

*Проект*

**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»**

Демонстрационный вариант ЕГЭ 2012 г. ИНФОРМАТИКА и ИКТ, 11 класс.

(2012 - 2 / 37)

**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 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.

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):

- a) *отрицание* (инверсия, логическое НЕ) обозначается  $\neg$  (например,  $\neg A$ );
- b) *конъюнкция* (логическое умножение, логическое И) обозначается  $\wedge$  (например,  $A \wedge B$ ) либо  $\&$  (например,  $A \& B$ );
- c) *дизъюнкция* (логическое сложение, логическое ИЛИ) обозначается  $\vee$  (например,  $A \vee B$ );
- d) *следование* (импликация) обозначается  $\rightarrow$  (например,  $A \rightarrow B$ );
- e) *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);
- f) 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). 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  $(A \vee B) \vee C$ .

**Часть 1**

При выполнении заданий этой части в бланке ответов № 1 под номером выполняемого Вами задания (A1–A14) 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 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		7	
C	4	1		3	4	
D			3		3	
E		7	4	3		2
F					2	

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

**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) Иванов Т.М.
- 2) Черных И.А.
- 3) Цейс Т.Н.
- 4) Петренко Н.Н.

**A7** Три страны: Королевство Бельгия, Королевство Нидерланды и Великое Герцогство Люксембург образуют экономико-политический союз, который носит название Бенилюкс. Ниже приведен фрагмент электронной таблицы, характеризующий каждую из стран союза и союз в целом:

	A	B	C	D
1	Страна	Население (тыс. чел)	Площадь (кв. км)	Плотность населения (чел / кв.км)
2	Бельгия	10 415	30 528	341
3	Нидерланды	16 357	41 526	394
4	Люксембург	502	2 586	194
5	Бенилюкс целом	27 274	74 640	

Какое значение должно стоять в ячейке D5?

- 1) 365
- 2) 929
- 3) 310
- 4) 2,74

**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. Какое из приведенных ниже чисел наиболее близко к размеру полученного файла, выраженному в мегабайтах?

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

**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) 01
- 3) 11
- 4) 010

**A10** Какое из приведённых имен 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.SetValue(9-i, i) Next For i=0 To 4   K = A.GetValue(i)   A.SetValue(A.GetValue(9-i), i)   A.SetValue(k, 9-i) Next                     </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 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

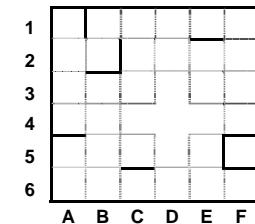
ПОКА <справа свободно> вниз

ПОКА <снизу свободно> влево

ПОКА <слева свободно> вверх

ПОКА <сверху свободно> вправо

КОНЕЦ



- 1) 1  
 2) 3  
 3) 5  
 4) 7

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

Бейсик	Паскаль
<pre>Module A14 Sub Main() Dim d, a, b, t, M, R As Double a = -3 : d = 0.1 d = 0.1 t = a: M = a: R = F(a) While t &lt; b If F(t) &lt; 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</pre>	<pre>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:=-3; b:=3; d:=0.1; t:=a; M:=a; R:=F(a); while t&lt;b do begin if (F(t)&lt;R) then begin M:=t; R:=F(t); end; t:=t+d; end; write(M); END.</pre>
Си	Алгоритмический язык
<pre>#include &lt;stdio.h&gt; double F(double x) { return (x-1)*(x-3); } void main() { double d, a, b, t, M, R; a = -3; b = 3; d = 0.1; t = a; M = a; R = F(a); while (t&lt;b) { if ( F(t)&lt;R ) { M = t; R = F(t); } t = t + d; } printf("%f", M); }</pre>	<pre>алг A14 нач вещ d, a, b, t, M, R a:= -3; b:= 3 d:= 0.1 t:= a; M:= a; R:= F(a) нц пока t&lt;b если F(t)&lt; 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</pre>

- 1) -1                      2) 2                      3) -3                      4) 24

**Часть 2**

*Ответом к заданиям этой части (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 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 End While Console.Write(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
<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

.....

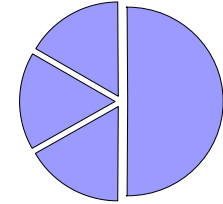
Запишите слово, которое стоит на **240-м месте** от начала списка.

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 = 10 b = - a / 2 * b If a &lt; b Then c = b - a Else c = a - 2 * b End If</pre>	<pre>a := 40; b := 10; 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 = 10; b = - a / 2 * b; if (a &lt; b) c = b - a; else c = a - 2 * b;</pre>	<pre>если a &lt; b то c := b - a иначе c := a - 2 * b все</pre>

Ответ: \_\_\_\_\_.

