "—" (прочерк), если условие не будет проверяться, «не изв.», если программа ведет себя по-разному для разных значений, принадлежащих данной области. В столбце "Программа выведет" укажите, что программа выведет на экран. Если программа ничего не выводит, напишите "—" (прочерк). Если для разных значений, принадлежащих области, будут выведены разные тексты, напишите «не изв.». В последнем столбце укажите "да" или "нет".

2. Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, достаточно указать любой способ доработки исходной программы.)

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)					
Элементы ответа:					
1.					
Область	Условие 1 ( $y \geq x$ )	Условие 2 ( $y \geq 0$ )	Условие 3 ( $y \leq 2 - x^2$ )	Программа выведет	Область обрабатывается верно
A	да	да	нет	не принадлежит	да
B	да	нет	—	—	нет
C	нет	—	—	—	нет
D	нет	—	—	—	нет
E	да	да	да	принадлежит	да
F	да	нет	—	—	нет
G	нет	—	—	—	нет
H	нет	—	—	—	нет

2. Возможная доработка (Паскаль):  

```
if (x<0) and (y<=2-x*x) and (y>=x) or (x>=0) and (y>=0)
and (y<=2-x*x) then
  write('принадлежит')
else
  write('не принадлежит')
```

 Возможны и другие способы доработки.

Пример: <pre>if (y&lt;=2-x*x) and ((y&gt;=0) or (y&gt;=x)) then   write('принадлежит') else   write('не принадлежит')</pre>	
<b>Указания по оцениванию</b>	<b>Баллы</b>
<p>Обратите внимание! В задаче требуется выполнить три следующих действия.</p> <p>1. Указать для каждой области, как будет работать программа - заполнить таблицу.</p> <p>2. Исправить ошибку, связанную с неверной расстановкой операторных скобок (отсутствуют случаи ELSE). Исправлением этой ошибки может быть либо добавление случая ELSE к каждому условию IF, либо объединение всех условий IF в одно при помощи конъюнкции.</p> <p>В сложных случаях это действие считается выполненным, если программа выдает одно из двух сообщений «принадлежит» или «не принадлежит» для любых чисел <math>x</math> и <math>y</math>, при этом программа не стала работать хуже, чем раньше, то есть для всех точек, для которых программа ранее выдавала верный ответ, доработанная программа также должна выдавать верный ответ.</p> <p>3. Исправить ошибку в проверяемых условиях: приведенным трем ограничениям не удовлетворяют точки плоскости, у которых <math>y \leq 2-x^2</math>, <math>y &gt;= x</math> и <math>y &lt; 0</math> а также <math>y &lt;= 2-x^2</math>, <math>y &lt; x</math> и <math>y &gt;= 0</math>. Исправлением этой ошибки может быть разбиение области на две части и использование дизъюнкции либо отбрасывание от большей области ее части.</p> <p>В сложных случаях это действие считается выполненным, если верно определена закрашенная область, то есть программа выводит сообщение «принадлежит» для всех точек закрашенной области и только для них, для точек вне закрашенной области программа выводит «не принадлежит» или не выводит ничего.</p> <p>В целом, критерий оценки работ такой: количество баллов равно количеству выполненных действий. Детально критерии для каждого случая рассмотрены ниже.</p>	
Задание выполнено полностью: 1) Правильно заполнена вся таблица. 2) Написана правильно работающая программа, т.е. программа для всех пар чисел $x$ , $y$ верно определяет принадлежность точки закрашенной области. В работе (во фрагментах программ) допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора решения.	3

1. Правильно выполнены два действия из трех. При этом: а) Действие по заполнению таблицы засчитывается, даже если ошибки есть, но не более, чем в одной строке. б) При написании операций сравнения допускается одно неправильное использование строгих/нестрогих неравенств (считается несущественной ошибкой, погрешностью записи). Например, вместо « $y >= -x$ » один раз используется « $y > -x$ ». в) Если ученик написал программу, которая отличается от правильной программы (с учетом п.б) только расстановкой скобок в логических выражениях (например, неправильно учтены приоритеты операций), то считается, что выполнено одно действие по исправлению ошибок из двух. В работе (во фрагментах программ) допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора решения.	2
Правильно выполнено только одно действие из трех. При этом: а) Действие по заполнению таблицы засчитывается, если ошибки есть не более, чем в двух строках (в отличие от предыдущего пункта). б) При написании операций сравнения допускается любое количество неправильных использований строгих/нестрогих неравенств (т.е. не учитывается корректность работы программы на границах областей) в) Если ученик написал программу, которая отличается от правильной программы только расстановкой скобок в логических выражениях (например, неправильно учтены приоритеты операций), то считается, что выполнено одно действие по исправлению ошибок из двух. В работе (во фрагментах программ) допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора решения.	1
Ни один из пунктов задания не выполнен, т.е. не выполнены критерии, позволяющие поставить ученику 1 балл.	0
<i>Максимальный балл</i>	3

