

Диагностическая работа № 1
по ИНФОРМАТИКЕ
14 декабря 2012 года
11 класс

Вариант 1

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 развёрнутый 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 удается 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.

Район.	_____
Город (населённый пункт)	_____
Школа.	_____
Класс.	_____
Фамилия.	_____
Имя	_____
Отчество.	_____

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, $A \vee B$) либо $|$ (например, $A | B$);

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 А и В 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, содержащие 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, $\neg A \wedge B \vee C \wedge D$ совпадает с $((\neg A) \wedge B) \vee (C \wedge D)$. 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. Везде в задачах 1 Кбайт = 1024 байт; 1 Мбайт = 1024 Кбайт.

Часть 1

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

A1 Для каждого из перечисленных ниже десятичных чисел построили двоичную запись. Укажите число, двоичная запись которого содержит наибольшее количество единиц.

- 1) 13 2) 14 3) 15 4) 16

A2 Между населёнными пунктами A, B, C, D, E, F, Z построены дороги, протяжённость которых приведена в таблице. (Отсутствие числа в таблице означает, что прямой дороги между пунктами нет.)

	A	B	C	D	E	F	Z
A		4	6				43
B	4		1				
C	6	1		15			32
D			15		4	6	10
E				4			8
F				6			2
Z	43		32	10	8	2	

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

- 1) 28 2) 38 3) 41 4) 43

A3 Дан фрагмент таблицы истинности выражения F:

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

Каким выражением может быть F?

- 1) $x_1 \wedge \neg x_2 \wedge x_3 \wedge \neg x_4 \wedge x_5 \wedge \neg x_6 \wedge x_7$
 2) $x_1 \vee \neg x_2 \vee x_3 \vee \neg x_4 \vee x_5 \vee \neg x_6 \vee x_7$
 3) $\neg x_1 \vee x_2 \vee \neg x_3 \vee x_4 \vee \neg x_5 \vee x_6 \vee \neg x_7$
 4) $\neg x_1 \wedge x_2 \wedge \neg x_3 \wedge x_4 \wedge \neg x_5 \wedge x_6 \wedge \neg x_7$

A4 Для групповых операций с файлами используются **маски имён файлов**. Мaska 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:

astarta.doc
catarsis.dat
catarsis.doc
plataria.docx
start.doc
tartar.docx

Определите, по какой из масок из каталога будет отобрана указанная группа файлов:

astarta.doc
catarsis.doc
plataria.docx
start.doc

- 1) *tar?*.d*
2) ?tar*.doc

- 3) *?tar?*.do*
4) *tar?.doc*

A5 Автомат получает на вход четырёхзначное восьмеричное число. По этому числу строится новое число по следующим правилам.

1. Складываются первая и вторая, а также третья и четвёртая цифры.
2. Полученные два числа записываются друг за другом в порядке возрастания (без разделителей).

Пример. Исходное число: 4531. Суммы: $4+5=11$; $3+1=4$. Результат: 411.

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

- 1) 117 2) 1213 3) 1511 4) 1517

A6 В фрагменте базы данных представлены сведения о родственных отношениях. На основании приведённых данных определите фамилию и инициалы бабушки Гейко П.А.

Таблица 1

ID	Фамилия_И.О.	Пол
940	Карпец Д.К.	Ж
941	Базилевич Б.Ф.	М
953	Лемешко В.А.	Ж
959	Месхи К.Г.	М
974	Гейко Л.А.	Ж
978	Гейко А.И.	Ж
985	Базилевич А.Б.	Ж
1006	Фирс П.А.	Ж
1012	Базилевич И.Б.	М
1023	Фоменко Г.Р.	Ж
1044	Гейко А.П.	М
1069	Базилевич Т.И.	Ж
1091	Базилевич П.И.	М
1100	Мухина Р.Г.	Ж
1115	Звавич М.А	Ж
1130	Гейко П.А.	М
...

Таблица 2

ID_Родителя	ID_Ребёнка
940	985
940	1012
941	985
941	1012
985	953
985	1115
985	1130
1012	1069
1012	1091
1023	1069
1023	1091
1044	953
1044	1115
1044	1130
1100	974
1100	1006
....	...

- 1) Базилевич Б.Ф. 2) Гейко А.И. 3) Гейко Л.А. 4) Карпец Д.К.

A7 В ячейки диапазона C3:F6 электронной таблицы записаны числа, как показано на рисунке.

	A	B	C	D	E	F
1						
2						
3		1	2	3	4	
4		11	13	15	17	
5		21	24	27	30	
6		31	35	39	43	

В ячейке B2 записали формулу =E\$5-\$D4. После этого ячейку B2 скопировали в ячейку A1. Какое число будет показано в ячейке A1?

Примечание: знак \$ используется для обозначения абсолютной адресации.

