

**Проект подготовлен к общественно-профессиональному  
обсуждению**

Единый государственный экзамен по ИНФОРМАТИКЕ и ИКТ

**Демонстрационный вариант  
контрольных измерительных материалов единого  
государственного экзамена 2013 года  
по информатике и ИКТ**

подготовлен Федеральным государственным бюджетным  
научным учреждением

«ФЕДЕРАЛЬНЫЙ ИНСТИТУТ ПЕДАГОГИЧЕСКИХ ИЗМЕРЕНИЙ»

Демонстрационный вариант ЕГЭ 2013 г. ИНФОРМАТИКА и ИКТ, 11 класс.

(2013 - 2 / 36)

**Единый государственный экзамен по ИНФОРМАТИКЕ и ИКТ**

**Пояснения к демонстрационному варианту контрольных  
измерительных материалов единого государственного экзамена  
2013 года по ИНФОРМАТИКЕ и ИКТ**

При ознакомлении с демонстрационным вариантом контрольных измерительных материалов ЕГЭ 2013 г. следует иметь в виду, что задания, включённые в него, не отражают всех вопросов содержания, которые будут проверяться с помощью вариантов КИМ в 2013 г. Полный перечень вопросов, которые могут контролироваться на едином государственном экзамене 2013 г., приведён в кодификаторе элементов содержания и требований к уровню подготовки выпускников общеобразовательных учреждений для единого государственного экзамена 2013 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.

**Демонстрационный вариант  
контрольных измерительных материалов  
для проведения в 2013 году единого государственного экзамена  
по ИНФОРМАТИКЕ и ИКТ**

**Инструкция по выполнению работы**

На выполнение экзаменационной работы по информатике и ИКТ отводится 235 минут. Экзаменационная работа состоит из 3 частей, содержащих 32 задания. Рекомендуем не более 1,5 часов (90 минут) отвести на выполнение заданий частей 1 и 2, а остальное время – на часть 3.

Часть 1 содержит 13 заданий (A1–A13). К каждому заданию даётся четыре варианта ответа, из которых только один правильный.

Часть 2 состоит из 15 заданий с кратким ответом (B1–B15). К этим заданиям Вы должны самостоятельно сформулировать и записать ответ.

Часть 3 состоит из 4 заданий (C1–C4). Для выполнения заданий этой части Вам необходимо написать развернутый ответ в произвольной форме.

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

При выполнении заданий Вы можете пользоваться черновиком. Обращаем Ваше внимание на то, что записи в черновике не будут учитываться при оценивании 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  $A \wedge B$  не равносильны (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 «х» в клеточке, номер которой соответствует номеру выбранного Вами ответа.**

**A1** Сколько единиц в двоичной записи десятичного числа 255?

- 1) 1                      2) 2                      3) 7                      4) 8

**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		3				
B	3		7	4	7	
C		7			5	
D		4			2	
E		7	5	2		3
F					3	

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

- 1) 11                      2) 12                      3) 13                      4) 18

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

x1	x2	x3	x4	x5	x6	x7	F
1	1	0	1	1	1	1	0
1	0	1	0	1	1	0	0
0	1	0	1	1	0	0	1

Каким из приведённых ниже выражений может быть F?

- 1)  $\neg x1 \wedge x2 \wedge \neg x3 \wedge x4 \wedge x5 \wedge \neg x6 \wedge \neg x7$   
 2)  $\neg x1 \vee x2 \vee \neg x3 \vee x4 \vee \neg x5 \vee \neg x6 \vee x7$   
 3)  $x1 \wedge \neg x2 \wedge x3 \wedge \neg x4 \wedge x5 \wedge x6 \wedge \neg x7$   
 4)  $x1 \vee \neg x2 \vee x3 \vee \neg x4 \vee \neg x5 \vee x6 \vee \neg x7$

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

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

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

В каталоге находится 6 файлов:

- asc.wma
- casting.wmv
- last.wma
- pasta.wmvx
- pasta.wri
- vast.wma

Определите, по какой из перечисленных масок из этих 6 файлов будет отобрана указанная группа файлов:

- casting.wmv
- last.wma
- pasta.wmvx
- vast.wma

- 1) ?as\*.wm?  
 2) \*as?.wm\*  
 3) ?as\*.wm\*  
 4) ?as\*.w\*

**A5** Автомат получает на вход два двузначных шестнадцатеричных числа. В этих числах все цифры не превосходят цифру 6 (если в числе есть цифра больше 6, автомат отказывается работать). По этим числам строится новое шестнадцатеричное число по следующим правилам.

1. Вычисляются два шестнадцатеричных числа – сумма старших разрядов полученных чисел и сумма младших разрядов этих чисел.

2. Полученные два шестнадцатеричных числа записываются друг за другом в порядке возрастания (без разделителей).

*Пример. Исходные числа: 66, 43. Поразрядные суммы: А, 9. Результат: 9А.*

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

- 1) 9F                      2) 911                      3) 42                      4) 7A

**A6** Ниже представлены две таблицы из базы данных. Каждая строка таблицы 2 содержит информацию о ребёнке и об одном из его родителей. Информация представлена значением поля ID в соответствующей строке таблицы 1. Определите на основании приведённых данных фамилию и инициалы внучки Петровой С.М.

Таблица 1		
ID	Фамилия И.О.	Пол
25	Жвания К.Г.	Ж
49	Черняк А.П.	М
62	Петрова М.Н.	Ж
76	Ильченко Т.В.	Ж
82	Петрова С.М.	Ж
96	Басис В.В.	Ж
102	Ильченко В.И.	М
123	Павлыш Н.П.	Ж
134	Черняк П.Р.	М
...	...	...

Таблица 2	
ID Родителя	ID Ребёнка
25	134
76	49
76	123
82	76
82	96
102	76
102	96
134	49
134	123
...	...