**C2**

Дан целочисленный массив из 20 элементов. Элементы массива могут принимать целые значения от 0 до 1000. Опишите на русском 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 три. Гарантируется, что в исходном массиве есть хотя бы один элемент, значение которого чётно и не кратно трем.

Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но использовать все описанные переменные не обязательно.

<b>Паскаль</b>	<b>Алгоритмический язык</b>
<pre>const   N = 20; var   a: array [1..N] of integer;   i, j, min: integer; begin   for i := 1 to N do     readln(a[i]);   ... end.</pre>	<pre>алг нач   цел N = 20   целтаб a[1:N]   цел i, j, MIN   нц для i от 1 до N     ввод a[i]   кц   ... кон</pre>
<b>Бейсик</b>	<b>СИ</b>
<pre>N = 20 DIM A(N) AS INTEGER DIM I, J, MIN AS INTEGER FOR I = 1 TO N   INPUT A(I) NEXT I ... END</pre>	<pre>#include &lt;stdio.h&gt; #define N 20 void main(void){   int a[N];   int i, j, min;   for (i=0; i&lt;N; i++)     scanf("%d", &amp;a[i]);   ... }</pre>
<b>Русский (естественный) язык</b>	
<p>Объявляем массив A из 20 элементов.          Объявляем целочисленные переменные I, J, MIN.          В цикле от 1 до 20 вводим элементы массива A с 1-го по 20-й.          ...</p>	

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 те же самые исходные данные и переменные, какие были предложены в условии (например, в образце, записанном на естественном языке).

<b>Содержание верного ответа и указания по оцениванию</b> (допускаются иные формулировки ответа, не искажающие его смысла)	
<b>На языке Паскаль</b>	
<pre>min := 1000; for i:=1 to N do   if (a[i] mod 2=0) and (a[i] mod 3&lt;&gt;0) and (a[i]&lt;min) then     min := a[i]; writeln(min);</pre>	
<b>На Алгоритмическом языке</b>	
<pre>MIN := 1000 нц для i от 1 до N   если mod(a[i],2)=0 и mod(a[i],3)&lt;&gt;0 и a[i]&lt;MIN     то       MIN := a[i]   все кц вывод MIN</pre>	
<b>На языке Бейсик</b>	
<pre>MIN = 1000 FOR I = 1 TO N   IF (A(I) MOD 2 = 0) AND (A(I) MOD 3 &lt;&gt; 0) AND (A(I) &lt; MIN) THEN     MIN = A(I)   END IF NEXT I PRINT MIN</pre>	
<b>На языке СИ</b>	
<pre>min = 1000; for (i=0; i&lt;N; i++)   if (a[i]%2==0 &amp;&amp; a[i]%3!=0 &amp;&amp; a[i]&lt;min)     min = a[i]; printf("%d", min);</pre>	
<b>На естественном языке</b>	
<p>Записываем в переменную MIN начальное значение, равное 1000. В цикле от первого элемента до двадцатого находим остаток от деления элемента исходного массива на два и на три. Если остаток от деления на два равен нулю и остаток от деления на три не равен нулю, то сравниваем значение текущего элемента массива со значением переменной MIN. Если текущий элемент массива меньше MIN, то записываем в MIN значение этого элемента массива. Переходим к следующему элементу.          После завершения цикла выводим значение переменной MIN.</p>	
<b>Указания по оцениванию</b>	<b>Баллы</b>
<p>Предложен правильный алгоритм, выдающий верное значение.          Допускается запись алгоритма на другом языке, использующая аналогичные переменные. В случае, если язык программирования использует типизированные переменные, описания переменных должны быть аналогичны описаниям переменных на естественном языке. Использование нетипизированных или необъявленных переменных возможно только в случае, если это допускается</p>	2