**B7**

Ниже приведены тексты одной и той же программы, записанные на четырех 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]:=0; R[4]:=1; R[5]:=0;
  Multiply3_2(L, R, p)
  если p>0 то
    вывод "Переполнение"; стоп
  все
  N:=Calc3(L, R)
  вывод N, нс
кон

алг Multiply3_2(цел L, аргрез целтаб R[1:L], рез цел p)
нач
  цел i, n, t
  p:=0
  нц для i от 1 до L
    t:=2*R[i] + p
    R[i]:=mod(t, 3)
    p:= div(t,3)
  кц
кон

алг цел Calc3(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:=T*3
  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(0, 3)
    R.SetValue(1, 4)
    R.SetValue(0, 5)
    Multiply3_2(L, R, p)
    If p > 0 Then
      Console.WriteLine("Переполнение")
      Stop
    End If
    N = Calc3(L, R)
    Console.WriteLine(N)
    Console.WriteLine()
  End Sub

  Sub Multiply3_2(ByVal L As Integer, ByRef R As Array, ByRef
p As Integer)
    Dim i, n, t As Integer
    p = 0
    For i = 1 To L
      t = 2 * R.GetValue(i) + p
      R.SetValue(t Mod 3, i)
      p = t \ 3
    Next
  End Sub

  Function Calc3(ByVal L As Integer, ByRef R As Array) As
Integer
    Dim N, i, T
    N = 0
    T = 1
    For i = 1 To L
      N = N + T * R.GetValue(i)
      T = T * 3
    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 Multiply3_2(L, p : integer; var R : atype );
    var i,n,t : integer;
begin
p:=0;
for i:=1 to L do
  begin
  t:=2*R[i]+p;
  R[i]:= (t) mod(3);
  p:=(t) div(3);
  end;
end;

Function Calc3 (L : integer; 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:=T*3;
  end;
  Calc3:=N;
end;

BEGIN
  R[1]:=2; R[2]:=2; R[3]:=0; R[4]:=1; R[5]:=0;
  Multiply3_2(L, p, R);
  if (p>0) then
  begin
    write(' Переполнение');
    halt;
  end;
  N:=Calc3(L,R);
  write(N);
  writeln;
END.

```

**Си**

```

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

void Multiply3_2(int L, int* R, int *p)
{
  int i, n, t;
  *p = 0;
  for ( i=0; i<L; i++ ) {
    t = 2*R[i] + *p;
    R[i] = t % 3;
    *p = t / 3;
  }
}