- 1) 22 2) 14 3) 12 4) 4

A8 Производилась двухканальная (стерео) звукозапись с частотой дискретизации 32 кГц и 24-битным разрешением. В результате был получен файл размером 60 Мбайт, сжатие данных не производилось. Какая из приведённых ниже величин наиболее близка к времени, в течение которого проводилась запись?

- 1) 2 мин 2) 5 мин 3) 10 мин 4) 15 мин

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. Для букв А, Б, В и Г использовали такие кодовые слова: А – 111, Б – 110, В – 101, Г – 100.

Укажите, каким кодовым словом из перечисленных ниже может быть закодирована буква Д.

Код должен удовлетворять свойству однозначного декодирования. Если можно использовать более одного кодового слова, укажите кратчайшее из них.

- 1) 1 2) 0 3) 01 4) 10

A10 На числовой прямой даны два отрезка: $P = [3, 9]$ и $Q = [4, 12]$.

Выберите такой отрезок A, что формула

$$((x \in A) \rightarrow (x \in P)) \vee (x \in Q)$$

тождественно истинна, то есть принимает значение 1 при любом значении переменной x.

- 1) [0, 5] 2) [5, 10] 3) [10, 15] 4) [15, 20]

A11 При регистрации в компьютерной системе каждому пользователю выдаётся пароль, состоящий из 15 символов и содержащий только символы К, О, М, П, Б, Ю, Т, Е, Р. Каждый такой пароль 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 30 паролей.

- 1) 180 байт 2) 210 байт 3) 240 байт 4) 270 байт

A12 В программе описан одномерный целочисленный массив с индексами от 0 до 10. Ниже представлен записанный на разных языках программирования фрагмент одной и той же программы, обрабатывающей данный массив.

Бейсик	<pre>s = 0 n = 10 FOR i = 0 TO n IF i=n-i THEN s = s+A(i)+A(i+1) END IF NEXT i</pre>
--------	--

Паскаль	<pre>s:=0; n:=10; for i:=0 to n do begin if i = n-i then s:=s+A[i]+A[i+1]; end;</pre>
---------	---

Си	<pre>s = 0; n=10; for (i = 0; i <= n; i++) if (i==n-i) s=s+A[i]+A[i+1];</pre>
----	--

Алгоритмический язык	<pre>s:=0 n:=10 нц для i от 0 до n если i = n-i то s:=s+A[i]+A[i+1] все кц</pre>
----------------------	--

В начале выполнения этого фрагмента в массиве находились числа 0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, т. е. $A[0]=0$, $A[1]=10$ и т. д. Чему будет равно значение переменной s после выполнения данной программы?

- 1) 0 2) 100 3) 110 4) 550

A13 Система команд исполнителя РОБОТ, «живущего» в прямоугольном лабиринте на клетчатой плоскости, состоит из 8 команд. Четыре команды – это команды-приказы:

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

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

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

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

Цикл
ПОКА условие
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)

В конструкциях ПОКА и ЕСЛИ условие может содержать команды проверки, а также слова И, ИЛИ, НЕ, обозначающие логические операции.

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

Сколько клеток лабиринта соответствуют требованию, что, начав движение в данной клетке и выполнив предложенную программу, РОБОТ уцелеет и остановится в закрашенной клетке (клетка F6)?

НАЧАЛО
ПОКА снизу свободно ИЛИ справа свободно
ПОКА снизу свободно

вниз

КОНЕЦ ПОКА

ЕСЛИ справа свободно

ТО

вправо

КОНЕЦ ЕСЛИ

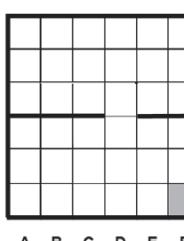
КОНЕЦ ПОКА

КОНЕЦ

1) 18

2) 24

3) 27



4) 30

Часть 2

Ответом к заданиям этой части (B1–B15) является число, последовательность букв или цифр. Впишите ответы сначала в текст работы, а затем перенесите их в бланк ответов № 1 справа от номера соответствующего задания, начиная с первой клеточки, без пробелов, запятых и других дополнительных символов. Каждую букву или цифру пишите в отдельной клеточке в соответствии с приведёнными в бланке образцами.

B1 У исполнителя Квадр две команды, которым присвоены номера:

1. прибавь 1,
2. возвели в квадрат.

Первая из этих команд увеличивает число на экране на 1, вторая – возводит в квадрат. Программа для исполнителя Квадр – это последовательность номеров команд.

Например, 21121 – это программа

возвели в квадрат

прибавь 1

прибавь 1

возвели в квадрат

прибавь 1

Эта программа преобразует число 3 в число 122.

Запишите программу, которая преобразует **число 1 в число 101** и содержит не более 6 команд. Если таких программ более одной, то запишите любую из них.

Ответ:

B2 Определите значение переменной **c** после выполнения следующего фрагмента программы (записанного ниже на разных языках программирования). Ответ запишите в виде целого числа.