языком программирования, при этом количество переменных и их идентификаторы должны соответствовать условию задачи. В алгоритме, записанном на языке программирования, допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора программы.	
В любом варианте решения может присутствовать не более одной ошибки из числа следующих. <ol style="list-style-type: none"> <li>1. Не инициализируется или неверно инициализируется переменная MIN (например, ей присваивается значение a[1] или число, меньшее 1000).</li> <li>2. Неверно осуществляется проверка чётности элемента массива.</li> <li>3. На чётность проверяется не значение элемента, а его индекс.</li> <li>4. Неверно осуществляется проверка делимости на три.</li> <li>5. На делимость на три проверяется не значение элемента, а его индекс.</li> <li>6. В условии проверки на делимость вместо логической операции «И» используется логическая операция «ИЛИ».</li> <li>7. Отсутствует вывод ответа.</li> <li>8. Используется переменная, не объявленная в разделе описания переменных.</li> <li>9. Не указано или неверно указано условие завершения цикла.</li> <li>10. Индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно.</li> <li>11. Неверно расставлены операторные скобки.</li> </ol>	1
Ошибок, перечисленных в п. 1–11, две или больше, или алгоритм сформулирован неверно.	0
<i>Максимальный балл</i>	2

**С3**

У исполнителя Утроитель две команды, которым присвоены номера:

1. прибавь 1,
2. умножь на 3.

Первая из них увеличивает число на экране на 1, вторая – утраивает его.

Программа для Утроителя – это последовательность команд.

Сколько есть программ, которые число 1 преобразуют в число 29?

Ответ обоснуйте.

<b>Содержание верного ответа и указания к оцениванию</b> (допускаются иные формулировки ответа, не искажающие его смысла)
Обозначим $R(n)$ – количество программ, которые преобразуют число 1 в число $n$ . Обозначим $t(n)$ наибольшее кратное трем, не превосходящее $n$ . Обе команды исполнителя увеличивают исходное число, поэтому общее количество команд в программе не может превосходить 28.

Верны следующие соотношения:

1. Если  $n$  не делится на 3, то тогда  $R(n) = R(t(n))$ , так как существует единственный способ получения  $n$  из  $t(n)$  – прибавлением единиц.
2. Пусть  $n$  делится на 3.

Тогда  $R(n) = R(n/3) + R(n-1) = R(n/3) + R(n-3)$  (если  $n > 3$ ).

При  $n=3$   $R(n) = 2$  (два способа: прибавлением двух единиц или однократным умножением на 3).

Поэтому достаточно постепенно вычислить значения  $R(n)$  для всех чисел, кратных трем и не превосходящих 29: сначала вычисляем  $R(1)$ , затем  $R(3)$ ,  $R(6)$  и т.д.

Имеем:

$$R(2)=1$$

$$R(3) = 2 = R(4)=R(5)$$

$$R(6) = R(2)+R(3) = 1+2 = 3 = R(7)=R(8)$$

$$R(9) = R(3)+R(6) = 2+3 = 5 = R(10)=R(11)$$

$$R(12) = R(4)+R(9) = 2+5 = 7 = R(13)=R(14)$$

$$R(15) = R(5)+R(12) = 2+7 = 9 = R(16)=R(17)$$

$$R(18) = R(6)+R(15) = 3+9 = 12 = R(19)=R(20)$$

$$R(21) = R(7)+R(18) = 3+12 = 15 = R(22)=R(23)$$

$$R(24) = R(8)+R(21) = 3+15 = 18 = R(25)=R(26)$$

$$R(27) = R(9)+R(24) = 5+18 = 23 = R(28)=R(29)$$

Ответ: 23

*Другая форма решения*

Будем решать поставленную задачу последовательно для чисел 1, 2, 3, ..., 29 (то есть для каждого из чисел определим, сколько программ исполнителя существует для его получения). Количество программ, которые преобразуют число 1 в число  $n$ , будем обозначать через  $R(n)$ . Число 1 у нас уже есть, значит, его можно получить с помощью “пустой” программы. Любая непустая программа увеличит исходное число, т.е. даст число, больше 1. Значит,  $R(1) = 1$ . Для каждого следующего числа рассмотрим, из какого числа оно может быть получено за одну команду исполнителя. Если число не делится на три, то оно может быть получено только из предыдущего с помощью команды **прибавь 1**. Значит, количество искомых программ для такого числа равно количеству программ для предыдущего числа:  $R(i) = R(i-1)$ . Если число на 3 делится, то вариантов последней команды два: **прибавь 1** и **умножь на 3**, тогда  $R(i) = R(i-1) + R(i/3)$ . Заполним соответствующую таблицу по приведенным формулам слева направо:

<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>	<b>15</b>
1	1	2	2	2	3	3	3	5	5	5	7	7	7	9
<b>16</b>	<b>17</b>	<b>18</b>	<b>19</b>	<b>20</b>	<b>21</b>	<b>22</b>	<b>23</b>	<b>24</b>	<b>25</b>	<b>26</b>	<b>27</b>	<b>28</b>	<b>29</b>	
9	9	12	12	12	15	15	15	18	18	18	23	23	23	

При этом ячейки, относящиеся к числам, которые не делятся на 3, можно в решении и опустить (за исключением первого и последнего чисел):

<b>1</b>	<b>3</b>	<b>6</b>	<b>9</b>	<b>12</b>	<b>15</b>	<b>18</b>	<b>21</b>	<b>24</b>	<b>27</b>	<b>30</b>	<b>31</b>	
1	2	3	5	7	9	12	15	18	23	28	28	
<b>Указания по оцениванию</b>											<b>Баллы</b>	
Правильное указание количества возможных программ со строгим доказательством правильности (одним из приведенных выше способов или любым другим).											3	
Два балла ставятся в одном из двух случаев: 1. Правильное указание количества возможных программ, основанное на верных рассуждениях, но доказательство правильности неполно. В частности, оценка в 2 балла выставляется в случае, если просто перечислены все правильные программы и не доказано отсутствие других программ, кроме приведенных. 2. Приведены правильные и строгие рассуждения, доведенные до конца, но в вычислениях допущена арифметическая ошибка, в результате чего получен неверный ответ.											2	
Представленное решение обладает одним из свойств 1. Указано, что нужно рассматривать значения $n$ , меньшие, чем 29, и приведены правильные рекуррентные соотношения (см. выше), возможно, неполные. 2. Правильно выписаны и обоснованы значения $R(n)$ для небольших $n$ . 3. Правильно написан ответ, но нет его обоснования.											1	
Не выполнено ни одно из перечисленных выше условий											0	
<i>Максимальный балл</i>											3	

**C4**

В командных олимпиадах по программированию для решения предлагается не больше 11 задач. Команда может решать предложенные задачи в любом порядке. Подготовленные решения команда посылает в единую проверяющую систему соревнований. Вам предлагается написать эффективную, в том числе по используемой памяти, программу, которая будет статистически обрабатывать пришедшие запросы, чтобы определить наиболее популярные задачи. Следует учитывать, что количество запросов в списке может быть очень велико, так как многие соревнования проходят с использованием Интернет.

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 задачи в виде текстовой строки. Длина строки не превосходит 100 символов, название может содержать буквы, цифры, пробелы и знаки препинания.

*Пример входных данных:*

6

A+B  
 Крестики-Нолики  
 Прямоугольник  
 Простой делитель  
 A+B  
 Простой делитель

Программа должна вывести список из трёх наиболее популярных задач с указанием количества запросов по ним. Если в запросах упоминаются менее трех задач, то выведите информацию об имеющихся задачах. Если несколько задач имеют ту же частоту встречаемости, что и третья по частоте встречаемости задача, их тоже нужно вывести.