int Calc3(int L, int* R)
{
  int N, i, T;
  N = 0;
  T = 1;
  for ( i=0; i<L; i++ ) {
    N = N + T*R[i];
    T = T*3;
  }
  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] = 0; R[3] = 1; R[4] = 0;
  Multiply3_2(L, R, &p);
  if ( p>0 ) {
    printf("Переполнение");
    free(R);
    return;
  }
  N = Calc3(L, R);
  printf("%d\n", N);
  free(R);
}

```

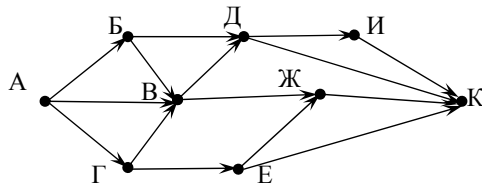
Ответ: \_\_\_\_\_.

**B8**

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

Ответ: \_\_\_\_\_.

**B9** 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: \_\_\_\_\_.

**B10** 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: \_\_\_\_\_.

**B11** В терминологии сетей TCP/IP маской подсети называется 32-разрядное двоичное число, определяющее, какие именно разряды IP-адреса компьютера являются общими для всей подсети - в этих разрядах маски стоит 1. Обычно маски записываются в виде четверки десятичных чисел - по тем же правилам, что и IP-адреса.

Для некоторой подсети используется маска 255.255.252.0.

Сколько различных адресов компьютеров теоретически допускает эта маска?

Примечание. На практике используются не все из этих адресов. Например, как правило, не используются IP-адреса, в десятичном представлении которых последнее (самое правое) число равно 0.

Ответ: \_\_\_\_\_.

**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)
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: \_\_\_\_\_.

**B13** У исполнителя Кузнечик две команды:

1. прибавь 3,
2. вычти 2.

Первая из них увеличивает число на экране на 1, 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: \_\_\_\_\_.

**B14** Сколько различных решений имеет система уравнений

$$((x_1 \equiv x_2) \vee (x_3 \equiv x_4)) \wedge (\neg(x_1 \equiv x_2) \vee \neg(x_3 \equiv x_4)) = 1$$

$$((x_3 \equiv x_4) \vee (x_5 \equiv x_6)) \wedge (\neg(x_3 \equiv x_4) \vee \neg(x_5 \equiv x_6)) = 1$$

...

$$((x_7 \equiv x_8) \vee (x_9 \equiv x_{10})) \wedge (\neg(x_7 \equiv x_8) \vee \neg(x_9 \equiv x_{10})) = 1$$

где  $x_1, x_2, \dots, x_{10}$  – логические переменные?

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

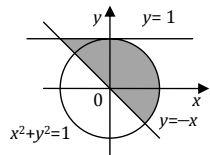
Ответ: \_\_\_\_\_.

**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>Module Program Sub Main() Dim x, y As Double x = Val(Console.ReadLine()) y = Val(Console.ReadLine()) If y&gt;=-x Then If y&lt;=1 Then If x*x+y*y&lt;=1 Then Console.Write("принадлежит") Else Console.Write("не принадлежит") End If End If End If End Sub End Module</pre>	<pre>var x,y: real; begin readln(x,y); if y&gt;=-x then if y&lt;=1 then if (x*x+y*y&lt;=1) then write('принадлежит') else write('не принадлежит') 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&lt;=1) if (x*x+y*y&lt;=1) printf("принадлежит"); else printf("не принадлежит"); }</pre>	<pre>алг нач вещ x,y ввод x,y если y&gt;=-x то если y&lt;=1 то если (x*x+y*y&lt;=1) то вывод 'принадлежит' иначе вывод 'принадлежит' все все все кон</pre>

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 20 элементов. Элементы массива могут принимать целые значения от  $-1000$  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 разрешается не использовать часть из них.

Паскаль	Бейсик
<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>Module program Sub Main() Dim N As Integer = 20 Dim A(N) As Integer Dim I, J, MIN As Integer For I = 1 To N A.SetValue(Val(Console.ReadLine()), I) Next ... End Sub End Module</pre>
СИ	Естественный язык
<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>	<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> <p>...</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**

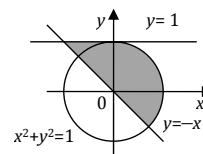
№ задания	Ответ
A1	2
A2	1
A3	1
A4	2
A5	2
A6	3
A7	1
A8	3
A9	4
A10	1
A11	1
A12	2
A13	3
A14	2

**Ответы к заданиям с кратким ответом**

№ задания	Ответ
B1	160
B2	12121
B3	103
B4	УУУОУ
B5	2
B6	440
B7	70
B8	3
B9	13
B10	9224
B11	1024
B12	3270
B13	6
B14	64

**Критерии оценивания заданий с развёрнутым 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.

<b>Бейсик</b>	<b>Паскаль</b>
<pre>Module Program Sub Main() Dim x, y As Double x = Val(Console.ReadLine()) y = Val(Console.ReadLine()) If y &gt;= -x Then If y &lt;= 1 Then If x*x+y*y &lt;= 1 Then Console.Write("принадлежит") Else Console.Write("не принадлежит") End If End If End Sub End Module</pre>	<pre>var x,y: real; begin readln(x,y); if y&gt;=-x then if y&lt;=1 then if (x*x+y*y&lt;=1) then write('принадлежит') else write('не принадлежит') end.</pre>
<b>Си</b>	<b>Алгоритмический язык</b>
<pre>void main(void) { float x,y; scanf("%f%f", &amp;x, &amp;y); if (y&gt;=-x) if (y&lt;=1) if (x*x+y*y&lt;=1) printf("принадлежит"); else printf("не принадлежит"); }</pre>	<pre>алг нач вещ x, y ввод x, y если y &gt;= -x то если y &lt;= 1 то если (x*x+y*y &lt;= 1) то вывод 'принадлежит' иначе вывод 'не принадлежит' все все все кон</pre>

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.)

<b>Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)</b>	
Элементы ответа:	
1. Пример: $x = -1, y = 0$ . Для данной точки программа не выведет никакого сообщения.	
В качестве ответа на первый вопрос подходит любая точка, у которой $y < -x$ или $y > 1$ , для этих точек программа не выведет никакого сообщения. Или точки, у которых $x^2 + y^2 > 1, y \leq 1, y \geq -x, x < 0$ , для этих точек программа выведет сообщение «не принадлежит», а должна вывести сообщение «принадлежит».	
2. Возможная доработка (Паскаль): <pre>if (y&gt;=-x) and (y&lt;=1) and ((x*x+y*y&lt;=1) or (x&lt;=0)) then   write('принадлежит') else   write('не принадлежит')</pre>	
Возможны и другие способы доработки.	
Указания по оцениванию	Баллы
Обратите внимание! В задаче требовалось выполнить <b>три</b> действия: указать пример входных данных, при которых программа работает неверно, и исправить две ошибки. Баллы за данное задание начисляются как сумма баллов за верное выполнение каждого действия.	
1. Верное указание пары чисел $(x, y)$ с пояснением, почему данная пара чисел является ответом на задания. В качестве такого пояснения достаточно указания, что программа не выводит ничего для точек, у которых $y < -x$ или $y > 1$ , или что программа выводит неверное сообщение «Не принадлежит» вместо «Принадлежит» для точек, у которых $x^2 + y^2 > 1, y \leq 1, y \geq -x, x < 0$ . Указание только точки без пояснения считается невыполненным действием, и баллы за данное действие в этом случае не начисляются.	