Бейсик	<pre>a = 30 b = 6 a = a * 3 / b IF a < b THEN c = 2*a - 10 * b ELSE c = 2*a + 10 * b END IF</pre>
--------	--

Паскаль	<pre>a := 30; b := 6; a := a * 3 / b; if a < b then c := 2*a - 10 * b else c := 2*a + 10 * b;</pre>
---------	--

Си	<pre>a = 30; b = 6; a = a * 3 / b; if (a < b) c = 2*a - 10 * b; else c = 2*a + 10 * b;</pre>
----	---

Алгоритмический	<pre>a := 30 b := 6 a := a * 3 / b если a < b то с := 2*a - 10 * b иначе с := 2*a + 10 * b все</pre>
-----------------	---

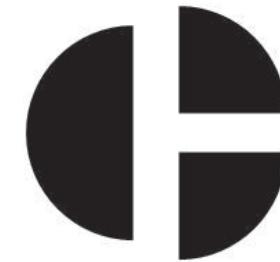
Ответ:

B3 Дан фрагмент электронной таблицы.

	A	B	C
1	2		=A1*2
2	$=(B1-A1)/2$	$=B1-C1$	$=B2+A1$

Какое число должно быть записано в ячейке B1, 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 более пяти** сигналов (точек и тире)?

Ответ:

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

Бейсик

```
DIM N, S AS INTEGER
N = 0
S = 0
WHILE S <= 365
    S = S + 33
    N = N + 5
WEND
PRINT N
```

Паскаль

```
var n, s: integer;
begin
    n := 0;
    s := 0;
    while s <= 365 do
    begin
        s := s + 33;
        n := n + 5
    end;
    write(n)
end.
```

Си

```
#include<stdio.h>
void main()
{
    int n, s;
    n = 0;
    s = 0;
    while (s <= 365)
    {
        s = s + 33;
        n = n + 5;
    }
    printf("%d", n);
}
```

Алгоритмический

```
алг
нач
    цел n, s
    n := 0
    s := 0
    цц пока s <= 365
        s := s + 33
        n := n + 5
    кц
    вывод n
кон
```

Ответ:

B6 Алгоритм вычисления значения функции $F(n)$, где n – натуральное число, задан следующими соотношениями:

$$\begin{aligned} F(1) &= 1; \quad F(2)=1; \\ F(n) &= F(n-2) * n \text{ при } n > 2. \end{aligned}$$

Чему равно значение функции $F(7)$? В ответе запишите только натуральное число.

Ответ:

B7 Десятичное число 57 в некоторой системе счисления записывается как 212. Определите основание системы счисления.

Ответ:

B8 Ниже на 4 языках программирования 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 3, а потом 7.

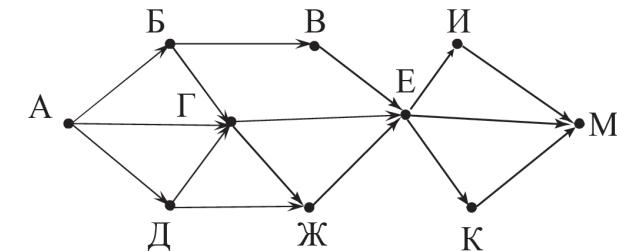
Бейсик

```
DIM X, A, B AS INTEGER
INPUT X
A=0: B=0
WHILE X > 0
    A = A+1
    B = B +(X MOD 10)
    X = X \ 10
WEND
PRINT A
PRINT B
```

Паскаль <pre>var x, a, b: integer; begin readln(x); a:=0; b:=0; while x>0 do begin a:=a + 1; b:=b + (x mod 10); x:=x div 10; end; writeln(a); write(b); end.</pre>
Си <pre>#include<stdio.h> void main() { int x, a, b; scanf("%d", &x); a=0; b=0; while (x>0) { a = a+1; b = b +(x%10); x = x/10; } printf("%d\n%d", a, b); }</pre>
Алгоритмический <pre>алг нач цел x, a, b ввод x a:=0; b:=0 нц пока x>0 a:=a+1 b:=b+mod(x,10) x:=div(x,10) кц вывод a, b кон</pre>

Ответ:

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



Ответ:

- B10** Документ объёмом 15 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% от исходного,
 – время, требуемое на сжатие документа, – 18 секунд, на распаковку – 2 секунды?

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

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

Слова «секунд», «сек.», «с.» к ответу добавлять не нужно.

Ответ:

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

IP-адрес узла: 224.23.251.133

Маска: 255.255.240.0

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

A	B	C	D	E	F	G	H
255	240	252	224	133	23	8	0

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:

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)
Лебедь & (Рак Щука)	320
Лебедь & Рак	200
Лебедь & Рак & Щука	50

Какое количество страниц (в тысячах) будет найдено по запросу
Лебедь & Щука

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 2,
2. умножь на 2.

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

Программа для Удвоителя – это последовательность команд.

Сколько есть программ, которые число 2 преобразуют в число 40?