- 1) Басис В.В.
- 2) Ильченко Т.В.
- 3) Павлыш Н.П.
- 4) Петрова М.Н.

**A7** Дан фрагмент электронной таблицы.

	A	B	C	D
<b>1</b>	1	2	3	
<b>2</b>	5	4	= \$A\$2 + B\$3	
<b>3</b>	6	7	= A3 + B3	

Чему станет равным значение ячейки D1, если в неё скопировать формулу из ячейки C2?

*Примечание: знак \$ обозначает абсолютную адресацию.*

- 1) 18
- 2) 12
- 3) 14
- 4) 17

**A8** Производится одноканальная (моно) цифровая звукозапись. Значение сигнала фиксируется 48 000 раз в секунду, для записи каждого значения используется 32 бит. Запись длится 4 минуты, её результаты записываются в файл, сжатия данных не производится. Какая из приведённых ниже величин наиболее близка к размеру полученного файла?

- 1) 44 Мбайт
- 2) 87 Мбайт
- 3) 125 Мбайт
- 4) 175 Мбайт

**A9** Для кодирования некоторой последовательности, состоящей из букв А, Б, В, Г и Д, используется неравномерный двоичный код, позволяющий однозначно декодировать полученную двоичную последовательность. Вот этот код: А – 00, Б – 01, В – 100, Г – 101, Д – 110. Можно ли сократить для одной из букв длину кодового слова так, чтобы код по-прежнему можно было декодировать однозначно? Коды остальных букв меняться не должны. Выберите правильный вариант ответа.

- 1) для буквы Д – 11
- 2) это невозможно
- 3) для буквы Г – 10
- 4) для буквы Д – 10

**A10** Для какого из приведённых чисел X истинно логическое условие:  $\neg ((X \text{ кратно } 2) \rightarrow (X \text{ кратно } 4))$ ?

- 1) 7
- 2) 8
- 3) 10
- 4) 12

**A11** В некоторой стране автомобильный номер длиной 5 символов составляют из заглавных букв (задействовано 30 различных букв) и любых десятичных цифр в любом порядке.

Каждый такой номер в компьютерной программе записывается минимально возможным и одинаковым целым количеством байт (при этом используют посимвольное кодирование и все символы кодируются одинаковым и минимально возможным количеством бит).

Определите объём памяти, отводимый этой программой для записи 50 номеров.

- 1) 100 байт
- 2) 150 байт
- 3) 200 байт
- 4) 250 байт

**A12** В программе описан одномерный целочисленный массив А с индексами от 0 до 10. Ниже представлен фрагмент этой программы, записанный на разных языках программирования, в котором значения элементов массива сначала задаются, а затем меняются.

Бейсик	Паскаль
<pre>FOR i=0 TO 10 A(i)=i-1 NEXT i FOR i=1 TO 10 A(i-1)=A(i) NEXT i A(10)=10</pre>	<pre>for i:=0 to 10 do   A[i]:=i-1; for i:=1 to 10 do   A[i-1]:=A[i]; A[10]:=10;</pre>
Си	Алгоритмический язык
<pre>for (i=0;i&lt;=10;i++)   A[i]=i-1; for (i=1;i&lt;=10;i++)   A[i-1]=A[i]; A[10]=10;</pre>	<pre>нц для i от 0 до 10   A[i]:=i-1 кц нц для i от 1 до 10   A[i-1]:=A[i] кц A[10]:=10</pre>

Как изменятся элементы этого массива после выполнения фрагмента программы?

- 1) все элементы, кроме последнего, окажутся равны между собой
- 2) все элементы окажутся равны своим индексам
- 3) все элементы, кроме последнего, будут сдвинуты на один элемент вправо
- 4) все элементы, кроме последнего, уменьшатся на единицу

**A13** Система команд исполнителя РОБОТ, «живущего» в прямоугольном лабиринте на клетчатой плоскости, включает в себя 4 команды-приказа и 4 команды проверки условия.

Команды-приказы:

вверх	вниз	влево	вправо
-------	------	-------	--------

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

Если РОБОТ начнёт движение в сторону находящейся рядом с ним стены, то он разрушится, и программа прервётся.

Другие 4 команды проверяют истинность *условия* отсутствия стены у каждой стороны той клетки, где находится РОБОТ:

сверху свободно	снизу свободно	слева свободно	справа свободно
-----------------	----------------	----------------	-----------------

Цикл

ПОКА <условие >  
    *последовательность команд*

КОНЕЦ ПОКА

выполняется, пока условие истинно.

В конструкции

ЕСЛИ <условие >

    ТО команда1

    ИНАЧЕ команда2

КОНЕЦ ЕСЛИ

выполняется *команда1* (если условие истинно) или *команда2* (если условие ложно).

Сколько клеток лабиринта соответствуют требованию, что, начав движение в ней и выполнив предложенную программу, РОБОТ уцелеет и остановится в закрашенной клетке (клетка A1)?

НАЧАЛО

ПОКА <слева свободно ИЛИ сверху свободно >

    ЕСЛИ <слева свободно >

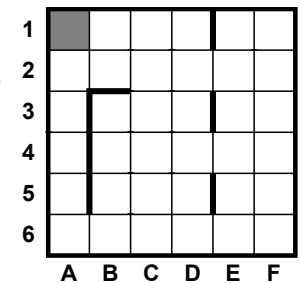
        ТО влево

        ИНАЧЕ вверх

    КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ



1) 8

2) 12

3) 17

4) 21

**Часть 2**

*Ответом к заданиям этой части (В1–В15) 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.*

**В1** У исполнителя Арифметик две команды, которым присвоены номера:

1. прибавь 2,
2. умножь на 3.

Первая из них увеличивает число на экране на 2, вторая утраивает его.