2. Неправильное использование условного оператора, в результате чего при невыполнении первого или второго условия программа не выдавала ничего (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.	
В сложных случаях это действие считается выполненным, если программа выдаёт одно из двух сообщений «принадлежит» или «не принадлежит» для любых чисел $x$ и $y$ , 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.	

3. Приведённым трём ограничениям не удовлетворяют точки плоскости, у которых $x^2 + y^2 > 1, y \leq 1, y \geq -x, x < 0$ . Исправлением этой ошибки может быть разбиение области на две части и использование дизъюнкции либо отбрасывание от большей области её части. 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.	3
Правильно выполнены оба пункта задания. Исправлены две ошибки. 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.	2
1. Правильно выполнены два действия из трёх (исправлены обе ошибки, но не указан, неправильно указан или указан без обоснования пример требуемых входных данных, либо правильно указан и обоснован пример входных данных, но исправлена только одна ошибка). При написании операций сравнения допускается одно неправильное использование строгих/нестрогих неравенств (считается несущественной ошибкой, погрешностью записи). Например, вместо « $y > -x$ » используется « $y > -x$ ».	1
2. Или выполнены все три действия, но при этом в логическом выражении неверно учтены приоритеты логических операций (не расставлены или неправильно расставлены скобки в выражениях).	0
Правильно выполнено только одно действие из трёх, то есть, либо только приведён и обоснован пример входных данных, либо он не приведён (или приведён неверно или необоснован), но исправлена одна ошибка. При оценивании этого задания на 1 балл допускается не учитывать корректность работы программ на точках границ областей (вместо нестрогих неравенств в решении были использованы строгие неравенства или наоборот).	3
Все пункты задания выполнены неверно (пример входных данных не указан, указан неверно или необоснован, программа не приведена, либо ни одна из двух ошибок не исправлена).	0
<i>Максимальный балл</i>	3

**C2**

Дан целочисленный массив из 20 элементов. Элементы массива могут принимать целые значения от –1000 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 разрешается не использовать часть из них.

<b>Паскаль</b>	<b>Бейсик</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>Module program   Sub Main()     Dim N As Integer = 20     Dim A(N) As Integer     Dim I, J, MIN As Integer     For I = 1 To N       A.SetValue(Val(Console.ReadLine()), I)     Next   ... End Sub End Module</pre>
<b>СИ</b>	<b>Естественный язык</b>
<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>	<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 языке Бейсик</b>	
<pre>MIN = 1000 For I = 1 To N   If A.GetValue(I) Mod 2=0 And A.GetValue(I) Mod 3&lt;&gt;0 And A.GetValue(I)&lt;MIN Then     MIN = A.GetValue(I)   End If Next Console.Write(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>
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 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.	2
В любом варианте решения может присутствовать не более одной	1

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>	
Ошибок, перечисленных в п. 1–11, две или больше, или алгоритм сформулирован неверно.	0
<i>Максимальный балл</i>	2

**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.

<b>Содержание верного ответа и указания к оцениванию</b> (допускаются иные формулировки ответа, не искажающие его смысла)	
<p>Обозначим <math>R(n)</math> – количество программ, которые преобразуют число 1 в число <math>n</math>. Обозначим <math>t(n)</math> наибольшее кратное трем, не превосходящее <math>n</math>. Обе команды исполнителя увеличивают исходное число, поэтому общее количество команд в программе не может превосходить 28.                      Верны следующие соотношения:</p> <ol style="list-style-type: none"> <li>3. Если <math>n</math> не делится на 3, то тогда <math>R(n) = R(t(n))</math>, так как существует единственный способ получения <math>n</math> из <math>t(n)</math> – прибавлением единиц.</li> <li>4. Пусть <math>n</math> делится на 3.                      Тогда <math>R(n) = R(n/3) + R(n-1) = R(n/3) + R(n-3)</math> (если <math>n &gt; 3</math>).                      При <math>n=3</math> <math>R(n) = 2</math> (два способа: прибавлением двух единиц или однократным умножением на 3).                      Поэтому достаточно по индукции вычислить значения <math>R(n)</math> для всех чисел, кратных трем и не превосходящих 29.                      Имеем:  <math>R(2)=1</math>  <math>R(3) = 2 = R(4)=R(5)</math>  <math>R(6) = R(2)+R(3) = 1+2 = 3 = R(7)=R(8)</math>  <math>R(9) = R(3)+R(6) = 2+3 = 5 = R(10)=R(11)</math>  <math>R(12) = R(4)+R(9) = 2+5 = 7 = R(13)=R(14)</math>  <math>R(15) = R(5)+R(12) = 2+7 = 9 = R(16)=R(17)</math>  <math>R(18) = R(6)+R(15) = 3+9 = 12 = R(19)=R(20)</math>  <math>R(21) = R(7)+R(18) = 3+12 = 15 = R(22)=R(23)</math>  <math>R(24) = R(8)+R(21) = 3+15 = 18 = R(25)=R(26)</math>  <math>R(27) = R(9)+R(24) = 5+18 = 23 = R(28)=R(29)</math>                      Ответ: 23</li> </ol>	
Указания по оцениванию	Баллы
Правильное указание количества возможных программ со строгим доказательством правильности (приведенным выше способом или любым другим).	3
Правильное указание количества возможных программ, основанное на верных рассуждениях, но доказательство правильности неполно. В частности, оценка в 2 балла выставляется в случае, если не доказано отсутствие других программ, кроме приведенных (в случае решения методом полного перебора).	2



Представленное решение обладает одним из свойств	1
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 никакого обоснования	
Не выполнено ни одно из перечисленных выше условий	0
<i>Максимальный балл</i>	<i>3</i>

**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)
<p>Программа читает все входные данные один раз, не запоминая их в массиве, размер которого равен <math>N</math>, 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 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.</p> <p><b>Пример правильной и эффективной программы на языке Паскаль:</b></p> <pre> 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&lt;=Num) and (s&lt;&gt;Names[j]) do j:=j+1;             {Если она найдена}             if j&lt;=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;     end; {Сортируем массивы Names и Count в порядке убывания значений} </pre>