Ответ:

B14 Определите, какое число будет напечатано в результате выполнения следующего алгоритма (для Вашего удобства алгоритм представлен на четырёх языках):

Бейсик	<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) < R THEN M = T R = F(T) END IF NEXT T PRINT R FUNCTION F(x) F = 16*(9-x)*(9-x)+127 END FUNCTION </pre>
--------	---

Паскаль	<pre> var a,b,t,M,R :integer; Function F(x: integer):integer; begin F := 16*(9-x)*(9-x)+127; end; BEGIN a := -20; b := 20; M := a; R := F(a); for t := a to b do begin if (F(t) < R) then begin M := t; R := F(t); end; end; write(R); END. </pre>
---------	---

Си

```
#include<stdio.h>
int F(int x)
{
    return 16*(9-x)*(9-x)+127;
}
void main()
{
    int a, b, t, M, R;
    a = -20; b = 20;
    M = a; R = F(a);
    for (t=a; t<=b; t++) {
        if (F(t) < R) {
            M = t; R = F(t);
        }
    }
    printf("%d", R);
}
```

Алгоритмический

```
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) < R
            то
                M := t; R := F(t)
        все
    вывод R
кон
алг цел F(цел x)
нач
    знач := 16*(9-x)*(9-x)+127
кон
```

Ответ:

B15 Сколько существует различных наборов значений логических переменных $x_1, x_2, x_3, x_4, x_5, y_1, y_2, y_3, y_4, y_5$, которые удовлетворяют всем перечисленным ниже условиям?

$$(x_1 \rightarrow x_2) \wedge (x_2 \rightarrow x_3) \wedge (x_3 \rightarrow x_4) \wedge (x_4 \rightarrow x_5) = 1$$

$$(y_1 \rightarrow y_2) \wedge (y_2 \rightarrow y_3) \wedge (y_3 \rightarrow y_4) \wedge (y_4 \rightarrow y_5) = 1$$

$$x_1 \rightarrow y_1 = 1$$

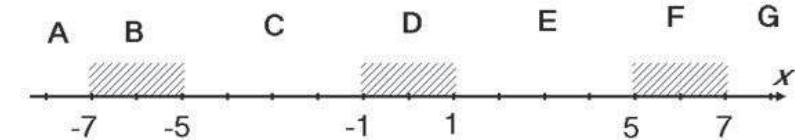
В ответе **не нужно** перечислять все различные наборы значений переменных $x_1, x_2, x_3, x_4, x_5, y_1, y_2, y_3, y_4, y_5$, при которых выполнена данная система равенств. В качестве ответа Вам нужно указать количество таких наборов.

Ответ:

Часть 3

Для записи ответов к заданиям этой части (C1–C4) используйте бланк ответов № 2. Запишите сначала номер задания (C1 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 (включая границы). Программист торопился и написал программу неправильно.



Бейсик	<pre>INPUT x IF x>=-7 OR x<=1 THEN IF x>=-5 AND x<=5 THEN IF x>=-1 AND x<=7 THEN PRINT "принадлежит" ELSE PRINT "не принадлежит" END IF END IF END IF</pre>
---------------	---

Паскаль

```

var x: real;
begin
    readln(x);
    if (x>=-7) or (x<=1) then
        if (x>=-5) and (x<=5) then
            if (x>=-1) and (x<=7) then
                write('принадлежит')
            else
                write('не принадлежит')
        end.
    
```

Си

```

#include <stdio.h>
void main(){
    float x;
    scanf("%f", &x);
    if (x>=-7 || x<=1)
        if (x>=-5 && x<=5)
            if (x>=-1 && x<=7)
                printf("принадлежит");
            else
                printf("не принадлежит");
    }
}

```

Алгоритмический язык

```

алг
нач
    веш x
    ввод x
    если x>=-7 или x<=1 то
        если x>=-5 и x<=5 то
            если x>=-1 и x<=7 то
                вывод 'принадлежит'
            иначе
                вывод 'не принадлежит'
            все
        все
    кон

```

Последовательно выполните следующее.

- Пересвойте и заполните таблицу, которая показывает, как работает программа при аргументе, принадлежащем различным областям (A, B, C, D, E, F, G). Точки, лежащие на границах областей, отдельно не рассматривать.

Область	Условие 1 ($x \geq -7$ или $x \leq 1$)	Условие 2 ($x \geq -5$ и $x \leq 5$)	Условие 3 ($x \geq -1$ и $x \leq 7$)	Программа выведет	Область обрабатывается верно
A					
B					
C					
D					
E					
F					
G					

В столбцах условий укажите «да», если условие выполнится, «нет», если условие не выполнится, «—» (прочерк), если условие не будет проверяться, «не изв.», если программа ведёт 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 массив, содержащий 70 целых чисел. Опишите на одном из языков программирования алгоритм, позволяющий найти и вывести наименьшее содержащееся в массиве положительное число, десятичная запись которого оканчивается цифрой 7. Гарантируется, что в массиве есть хотя бы один положительный элемент, десятичная запись которого оканчивается цифрой 7. Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

Паскаль

```

const
    N=70;
var
    a: array [1..N] of integer;
    i, j, m: integer;
begin
    for i:=1 to N do
        readln(a[i]);
    ...
end.