Например, 21211 – это программа

- умножь на 3  
 прибавь 2  
 умножь на 3  
 прибавь 2  
 прибавь 2,

которая преобразует число 1 в число 19.

Запишите порядок команд в программе преобразования числа 3 в число 69, содержащей не более 5 команд, указывая лишь номера команд. Если таких программ более одной, то запишите любую из них.

Ответ: \_\_\_\_\_.

**В2** Определите значение переменной *c* после выполнения следующего фрагмента программы (записанного ниже на разных языках программирования). Ответ запишите в виде целого числа.

Бейсик	Паскаль
<pre>a = 30 b = 14 a = a - 2 * b IF a &gt; b THEN     c = b + 2 * a ELSE     c = b - 2 * a ENDIF</pre>	<pre>a := 30; b := 14; a := a - 2 * b; if a &gt; b then     c := b + 2 * a else     c := b - 2 * a;</pre>
Си	Алгоритмический
<pre>a = 30; b = 14; a = a - 2 * b; if (a &gt; b)     c = b + 2 * a; else     c = b - 2 * a;</pre>	<pre>a := 30 b := 14 a := a - 2 * b если a &gt; b     то c := b + 2 * a     иначе c := b - 2 * a все</pre>

Ответ: \_\_\_\_\_.

**В3** Дан фрагмент электронной таблицы.

	A	B	C
1	2	4	
2	$= (B1 - A1) / 2$	$= 2 - A1 / 2$	$= (C1 - A1) * 2 - 4$

Какое целое число должно быть записано в ячейке C1, чтобы построенная после выполнения вычислений диаграмма по значениям диапазона ячеек A2 : C2 соответствовала рисунку?



Известно, что все значения диапазона, по которым построена диаграмма, имеют один и тот же знак.

Ответ: \_\_\_\_\_.

**B4** Азбука Морзе позволяет кодировать символы для сообщений по радиосвязи, задавая комбинацию точек и тире. Сколько различных символов (цифр, букв, знаков пунктуации и т. д.) можно закодировать, используя код азбуки Морзе длиной **не менее четырёх** и **не более пяти** сигналов (точек и тире)?

Ответ: \_\_\_\_\_.

**B5** Определите, что будет напечатано в результате выполнения программы (записанной ниже на разных языках программирования).

Бейсик	Паскаль
<pre>DIM N, S AS INTEGER N = 0 S = 0 WHILE S &lt;= 35   N = N + 1   S = S + 4 WEND PRINT N</pre>	<pre>var n, s: integer; begin   n := 0;   s := 0;   while s &lt;= 35 do     begin       n := n + 1;       s := s + 4     end;   write(n) end.</pre>
Си	Алгоритмический
<pre>#include&lt;stdio.h&gt; void main() {   int n, s;   n = 0;   s = 0;   while (s &lt;= 35)   {     n = n + 1;     s = s + 4;   }   printf("%d", n); }</pre>	<pre><u>алг</u> <u>нач</u>   <u>цел</u> n, s   n := 0   s := 0   <u>нц пока</u> s &lt;= 35     n := n + 1     s := s + 4   <u>кц</u>   <u>вывод</u> n <u>кон</u></pre>

Ответ: \_\_\_\_\_.

**B6** Алгоритм вычисления значения функции  $F(n)$ , где  $n$  – натуральное число, задан следующими соотношениями:  
 $F(1) = 1$   
 $F(n) = F(n-1) * n$ , при  $n > 1$   
 Чему равно значение функции  $F(5)$ ?  
*В ответе запишите только натуральное число.*

Ответ: \_\_\_\_\_.

**B7** Запись десятичного числа в системах счисления с основаниями 3 и 5 в обоих случаях имеет последней цифрой 0. Какое минимальное натуральное десятичное число удовлетворяет этому требованию?

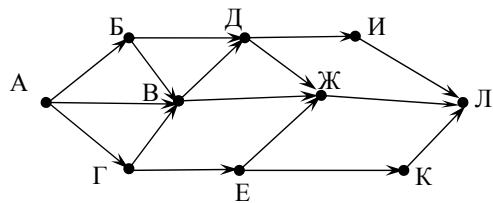
Ответ: \_\_\_\_\_.

**B8** Ниже на четырёх языках записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 21.

Бейсик	Паскаль
<pre>DIM X, A, B AS INTEGER INPUT X A=0: B=1 WHILE X &gt; 0   A = A+1   B = B*(X MOD 10)   X = X \ 10 WEND PRINT A PRINT B</pre>	<pre>var x, a, b: integer; begin   readln(x);   a:=0; b:=1;   while x&gt;0 do     begin       a:=a+1;       b:=b*(x mod 10);       x:= x div 10     end;   writeln(a); write(b); end.</pre>
Си	Алгоритмический
<pre>#include&lt;stdio.h&gt; void main() {   int x, a, b;   scanf("%d", &amp;x);   a=0; b=1;   while (x&gt;0){     a=a+1;     b=b*(x%10);     x= x/10;   }   printf("%d\n%d", a, b); }</pre>	<pre><u>алг</u> <u>нач</u>   <u>цел</u> x, a, b   <u>ввод</u> x   a:=0; b:=1   <u>нц пока</u> x&gt;0     a:=a+1     b:=b*mod(x,10)     x:=div(x,10)   <u>кц</u>   <u>вывод</u> a, <u>нс</u>, b <u>кон</u></pre>

Ответ: \_\_\_\_\_.

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



Ответ: \_\_\_\_\_.

**В10** Документ объёмом 20 Мбайт можно передать с одного компьютера на другой двумя способами.  
 А. Сжать архиватором, передать архив по каналу связи, распаковать.  
 Б. Передать по каналу связи без использования архиватора.