*Пример выходных данных для приведённого выше примера входных данных:*

A+B 2  
 Простой делитель 2  
 Крестики-Нолики 1  
 Прямоугольник 1

<b>Содержание верного ответа и указания по оцениванию</b> (допускаются иные формулировки ответа, не искажающие его смысла)
Программа читает все входные данные один раз, не запоминая их в массиве, размер которого равен $N$ , а составляя только список встретившихся задач и количества запросов по каждой из них. Во время чтения данных об очередной задаче просматривается список ранее сохранённых задач; если она уже есть в списке, то количество запросов по ней увеличивается на 1, иначе задача добавляется в массив упомянутых в запросах задач (при корректных данных он не может быть больше 11). После окончания ввода производится сортировка массивов задач и количества запросов, отданных за них, в порядке убывания количества запросов, затем выводится список из трёх первых задач с указанием частоты встречаемости (или весь список, если его длина меньше трёх). Вместо сортировки можно применить и алгоритм поиска трёх максимальных элементов в массиве. Затем выводятся задачи, частота встречаемости которых не ниже, чем у третьей задачи. Баллы начисляются только за программу, которая решает задачу хотя бы для одного частного случая. Ниже приведены примеры решения задания на Алгоритмическом языке, а также на языках Паскаль и Бейсик. Допускаются решения, записанные на других языках программирования. При оценивании решений на других языках программирования необходимо учитывать особенности этих языков программирования. Так, на языке C++ при считывании строковой переменной будет считано не все название задачи, а только его первое слово, поэтому следует использовать функцию <code>getline(cin, s)</code> , аналогичная проблема возникает и в языке Си.



**Пример правильной и эффективной программы на языке Паскаль:**

```

Var N, Num, i, j, t: integer;
    Count: array[1..11] of integer;
    s: string;
    Names: array[1..11] of string;
Begin
    Num:=0; {Число различных задач в списке запросов}
    ReadLn(N); {Считываем количество запросов}
    for i:=1 to N do
    begin
        ReadLn(s); {считали очередную задачу}
        {Осуществляем ее поиск в списке уже встретившихся}
        j:=1;
        while (j<=Num) and (s<>Names[j]) do j:=j+1;
        {Если она найдена}
        if j<=Num then {Увеличиваем счетчик числа запросов}
            Count[j]:=Count[j]+1
        else begin {Иначе добавляем задачу в конец списка}
            Names[j]:=s;
            Count[j]:=1;
            Num:=Num+1
        end
    end;
    {Сортируем массивы Names и Count в порядке убывания значений
    массива Count}
    for i:=Num downto 2 do
    for j:=2 to i do if Count[j-1]<Count[j] then
    begin
        t:=Count[j]; Count[j]:=Count[j-1]; Count[j-1]:=t;
        s:=Names[j]; Names[j]:=Names[j-1]; Names[j-1]:=s;
    end;
    if Num >= 3 then j := 3 else j := Num;
    i := 1;
    while (i <= Num) and (Count[i] >= Count[j]) do
    begin
        WriteLn(Names[i], ' ', Count[i]);
        i := i + 1;
    end
end.

```

**Пример правильной и эффективной программы на Алгоритмическом языке:**

```

литтаб Names[1:11] | названия задач
целтаб Count[1:11] | счетчики числа запросов по каждой задаче
цел i, j, t
лит s
|
| 1. Чтение списка запросов
| 1.1. Инициализация количества запросов и счетчика задач
Num:=0 | Число различных задач в списке запросов
ввод N | Считываем количество запросов
| 1.2. Цикл чтения
нц для i от 1 до N
    ввод s | Считали очередную задачу
    | Осуществляем ее поиск в списке уже встретившихся
    j:=1
    нц пока (j<=Num) и (s<>Names[j])
        j:=j+1
    кц
    | Обработываем очередную задачу
    если j<=Num | Если задача найдена в списке
        то | Увеличиваем счетчик числа запросов
            Count[j]:=Count[j]+1
        иначе | Добавляем задачу в конец списка
            Names[j]:=s
            Count[j]:=1
            Num:=Num+1
    все
кц
|
| 2. Совместно сортируем массивы Names и Count
| в порядке убывания значений массива Count
нц для i от Num до 2 шаг -1
    нц для j от 2 до i
        если Count[j-1]<Count[j] то
            t:=Count[j]; Count[j]:=Count[j-1]; Count[j-1]:=t
            s:=Names[j]; Names[j]:=Names[j-1]; Names[j-1]:=s
        все
    кц
кц
|
| 3. Вывод задач-"призеров"
| 3.1. Определение порога для количества запросов по задаче
| Порог равен Count[j]
если Num >= 3
    то j := 3
    иначе j := Num
все
| 3.2. Цикл вывода
i := 1;
нц пока (i <= Num) и (Count[i] >= Count[j])
    вывод нс, Names[i], ' ', Count[i]
    i := i + 1
кц
кон