```

Бейсик

```
N=70
DIM A(N) AS INTEGER
DIM I, J, M AS INTEGER
FOR I = 1 TO N
    INPUT A(I)
NEXT I
...
END
```

Си

```
#include <stdio.h>
#define N 70
void main(){
    int a[N];
    int i, j, m;
    for (i=0; i<N; i++)
        scanf("%d", &a[i]);
    ...
}
```

Алгоритмический язык

```
алг
нач
    цел N=70
    целтаб a[1:N]
    цел i, j, m
    нц для i от 1 до N
        ввод a[i]
    кц
    ...
кон
```

В качестве ответа Вам необходимо привести фрагмент программы, 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.

C3 Два игрока, Паша и Вова, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Паша. За один ход игрок может добавить в кучу 1 камень или 10 камней. Например, имея кучу из 7 камней, за один ход можно получить кучу из 8 или 17 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней. Игра завершается в тот момент, когда количество камней в куче становится не менее 31. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 31 или больше камней.

В начальный момент в куче было S камней, $1 \leq S \leq 30$.

Будем говорить, что игрок имеет *выигрышную стратегию*, если он может выиграть при любых ходах противника. Описать стратегию игрока – значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника.

Выполните следующие задания. Во всех случаях обосновывайте свой ответ.

1. а) Укажите все такие значения числа S , при которых Паша может выиграть в один ход. Обоснуйте, что найдены все нужные значения S , и укажите выигрышающие ходы.
- б) Укажите такое значение S , при котором Паша не может выиграть за один ход, но при любом ходе Паша Вова может выиграть своим первым ходом. Опишите выигрышную стратегию Вовы.
2. Укажите два значения S , при которых у Паши есть выигрышная стратегия, причём Паши не может выиграть за один ход, но может выиграть своим вторым ходом независимо от того, как будет ходить Вова. Для указанных значений S опишите выигрышную стратегию Паши.
3. Укажите значение S , при котором у Вовы есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Паши, однако у Вовы нет стратегии, которая позволит ему гарантированно выиграть первым ходом. Для указанного значения S опишите выигрышную стратегию Вовы. Постройте дерево всех партий, возможных при этой выигрышной стратегии Вовы (в виде рисунка или таблицы). На ребрах дерева указывайте, кто делает ход, в узлах – количество камней в куче.

C4 На плоскости дан набор точек с целочисленными координатами. Необходимо найти такой треугольник наибольшей площади с вершинами в этих точках, у которого нет общих точек с осью Oy , а одна из сторон лежит на оси Ox .

Напишите эффективную, в том числе по памяти, программу, которая будет решать эту задачу. Размер памяти, которую использует Ваша программа, не должен зависеть от количества точек.

Перед текстом программы кратко опишите используемый алгоритм решения задачи и укажите используемый язык программирования и его версию.

Описание входных данных

В первой строке вводится одно целое положительное число – количество точек N .

Каждая из следующих N строк содержит два целых числа – сначала координата x , затем координата y очередной точки. Числа разделены пробелом.

Описание выходных данных

Программа должна вывести одно число – максимальную площадь треугольника, удовлетворяющего условиям задачи. Если такого треугольника не существует, программа должна вывести ноль.

Пример входных данных:

```
8
-10 0
2 0
0 4
3 3
7 0
5 5
4 0
9 -9
```

Пример выходных данных для приведённого выше примера входных данных:

22.5

Диагностическая работа № 1
по ИНФОРМАТИКЕ
14 декабря 2012 года
11 класс

Вариант 2

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 развёрнутый 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 удается 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.

Район.	_____
Город (населённый пункт)	_____
Школа.	_____
Класс.	_____
Фамилия.	_____
Имя	_____
Отчество.	_____

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, $A \vee B$) либо $|$ (например, $A | B$);
- 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 А и В 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, содержащие 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, $\neg A \wedge B \vee C \wedge D$ совпадает с $((\neg A) \wedge B) \vee (C \wedge D)$. 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. Везде в задачах 1 Кбайт = 1024 байт; 1 Мбайт = 1024 Кбайт.

Часть 1

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

A1 Для каждого из перечисленных ниже десятичных чисел построили двоичную запись. Укажите число, двоичная запись которого содержит наибольшее количество значащих нулей.

- 1) 1 2) 7 3) 11 4) 15

A2 Между населёнными пунктами A, B, C, D, E, F, Z построены дороги, протяжённость которых приведена в таблице. (Отсутствие числа в таблице означает, что прямой дороги между пунктами нет.)

	A	B	C	D	E	F	Z
A		4	6				33
B	4		1				
C	6	1		5			27
D			5		4	8	10
E				4		1	8
F				8	1		2
Z	33		27	10	8	2	

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

- 1) 13 2) 17 3) 21 4) 33

A3 Дан фрагмент таблицы истинности выражения F:

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

Каким выражением может быть F?