Какой способ быстрее и насколько, если:

- средняя скорость передачи данных по каналу связи составляет  $2^{20}$  бит в секунду;
- объём сжатого архиватором документа равен 20% исходного;
- время, требуемое на сжатие документа, – 5 секунд, на распаковку – 1 секунда?

В ответе напишите букву А, если быстрее способ А, или Б, если быстрее способ Б. Сразу после буквы напишите число, обозначающее, на сколько секунд один способ быстрее другого.

Так, например, если способ Б быстрее способа А на 23 секунды, в ответе нужно написать Б23.

Единиц измерения «секунд», «сек.», «с.» к ответу 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: 217.19.128.131

Маска: 255.255.192.0

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

A	B	C	D	E	F	G	H
0	16	19	64	128	131	192	217

*Пример.*

*Пусть искомый 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)
Фрегат   Эсминец	3400
Фрегат & Эсминец	900
Фрегат	2100

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

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 Удвоитель две команды, которым присвоены номера:

**1. прибавь 1,**

**2. умножь на 2.**

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

Программа для Удвоителя – это последовательность команд.

Сколько есть программ, которые число 3 преобразуют в число 23?

Ответ: \_\_\_\_\_.

**B14** Определите, какое число будет напечатано в результате выполнения следующего алгоритма (для Вашего удобства алгоритм представлен на четырёх 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)     ENDIF NEXT T PRINT M  FUNCTION F(x)     F = 3*(x-8)*(x-8) END FUNCTION                     </pre>	<pre> var a,b,t,M,R :integer; Function F(x:integer):integer; begin     F := 3*(x-8)*(x-8) 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
<pre> #include&lt;stdio.h&gt; int F(int x) {     return 3*(x-8)*(x-8); } 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, R, M     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 := 3*(x-8)*(x-8) кон                     </pre>

Ответ: \_\_\_\_\_.

**B15** Сколько существует различных наборов значений логических переменных  $x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4$ , которые удовлетворяют всем перечисленным ниже условиям?

$$(x_1 \rightarrow x_2) \wedge (x_2 \rightarrow x_3) \wedge (x_3 \rightarrow x_4) = 1$$

$$(\neg y_1 \vee y_2) \wedge (\neg y_2 \vee y_3) \wedge (\neg y_3 \vee y_4) = 1$$

$$(y_1 \rightarrow x_1) \wedge (y_2 \rightarrow x_2) \wedge (y_3 \rightarrow x_3) \wedge (y_4 \rightarrow x_4) = 1$$

В ответе **не нужно** перечислять все различные наборы значений переменных  $x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4$ , 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 считывается координата точки на прямой ( $x$  – действительное число) и определяется принадлежность этой точки одному из выделенных отрезков В и D (включая границы). Программист торопился и написал программу неправильно.

Бейсик	Паскаль
<pre>INPUT x IF x&gt;=-3 THEN   IF x&lt;=9 THEN     IF x&gt;1 THEN       PRINT "не принадлежит"     ELSE       PRINT "принадлежит"     ENDIF   ENDIF ENDIF END</pre>	<pre>var x: real; begin   readln(x);   if x&gt;=-3 then     if x&lt;=9 then       if x&gt;1 then         write('не принадлежит')       else         write('принадлежит')       end. end.</pre>
Си	Алгоритмический язык
<pre>void main(void) {   float x;   scanf("%f", &amp;x);   if(x&gt;=-3)     if(x&lt;=9)       if(x&gt;1)         printf("не принадлежит");       else         printf("принадлежит"); }</pre>	<pre>алг нач   вещ x   ввод x   если x&gt;=-3 то     если x&lt;=9 то       если x&gt;1 то         вывод 'не принадлежит'       иначе         вывод '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 и E). Границы (точки -3, 1, 5 и 9) принадлежат заштрихованным областям (B и D соответственно).

Область	Условие 1 ( $x \geq -3$ )	Условие 2 ( $x \leq 9$ )	Условие 3 ( $x > 1$ )	Программа выведет	Область обрабатывается верно
A					
B					
C					
D					
E					

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 её неправильной работы. (Это можно сделать несколькими способами, достаточно указать любой способ доработки исходной программы.)

**C2**

Дан целочисленный массив из 30 элементов. Элементы массива могут принимать целые значения от 0 до 100. Опишите на русском языке или на одном из языков программирования алгоритм, позволяющий найти и вывести произведение элементов массива, которые имеют нечётное значение и делятся на 3. Гарантируется, что в исходном массиве есть хотя бы один элемент, значение которого нечётно и кратно 3.

Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них. Исходные данные всегда подобраны так, что результат произведения не выходит за пределы объявленных типов данных.

Паскаль	Алгоритмический язык
<pre>const   N=30; var   a: array [1..N] of longint;   i, j, p: longint;</pre>	<pre>алг нач   цел N=30   целтаб a[1:N]   цел i, j, p</pre>

begin for i := 1 to N do readln(a[i]); ... end.	<u>НЦ</u> <u>ДЛЯ</u> <u>i</u> <u>ОТ</u> <u>1</u> <u>ДО</u> <u>N</u> <u>ВВОД</u> a[i] <u>КЦ</u> ... <u>КОН</u>
<b>Бейсик</b>	<b>СИ</b>
N=30 DIM A(N) AS LONG DIM I, J, P AS LONG FOR I = 1 TO N INPUT A(I) NEXT I ... END	#include <stdio.h> #define N 30 void main(void){ long a[N]; long i, j, p; for (i=0; i<N; i++) scanf("%ld", &a[i]); ... }
<b>Русский (естественный) язык</b>	
Объявляем массив A из 30 элементов. Объявляем целочисленные переменные I, J, P. В цикле от 1 до 30 вводим элементы массива A с 1-го по 30-й. ...	

В качестве ответа Вам необходимо привести фрагмент программы (или описание алгоритма на естественном языке), который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например Free Pascal 2.4) 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).