```

**Пример правильной и эффективной программы на языке Бейсик:**

```

DIM N, Num, i, j, t AS INTEGER
DIM Count(11) AS INTEGER
DIM s AS STRING
DIM Names(11) AS STRING
REM Число различных задач в списке запросов
Num = 0
REM считываем количество запросов
INPUT N
FOR i = 1 TO N
REM считываем очередную задачу
INPUT s
REM осуществляем ее поиск в списке уже встретившихся
j = 1
WHILE j <= Num AND s <> Names(j)
j = j + 1
WEND
IF j <= Num THEN
REM Если она найдена, увеличиваем счетчик числа запросов
Count(j) = Count(j)+1
ELSE
REM Иначе добавляем задачу в конец списка
Names(j) = s: Count(j) = 1
Num = Num + 1
ENDIF
NEXT i
REM Сортируем массивы Names и Count
REM в порядке убывания значений массива Count
FOR i = Num TO 2 Step -1
FOR j =2 TO i
IF Count(j-1) < Count(j) THEN
t = Count(j)
Count(j) = Count(j-1)
Count(j - 1)=t
s = Names(j)
Names(j) = Names(j-1)
Names(j - 1)=s
END IF
NEXT j
NEXT i
REM определение порога для количества появлений
REM задач из списка вывода; порог равен Count(j)
IF Num >= 3 THEN
j = 3
ELSE
j = Num
END IF
i = 1
REM Вывод наиболее популярных задач
WHILE i <= Num AND Count(i) >= Count(j)
PRINT Names(i), Count(i)
i = i + 1
WEND
    
```

Указания по оцениванию	Баллы
<p>Программа работает для любых входных данных произвольного размера и находит ответ, не сохраняя входные данные в массиве, размер которого соответствует числу N (количеству запросов). Программа просматривает входные данные один раз, сохраняя в массиве размером 11 данные о количестве решений, поданных для каждой из встретившихся в списке задач (и учитывает, что в списке их может быть и меньше 11). Допускается наличие в тексте программы одной синтаксической ошибки: пропущен или неверно указан знак пунктуации, неверно написано или пропущено зарезервированное слово языка программирования, не описана или неверно описана переменная, применяется операция, недопустимая для соответствующего типа данных (если одна и та же ошибка встречается несколько раз, то это считается за одну ошибку).</p>	4
<p>Программа работает верно, но входные данные запоминаются в массиве, размер которого соответствует числу N. Этот массив, возможно, потом сортируется. Допускается наличие от одной до трех синтаксических ошибок. Возможно, в принципиально верно организованном вводе данных есть одна ошибка (например, использование read вместо readln в Паскале или неверное считывание строки в C++). Три балла также выставляется, если в эффективной программе, удовлетворяющей критериям выставления 4 баллов, есть одна ошибка, в результате которой программа работает неверно на некоторых наборах нетипичных входных данных (например, все запросы относятся к одной и той же задаче).</p>	3
<p>Программа работает в целом верно, эффективно или нет, но в реализации алгоритма содержится до двух ошибок (неверная инициализация счётчиков – хотя в предложенных выше решениях обнулять их не требуется; возможно, программа неверно работает, если в списке упомянуто меньше 11 задач, выход за границу массива, допущена ошибка в принципиально верно организованной сортировке или алгоритме поиска минимальных элементов, используется знак “&lt;” вместо “&lt;=”, “or” вместо “and” и т. п.). Возможно, некорректно организовано считывание входных данных. Допускается наличие от одной до пяти синтаксических ошибок, описанных выше.</p>	2
<p>Программа, возможно, неверно работает при некоторых входных данных, но по приведённому тексту решения ясно, что экзаменуемый понимает, из каких этапов должно состоять решение задачи. При использовании сортировки она может быть реализована принципиально неверно (например, вместо двух циклов используется один), или допущена принципиальная ошибка в поиске трёх максимальных элементов. Всего допускается до 4 различных ошибок в реализации алгоритма, в том числе описанных в критериях присвоения двух баллов. Допускается наличие от одной до семи синтаксических ошибок, описанных выше.</p>	1
<p>Задание не выполнено или выполнено неверно.</p>	0
<p><i>Максимальный балл</i></p>	4