- 1) $x_1 \wedge x_2 \wedge \neg x_3 \wedge x_4 \wedge x_5 \wedge \neg x_6 \wedge x_7$
 2) $x_1 \vee x_2 \vee \neg x_3 \vee x_4 \vee x_5 \vee \neg x_6 \vee x_7$
 3) $\neg x_1 \vee \neg x_2 \vee x_3 \vee \neg x_4 \vee \neg x_5 \vee x_6 \vee \neg x_7$
 4) $\neg x_1 \wedge \neg x_2 \wedge x_3 \wedge \neg x_4 \wedge \neg x_5 \wedge x_6 \wedge \neg x_7$

A4 Для групповых операций с файлами используются **маски имён файлов**. Мaska 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:

korsten.docx

mikor5.docx

mokkorte.dat

mokkorte.doc

skorcher.doc

x-korvet.doc

Определите, по какой из масок из каталога будет отобрана указанная группа файлов:

mikor5.docx

mokkorte.doc

skorcher.doc

x-korvet.doc

1) *kor?*.d*

3) *?kor?*.do*

2) ?kor*.doc

4) *kor?.doc*

A5 Автомат получает на вход четырёхзначное восьмеричное число. По этому числу строится новое число по следующим правилам.

1. Складываются первая и вторая, а также третья и четвёртая цифры.

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

Пример. Исходное число: 4531. Суммы: $4+5 = 11$; $3+1 = 4$. Результат: 411.

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

1) 811

2) 717

3) 1511

4) 1214

A6 В фрагменте базы данных представлены сведения о родственных отношениях. На основании приведённых данных определите фамилию и инициалы дедушки Корзуна П.А.

Таблица 1

ID	Фамилия_И.О.	Пол
1072	Онищенко А.Б.	Ж
1028	Онищенко Б.Ф.	М
1099	Онищенко И.Б.	М
1178	Онищенко П.И.	М
1156	Онищенко Т.И.	Ж
1065	Корзун А.И.	Ж
1131	Корзун А.П.	М
1061	Корзун Л.Г.	М
1217	Корзун П.А.	М
1202	Зельдович М.А.	Ж
1027	Витюк Д.К.	Ж
1040	Лемешко В.А.	Ж
1046	Месяц К.Г.	М
1187	Лукина Р.Г.	Ж
1093	Фокс П.А.	Ж
1110	Друк Г.Р.	Ж
...

Таблица 2

ID_Родителя	ID_Ребёнка
1027	1072
1027	1099
1028	1072
1028	1099
1072	1040
1072	1202
1072	1217
1099	1156
1099	1178
1110	1156
1110	1178
1131	1040
1131	1202
1131	1217
1187	1061
1187	1093
...	...

1) Витюк Д.К.

3) Корзун Л.Г.

2) Корзун А.П.

4) Онищенко Б.Ф.

- A7** В ячейки диапазона C3:F6 электронной таблицы записаны числа, как показано на рисунке.

	A	B	C	D	E	F
1						
2						
3		1	2	3	4	
4		11	22	33	44	
5		22	11	44	33	
6		31	35	39	43	

В ячейке B2 записали формулу =E\$5+\$D4. После этого ячейку B2 скопировали в ячейку A1. Какое число будет показано в ячейке A1?

Примечание: знак \$ используется для обозначения абсолютной адресации.

- 1) 13 2) 23 3) 44 4) 66

- A8** Производилась двухканальная (стерео) звукозапись с частотой дискретизации 16 кГц и 24-битным разрешением. В результате был получен файл размером 60 Мбайт, сжатие данных не производилось. Какая из приведённых ниже величин наиболее близка к времени, в течение которого проводилась запись?

- 1) 2 мин 2) 5 мин 3) 10 мин 4) 15 мин

- 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. Для букв А, Б, В и Г использовали такие кодовые слова: А – 000, Б – 001, В – 010, Г – 011.

Укажите, каким кодовым словом из перечисленных ниже может быть закодирована буква Д.

Код должен удовлетворять свойству однозначного декодирования. Если можно использовать более одного кодового слова, укажите кратчайшее из них.

- 1) 1 2) 0 3) 01 4) 10

- A10** На числовой прямой даны два отрезка: $P = [4, 16]$ и $Q = [9, 18]$.

Выберите такой отрезок A, что формула

$$((x \in A) \rightarrow (x \in P)) \vee (x \in Q)$$

тождественно истинна, то есть принимает значение 1 при любом значении переменной x.

- 1) [1, 11] 2) [5, 15] 3) [11, 21] 4) [15, 25]

- A11** При регистрации в компьютерной системе каждому пользователю выдаётся пароль, состоящий из 15 символов и содержащий только символы К, О, М, П, Б, Ю, Т, Е, Р. Каждый такой пароль 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 25 паролей.