**С3** Два игрока, Петя и Ваня, играют в следующую игру. Перед ними лежат две кучки камней, в первой из которых – 4, а во второй – 3 камня. У каждого игрока неограниченно много камней. Игроки ходят по очереди, первый ход делает Петя. Ход состоит в том, что игрок или утраивает число камней в какой-либо куче, или добавляет 2 камня в какую-либо кучу. Игра завершается в тот момент, когда количество камней в одной из куч становится не менее 19. Если в момент завершения игры общее число камней в двух кучах не менее 35, то выиграл Ваня, в противном случае – Петя. Кто выигрывает при безошибочной игре обоих игроков? Каким должен быть первый ход выигрывающего игрока? Ответ обоснуйте.

**С4** На вход программе подаются сведения о пассажирах, желающих сдать свой багаж в камеру хранения на заранее известное время до полуночи. В первой строке сообщается число пассажиров N, которое не меньше 3, но не превосходит 1000; во второй строке – количество ячеек в камере хранения M, которое не меньше 10, но не превосходит 1000. Каждая из следующих N строк имеет следующий формат:  
<Фамилия> <время сдачи багажа> <время освобождения ячейки>,  
где <Фамилия> – строка, состоящая не более чем из 20 непробельных символов; <время сдачи багажа> – через двоеточие два целых числа, соответствующие часам (от 00 до 23 – ровно 2 символа) и минутам (от 00 до 59 – ровно 2 символа); <время освобождения ячейки> имеет тот же формат. <Фамилия> и <время сдачи багажа>, а также <время сдачи багажа> и <время освобождения ячейки> разделены одним пробелом. Время освобождения больше времени сдачи.  
Сведения отсортированы в порядке времени сдачи багажа. Каждому из пассажиров в камере хранения выделяется свободная ячейка с минимальным номером. Если в момент сдачи багажа свободных ячеек нет, то пассажир уходит, не дожидаясь освобождения одной из них.  
Требуется написать программу (укажите используемую версию языка программирования, например Borland Pascal 7.0), которая будет выводить на экран для каждого пассажира номер ему предоставленной ячейки (можно сразу после ввода данных очередного пассажира). Если ячейка пассажиру не предоставлена, то его фамилия не печатается.

Пример входных данных:

```
3
10
Иванов 09:45 12:00
Петров 10:00 11:00
Сидоров 12:00 13:12
```

Результат работы программы на этих входных данных:

```
Иванов 1
Петров 2
Сидоров 1
```

**Система оценивания экзаменационной работы по информатике и ИКТ**

**Часть 1**

За правильный ответ на каждое задание части 1 ставится 1 балл.

Если указаны два и более ответов (в том числе правильный), неверный ответ или ответ отсутствует – 0 баллов.

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

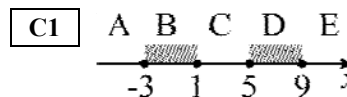
**Часть 2**

За правильный ответ на каждое задание части 2 ставится 1 балл, за неверный ответ или его отсутствие – 0 баллов.

№ задания	Ответ
B1	11212
B2	10
B3	5
B4	48
B5	9
B6	120
B7	15
B8	37
B9	13
B10	A122
B11	НСЕА
B12	2200
B13	22
B14	8
B15	15

**Часть 3**

**КРИТЕРИИ ОЦЕНИВАНИЯ ЗАДАНИЙ С РАЗВЕРНУТЫМ ОТВЕТОМ**



Требовалось написать программу, при выполнении которой с клавиатуры считывается координата точки на прямой ( $x$  – действительное число) и определяется принадлежность этой точки одному из выделенных отрезков В и D (включая границы). Программист торопился и написал программу неправильно.

Бейсик	Паскаль
<pre> INPUT x IF x&gt;=-3 THEN   IF x&lt;=9 THEN     IF x&gt;1 THEN       PRINT "не принадлежит"     ELSE       PRINT "принадлежит"     ENDIF   ENDIF ENDIF END                     </pre>	<pre> var x: real; begin   readln(x);   if x&gt;=-3 then     if x&lt;=9 then       if x&gt;1 then         write('не принадлежит')       else         write('принадлежит')       end     end   end.                     </pre>
Си	Алгоритмический язык
<pre> void main(void) {   float x;   scanf("%f",&amp;x);   if(x&gt;=-3)     if(x&lt;=9)       if(x&gt;1)         printf("не принадлежит");       else         printf("принадлежит"); }                     </pre>	<pre> алг нач   вещ x   ввод x   если x&gt;=-3 то     если x&lt;=9 то       если x&gt;1 то         вывод 'не принадлежит'       иначе         вывод 'принадлежит'       все     все   все конец                     </pre>

Последовательно выполните следующее.

1. Перерисуйте и заполните таблицу, которая показывает, как работает программа при аргументах, принадлежащих различным областям (A, B, C, D и E). Границы (точки -3, 1, 5 и 9) принадлежат заштрихованным областям (B и D соответственно).

Область	Условие 1 ( $x \geq -3$ )	Условие 2 ( $x \leq 9$ )	Условие 3 ( $x > 1$ )	Программа выведет	Область обрабатывается верно
A					
B					
C					
D					
E					

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 её 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 ( $x \geq -3$ )	Условие 2 ( $x \leq 9$ )	Условие 3 ( $x > 1$ )	Программа выведет	Область обрабатывается верно
A	Нет	—	—	—	Нет
B	Да	Да	Нет	Принадлежит	Да
C	Да	Да	Да	Не принадлежит	Да
D	Да	Да	Да	Не принадлежит	Нет
E	Да	Нет	—	—	Нет

2. Возможная доработка (Паскаль):

```
if (x>=-3) and (x<=1) or (x>=5) and (x<=9) then
  write('принадлежит')
else
  write('не принадлежит')
```

Возможны и другие способы доработки.