```

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 языке Бейсик:**

```

Module C4

Sub Main()
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
n = Val(Console.ReadLine())
For i = 1 To n
REM считываем очередную задачу
s = Console.ReadLine()
REM осуществляем ее поиск в списке уже встретившихся
j = 1
While j <= Num And s <> Names.GetValue(j)
j = j + 1
End While
If j <= Num Then
REM Если она найдена
REM Увеличиваем счетчик числа запросов
Count.SetValue(Count.GetValue(j) + 1, j)
Else
REM Иначе добавляем задачу в конец списка
Names.SetValue(s, j)
Count.SetValue(1, j)
Num = Num + 1
End If
Next

REM Сортируем массивы Names и Count в порядке убывания
значений массива Count
For i = Num To 2 Step -1
t = Count.GetValue(j)
Count.SetValue(Count.GetValue(j - 1), j)

```

```

Count.SetValue(t, j - 1)
s = Names.GetValue(j)
Names.SetValue(Names.GetValue(j - 1), j)
Names.SetValue(s, j - 1)
If Num >= 3 Then
j = 3
Else
j = Num
End If
i = 1
While i <= Num And Count.GetValue(i) >=
Count.GetValue(j)
Console.Write(Names.GetValue(i))
Console.Write(" ")
Console.Write(Count.GetValue(i))
Console.WriteLine()
i = i + 1
End While
Next
End Sub

End Module

```

Указания по оцениванию	Баллы
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 голосов, поданных за каждую 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).	4
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)	3
Программа работает в целом верно, эффективно или нет, но	2

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 “<” вместо “<=”, “or” вместо “and” и т. п.). Возможно, некорректно организовано считывание входных данных. Допускается наличие от одной до пяти синтаксических ошибок, описанных выше.	
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.	1
Задание не выполнено или выполнено неверно.	0
<i>Максимальный балл</i>	<i>4</i>