- 1) 150 байт 2) 200 байт 3) 225 байт 4) 250 байт

- A12** В программе описан одномерный целочисленный массив с индексами от 0 до 10. Ниже представлен записанный на разных языках программирования фрагмент одной и той же программы, обрабатывающей данный массив.

Бейсик	<pre>s = 0 n = 10 FOR i = 0 TO n IF i=n-i THEN s = s+A(i)+A(i+1) END IF NEXT i</pre>
--------	--

Паскаль	<pre>s:=0; n:=10; for i:=0 to n do begin if i = n-i then s:=s+A[i]+A[i+1]; end;</pre>
---------	---

Си	<pre>s = 0; n=10; for (i = 0; i <= n; i++) if (i==n-i) s=s+A[i]+A[i+1];</pre>
----	--

Алгоритмический язык	<pre>s:=0 n:=10 нц для i от 0 до n если i = n-i то s:=s+A[i]+A[i+1] все кц</pre>
----------------------	--

В начале выполнения этого фрагмента в массиве находились числа 0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, т. е. $A[0]=0$, $A[1]=2$ и т. д. Чему будет равно значение переменной s после выполнения данной программы?

- 1) 0 2) 11 3) 22 4) 110

A13 Система команд исполнителя РОБОТ, «живущего» в прямоугольном лабиринте на клетчатой плоскости, состоит из 8 команд. Четыре команды – это команды-приказы:

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

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

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

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

Цикл
ПОКА условие
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)

В конструкциях ПОКА и ЕСЛИ условие может содержать команды проверки, а также слова И, ИЛИ, НЕ, обозначающие логические операции.

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

Сколько клеток лабиринта соответствуют требованию, что, начав движение в данной клетке и выполнив предложенную программу, РОБОТ уцелеет и остановится в закрашенной клетке (клетка F6)?

НАЧАЛО

ПОКА снизу свободно ИЛИ справа свободно

ПОКА снизу свободно

вниз

КОНЕЦ ПОКА

ЕСЛИ справа свободно

ТО

вправо

КОНЕЦ ЕСЛИ

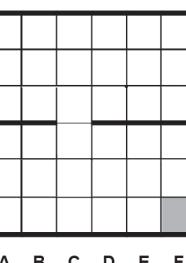
КОНЕЦ ПОКА

КОНЕЦ

1) 18

2) 24

3) 27



4) 30

Часть 2

Ответом к заданиям этой части (B1–B15) является число, последовательность букв или цифр. Впишите ответы сначала в текст работы, а затем перенесите их в бланк ответов № 1 справа от номера соответствующего задания, начиная с первой клеточки, без пробелов, запятых и других дополнительных символов. Каждую букву или цифру пишите в отдельной клеточке в соответствии с приведёнными в бланке образцами.

B1 У исполнителя Квадратор две команды, которым присвоены номера:

1. **прибавь 1,**
2. **возвели в квадрат.**

Первая из этих команд увеличивает число на экране на 1, вторая – возводит в квадрат. Программа для исполнителя Квадратор – это последовательность номеров команд.

Например, 21211 – это программа

возвели в квадрат

прибавь 1

возвели в квадрат

прибавь 1

прибавь 1

Эта программа преобразует число 2 в число 27.

Запишите программу, которая преобразует **число 2 в число 102** и содержит не более 6 команд. Если таких программ более одной, то запишите любую из них.

Ответ:

B2 Определите значение переменной **c** после выполнения следующего фрагмента программы (записанного ниже на разных языках программирования). Ответ запишите в виде целого числа.

Бейсик	<pre>a = 40 b = 6 a = a * 3 / b IF a < b THEN c = 2*a - 10 * b ELSE c = 2*a + 10 * b END IF</pre>
--------	--

Паскаль	<pre>a := 40; b := 6; a := a * 3 / b; if a < b then c := 2*a - 10 * b else c := 2*a + 10 * b;</pre>
---------	--

Си	<pre>a = 40; b = 6; a = a * 3 / b; if (a < b) c = 2*a - 10 * b; else c = 2*a + 10 * b;</pre>
----	---

Алгоритмический	<pre>a := 40 b := 6 a := a * 3 / b если a < b то c := 2*a - 10 * b иначе c := 2*a + 10 * b все</pre>
-----------------	---

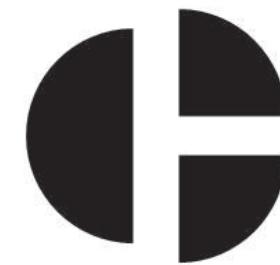
Ответ:

B3 Дан фрагмент электронной таблицы.