Например:

```
if x>=-3 then
  if x<=1 then
    write('принадлежит')
  else
    if x>=5 then
      if x<=9 then
        write('принадлежит')
      else
        write('не принадлежит')
    else
      write('не принадлежит')
else
  write('не принадлежит')
```

Другой пример:

```
if abs(abs(x-3)-4)<=2 then
  write('принадлежит')
else
  write('не принадлежит')
```

Указания по оцениванию	Баллы
<p>Обратите внимание! В задаче требуется выполнить <b>три</b> действия.</p> <ol style="list-style-type: none"> <li>Заполнить таблицу.</li> <li>Исправить ошибку в условном операторе.</li> <li>Исправить ошибку, связанную с неправильным набором условий.</li> </ol> <p>Баллы за данное задание начисляются как сумма баллов за верное выполнение каждого действия. Рассмотрим отдельно каждое действие.</p> <ol style="list-style-type: none"> <li>Действие по заполнению таблицы считается выполненным, если в таблице нет ошибок или ошибки присутствуют только в одной строке.</li> <li>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.</li> </ol> <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>, при этом программа не стала</p>	

<p>работать хуже, чем раньше, т.е. для всех точек, для которых программа ранее выдавала верный ответ, доработанная программа также должна выдавать верный ответ.</p> <p>3. Приведённых трёх ограничений недостаточно для описания двух областей (потеряно условие <math>x \geq 5</math>). Кроме того, необходимо учесть, что области не соединены. Исправлением этой ошибки может быть разбиение области на две части и использование дизъюнкции, либо использование сложной (для вывода) математической конструкции (<math>  x-3 -4  \leq 2</math>), либо использование сложной комбинации каскадных условий.</p> <p>В сложных случаях это действие считается выполненным, если верно определены заштрихованные области, т.е. программа выводит сообщение «принадлежит» для всех точек закрашенных областей, и только для них, для точек вне заштрихованных областей программа выводит «не принадлежит» или не выводит ничего.</p> <p>В работе (во фрагментах программ) допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора решения</p>	
Выполнены все три действия	3
<p>Правильно выполнены два действия из трёх (исправлены обе ошибки, но в пункте 1 задания не приведена таблица (либо таблица содержит ошибки в двух и более строках), либо приведена таблица (которая содержит ошибки не более чем в одной строке), но исправлена только одна ошибка программы).</p> <p>При написании операций сравнения допускается одно неправильное использование строгих/нестрогих неравенств (считается несущественной ошибкой, погрешностью записи). Например, вместо «<math>x \geq 5</math>» используется «<math>x &gt; 5</math>»</p>	2
<p>Правильно выполнено только одно действие из трёх, т.е. либо только приведена таблица, которая содержит ошибки в не более чем одной строке, либо таблица не приведена (или приведена и содержит ошибки более чем в одной строке), но исправлена одна ошибка программы. 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)</p>	1
<p>Все пункты задания выполнены неверно (таблица анализа правильности алгоритма не приведена либо содержит ошибки в двух и более строках, программа не приведена либо ни одна из двух ошибок не исправлена)</p>	0
<i>Максимальный балл</i>	3

**C2** Дан целочисленный массив из 30 элементов. Элементы массива могут принимать целые значения от 0 до 100. Опишите на русском языке или на одном из языков программирования алгоритм, позволяющий найти и вывести произведение элементов массива, которые имеют нечётное значение и делятся на 3. Гарантируется, что в исходном массиве есть хотя бы один элемент, значение которого нечётно и кратно 3.

Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них. Исходные данные всегда подобраны так, что результат произведения не выходит за пределы объявленных типов данных.

Паскаль	Алгоритмический язык
<pre>const     N=30; var     a: array [1..N] of longint;     i, j, p: longint; begin     for i := 1 to N do         readln(a[i]);     ... end.</pre>	<pre>алг нач     цел N=30     целтаб a[1:N]     цел i, j, p     нц для i от 1 до N         ввод a[i]     кц     ... кон</pre>
Бейсик	СИ
<pre>N=30 DIM A(N) AS LONG DIM I, J, P AS LONG FOR I = 1 TO N     INPUT A(I) NEXT I ... END</pre>	<pre>#include &lt;stdio.h&gt; #define N 30 void main(void){     long a[N];     long i, j, p;     for (i=0; i&lt;N; i++)         scanf("%ld", &amp;a[i]);     ... }</pre>
Русский (естественный) язык	
<p>Объявляем массив A из 30 элементов.                  Объявляем целочисленные переменные I, J, P.                  В цикле от 1 до 30 вводим элементы массива A с 1-го по 30-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 Free Pascal 2.4) 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>p := 1; for i := 1 to N do   if (a[i] mod 2 &lt;&gt; 0) and (a[i] mod 3 = 0) then     p := p * a[i]; writeln(p);</pre>
<b>На алгоритмическом языке</b>
<pre>p := 1 нц для i от 1 до N   если mod(a[i], 2) &lt;&gt; 0 и mod(a[i], 3) = 0     то       p := p * a[i]   все кц вывод p</pre>
<b>На языке Бейсик</b>
<pre>P = 1 FOR I = 1 TO N IF A(I) MOD 6 = 3 THEN P = P * A(I) ENDIF NEXT I PRINT P</pre>
<b>На языке СИ</b>
<pre>p=1; for (i=0; i&lt;N; i++)   if (a[i]%2!=0 &amp;&amp; a[i]%3==0)     p*=a[i]; printf("%ld", p);</pre>
<b>На естественном языке</b>
<p>Записываем в переменную P начальное значение, равное 1. В цикле от 1-го элемента до 30-го находим остаток от деления элемента исходного массива на 2 и на 3. Если остаток от деления на 2 не равен 0 и остаток от деления на 3 равен 0, то считаем произведение данного элемента массива и значения переменной P или достаточно проверить, что остаток от деления на 6 равен 3 (см. решение на языке Бейсик). Результат умножения сохраняем в переменную P. Переходим к следующему элементу массива. После завершения цикла выводим значение переменной P</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 ошибки из числа следующих. <ol style="list-style-type: none"> <li>1. Не инициализируется или неверно инициализируется переменная P (например, присваивается начальное значение, равное 0).</li> <li>2. Неверно осуществляется проверка на нечётность.</li> <li>3. На нечётность проверяется не значение элемента, а его индекс.</li> <li>4. Неверно осуществляется проверка делимости на 3.</li> <li>5. На делимость на 3 проверяется не значение элемента, а его индекс.</li> <li>6. В сложном условии вместо логической операции «И» используется логическая операция «ИЛИ».</li> <li>7. Неверно осуществляется накопление произведения в цикле (например, p := a[i]).</li> <li>8. Отсутствует вывод ответа.</li> <li>9. Используется переменная, не объявленная в разделе описания переменных.</li> <li>10. Не указано или неверно указано условие завершения цикла.</li> <li>11. Индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно.</li> <li>12. Неверно расставлены операторные скобки</li> </ol>	1
Ошибок, перечисленных в п. 1–12, две или больше, или алгоритм сформулирован неверно	0
<i>Максимальный балл</i>	2

**С3**

Два игрока, Петя и Ваня, играют в следующую игру. Перед ними лежат две кучки камней, в первой из которых 4, а во второй – 3 камня. У каждого игрока неограниченно много камней. Игроки ходят по очереди, первый ход делает Петя. Ход состоит в том, что игрок или утраивает число камней в какой-то куче, или добавляет 2 камня в какую-то кучу. Игра завершается в тот момент, когда количество камней в одной из куч становится не менее 19. Если в момент завершения игры общее число камней в двух кучах не менее 35, то выиграл Ваня, в противном случае – Петя. Кто выигрывает при безошибочной игре обоих игроков? Каким должен быть первый ход выигрывающего игрока? Ответ обоснуйте.

**Содержание верного ответа и указания к оцениванию**  
(допускаются иные формулировки ответа, не искажающие его смысла)

Выигрывает Петя, своим первым ходом он должен добавить 2 камня к первой куче. Для доказательства рассмотрим неполное дерево игры, оформленное в виде таблицы, где в каждой ячейке записаны пары чисел, разделённые запятой. Эти числа соответствуют количеству камней на каждом этапе игры в первой и второй кучах соответственно.

Позиция после первого хода	1-й ход	2-й ход	3-й ход	4-й ход
	Ваня (все варианты хода)	Петя (выигрышные ходы)	Ваня (все варианты хода, кроме непосредственно проигрышных)	Петя (выигрышные ходы, экзаменуемому достаточно указать один из вариантов)
<b>6.3</b>	6, 5	<b>6, 7</b>	18, 7	<b>20, 7</b>
			8, 7	<b>24, 7</b>
			6, 9	<b>6, 27</b>
	18, 3	<b>20, 3</b>	<b>Выигрыш Пети</b>	
	6, 9	<b>6, 27</b>		
	8, 3	<b>24, 3</b>		

Таблица содержит **все возможные** варианты ходов второго игрока. Из неё видно, что при любом ответе второго игрока у первого имеется ход, приводящий к победе.

Указания по оцениванию	Баллы
Правильное указание выигрывающего игрока и его ходов со строгим доказательством правильности (с помощью или без помощи дерева игры)	3
При наличии в представленном решении одного из пунктов. 1. Правильное указание выигрывающего игрока, стратегии игры, приводящей к победе, но при отсутствии доказательства её правильности. 2. Правильно указан выигрывающий игрок, приведено дерево игры, но отсутствует обоснование правильности выигрывающей стратегии. 3. Правильно указан выигрыш Пети, правильно указан первый ход, рассмотрены все варианты ответа Вани, для каждого из них правильно указан выигрывающий ответ Пети. Однако анализ игры не доведён до конца, и отсутствует обоснование стратегии	2
При наличии в представленном решении одного из пунктов. 1. Победитель указан неправильно, однако указаны все варианты первого хода Пети, и для каждого из них указан (выигрывающий, по мнению экзаменуемого) ответ Вани, но в анализе дальнейших действий были допущены ошибки. 2. Правильно указан выигрыш Пети, правильно указан первый ход, но описание выигрышной стратегии неполно, и для некоторых (больше одного, но не всех) вариантов ответа Вани правильно указан выигрывающий ответ Пети	1
Задание не выполнено, или в представленном решении полностью отсутствует описание элементов выигрышной стратегии и отсутствует анализ вариантов первого и второго ходов играющих (даже при наличии правильного указания выигрывающего игрока)	0
<i>Максимальный балл</i>	3

**С4**

На вход программе подаются сведения о пассажирах, желающих сдать свой багаж в камеру хранения на заранее известное время до полуночи. В первой строке сообщается количество пассажиров N, которое не меньше 3, но не превосходит 1000; во второй строке – количество ячеек в камере хранения M, которое не меньше 10, но не превосходит 1000. Каждая из следующих N строк имеет следующий формат:

<Фамилия> <время сдачи багажа> <время освобождения ячейки>,  
где <Фамилия> – строка, состоящая не более чем из 20 непробельных символов; <время сдачи багажа> – через двоеточие два целых числа, соответствующие часам (от 00 до 23 – ровно 2 символа) и минутам (от 00 до 59 – ровно 2 символа); <время освобождения ячейки> имеет тот же формат. <Фамилия> и <время сдачи багажа>, а также <время сдачи багажа> и <время освобождения ячейки> разделены одним пробелом. Время освобождения больше времени сдачи.



Сведения отсортированы в порядке времени сдачи багажа. Каждому из пассажиров в камере хранения выделяется свободная ячейка с минимальным номером. Если в момент сдачи багажа свободных ячеек нет, то пассажир уходит, не дожидаясь освобождения одной из них.

Требуется написать программу (укажите используемую версию языка программирования, например Borland Pascal 7.0), которая будет выводить на экран для каждого пассажира номер ячейки, которая ему будет предоставлена (можно сразу после ввода данных очередного пассажира). Если ячейка пассажиру не предоставлена, то его фамилия не печатается.

Пример входных данных:

```
3
10
Иванов 09:45 12:00
Петров 10:00 11:00
Сидоров 12:00 13:12
```

Результат работы программы на этих входных данных:

```
Иванов 1
Петров 2
Сидоров 1
```

#### Содержание верного ответа и указания по оцениванию

(допускаются иные формулировки ответа, не искажающие его смысла)

Программа верно читает входные данные, сразу запоминая только время окончания хранения багажа в массиве, соответствующем ячейкам камеры хранения. Подходящая ячейка определяется путём последовательного просмотра элементов этого массива до первого свободного или такого, в котором записано время окончания хранения, не превосходящее текущего времени сдачи очередного багажа. В случае удачного выбора ячейки фамилия и номер ячейки распечатываются. Баллы начисляются только за программу, которая решает задачу хотя бы для частного случая. Время можно как переводить в минуты, так и хранить в виде строки, сравнивая затем строки непосредственно. В последнем случае упрощается ввод данных

Пример правильной программы на языке Паскаль:

```
var p:array[1..1000] of integer;
    c,c1:char;
    i,j,N,K:integer;
    name:string;
    time1,time2:integer;
begin
  readln(N,K);
  for i:=1 to K do
    p[i]:=0;
  for i:=1 to N do
    begin
      name:='';
      repeat
        read(c);
        name:=name+c
      until c=' '; {читана фамилия}
      read(c,c1); {читаны часы первого времени}
      time1:=60*((ord(c)-ord('0'))*10+ ord(c1)-ord('0'));
      read(c,c,c1); {пропущено двоеточие, и считаны минуты}
      time1:=time1+(ord(c)-ord('0'))*10+ord(c1)-ord('0');
      read(c,c,c1); {читаны часы второго времени}
      time2:=60*((ord(c)-ord('0'))*10+ ord(c1)-ord('0'));
      readln(c,c,c1); {пропущено двоеточие, и считаны минуты}
      time2:=time2+(ord(c)-ord('0'))*10+ord(c1)-ord('0');
      for j:=1 to K do
        if p[j]<=time1 then
          begin
            p[j]:=time2;
            writeln(name,' ',j);
            break;
          end;
    end;
end.
```

**Пример правильной программы на языке Бейсик:**

```

DIM p(1000) AS INTEGER
DIM s AS STRING
DIM nm AS STRING
INPUT n
INPUT k
FOR i = 1 TO k
  p(i) = 0
NEXT i
FOR j = 1 TO n
LINE INPUT s
c$ = MID$(s, 1, 1)
i = 1
WHILE NOT (c$ = " ")
  i = i + 1
  c$ = MID$(s, i, 1)
WEND
nm = MID$(s, 1, i)
time1 = (ASC(MID$(s, i + 1, 1)) - ASC("0")) * 60 * 10
time1 = time1 + (ASC(MID$(s, i + 2, 1)) - ASC("0")) * 60
time1 = time1 + (ASC(MID$(s, i + 4, 1)) - ASC("0")) * 10
time1 = time1 + (ASC(MID$(s, i + 5, 1)) - ASC("0"))
time2 = (ASC(MID$(s, i + 7, 1)) - ASC("0")) * 60 * 10
time2 = time2 + (ASC(MID$(s, i + 8, 1)) - ASC("0")) * 60
time2 = time2 + (ASC(MID$(s, i + 10, 1)) - ASC("0")) * 10
time2 = time2 + (ASC(MID$(s, i + 11, 1)) - ASC("0"))
FOR i = 1 TO k
  IF time1 >= p(i) THEN
    p(i) = time2
    PRINT nm, i
    GOTO 10
  ENDIF
NEXT i
10 NEXT j
END

```

**Указания по оцениванию****Баллы**

Программа работает верно, т.е. корректно выделяет из входных данных время, ищет первую свободную ячейку и распечатывает в случае успеха результат. Фамилии пассажиров и время сдачи багажа при этом не запоминаются. 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, не допустимая для соответствующего типа данных. Допускается наличие одной из нерациональностей: сохраняются фамилии пассажиров; сохраняется как время сдачи багажа, так и время освобождения ячейки; до

4

анализа данных очередного пассажира анализируется массив, соответствующий ячейкам камеры хранения на предмет освобождения ячеек к данному времени	
Программа работает в целом верно, но содержит, по крайней мере, две из указанных выше нерациональностей; допускается наличие до трёх синтаксических ошибок, описанных выше	3
Программа работает в целом верно, но не всегда определяет для хранения допустимую ячейку с минимальным номером или некорректно работает в случае отсутствия свободных ячеек. Возможно, в реализации алгоритма содержится одна-две ошибки (используется знак «<» вместо «>», «ог» вместо «and» и т.п.). Возможно, некорректно организовано считывание входных данных. Допускается до трёх ошибок в ходе решения задачи. Допускается наличие до пяти синтаксических ошибок, описанных выше	2
Программа неверно работает при некоторых входных данных и, возможно, содержит ошибку в определении свободной ячейки. Допускается до четырёх различных ошибок в ходе решения задачи, в том числе описанных в критериях. Допускается наличие до семи синтаксических ошибок, описанных выше	1
Задание выполнено неверно	0