	A	B	C
1	5		=A1*2
2	=(B1-A1)/2	=B1-C1	=B2+A1

Какое число должно быть записано в ячейке B1, 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 двух 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

```
DIM N, S AS INTEGER
N = 0
S = 0
WHILE S <= 365
    S = S + 33
    N = N + 10
WEND
PRINT N
```

Паскаль

```
var n, s: integer;
begin
    n := 0;
    s := 0;
    while s <= 365 do
    begin
        s := s + 33;
        n := n + 10
    end;
    write(n)
end.
```

Си

```
#include<stdio.h>
void main()
{
    int n, s;
    n = 0;
    s = 0;
    while (s <= 365)
    {
        s = s + 33;
        n = n + 10;
    }
    printf("%d", n);
}
```

алг
нач

```
цел n, s
n := 0
s := 0
нц пока s <= 365
    s := s + 33
    n := n + 10
кц
вывод n
кон
```

Ответ:

B6 Алгоритм вычисления значения функции $F(n)$, где n – натуральное число, задан следующими соотношениями:

$$\begin{aligned} F(1) &= 1; \quad F(2)=1; \\ F(n) &= F(n-2) * n \text{ при } n > 2. \end{aligned}$$

Чему равно значение функции $F(8)$? В ответе запишите только натуральное число.

Ответ:

B7 Десятичное число 59 в некоторой системе счисления записывается как 214. Определите основание системы счисления.

Ответ:

B8 Ниже на 4 языках программирования 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 3, а потом 5.

Бейсик

```
DIM X, A, B AS INTEGER
INPUT X
A=0: B=0
WHILE X > 0
    A = A+1
    B = B +(X MOD 10)
    X = X \ 10
WEND
PRINT A
PRINT B
```

Паскаль

```

var x, a, b: integer;
begin
  readln(x);
  a:=0; b:=0;
  while x>0 do
  begin
    a:=a + 1;
    b:=b + (x mod 10);
    x:=x div 10;
  end;
  writeln(a); write(b);
end.
```

Си

```

#include<stdio.h>
void main()
{
  int x, a, b;
  scanf("%d", &x);
  a=0; b=0;
  while (x>0) {
    a = a+1;
    b = b +(x%10);
    x = x/10;
  }
  printf("%d\n%d", a, b);
}
```

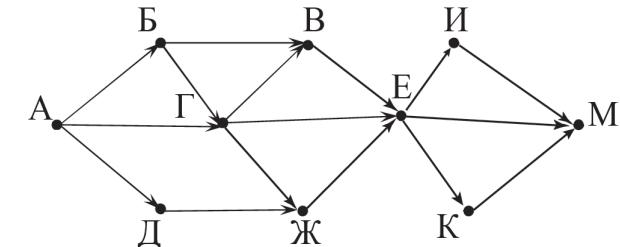
Алгоритмический

```

алг
нач
  цел x, a, b
  ввод x
  a:=0; b:=0
  нц пока x>0
    a:=a+1
    b:=b+mod(x,10)
    x:=div(x,10)
  кц
  вывод a, b
кон
```

Ответ: **B9**

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

**Ответ:** **B10**

Документ объёмом 8 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 25% от исходного,
- время, требуемое на сжатие документа – 10 секунд, на распаковку – 2 секунды?

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

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

Слова «секунд», «сек.», «с.» к ответу добавлять не нужно.

Ответ:

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

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

IP-адрес узла: 224.32.249.137

Маска: 255.255.240.0

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

A	B	C	D	E	F	G	H
255	249	240	224	137	32	8	0

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:

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)
Rишиль & (Amos Portos)	320
Rишиль & Amos	220
Rишиль & Amos & Portos	50

Какое количество страниц (в тысячах) будет найдено по запросу

Rишиль & Portos

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 2,
2. умножь на 2.

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

Программа для Удвоителя – это последовательность команд.

Сколько есть программ, которые число 2 преобразуют в число 42?

Ответ:

B14 Определите, какое число будет напечатано в результате выполнения следующего алгоритма (для Вашего удобства алгоритм представлен на четырёх языках):

Бейсик	<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) < R THEN M = T R = F(T) END IF NEXT T PRINT R FUNCTION F(x) F = 11*(9-x)*(9-x)+131 END FUNCTION </pre>
--------	---

Паскаль	<pre> var a,b,t,M,R :integer; Function F(x: integer):integer; begin F := 11*(9-x)*(9-x)+131; end; BEGIN a := -20; b := 20; M := a; R := F(a); for t := a to b do begin if (F(t) < R) then begin M := t; R := F(t); end; end; write(R); END. </pre>
---------	---

Си

```
#include<stdio.h>
int F(int x)
{
    return 11*(9-x)*(9-x)+131;
}
void main()
{
    int a, b, t, M, R;
    a = -20; b = 20;
    M = a; R = F(a);
    for (t=a; t<=b; t++) {
        if (F(t) < R) {
            M = t; R = F(t);
        }
    }
    printf("%d", R);
}
```

Алгоритмический

```
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) < R
            то
                M := t; R := F(t)
        все
    кон
    вывод R
кон
алг цел F(цел x)
нач
    знач := 11*(9-x)*(9-x)+131
кон
```

Ответ:

B15 Сколько существует различных наборов значений логических переменных $x_1, x_2, x_3, x_4, x_5, y_1, y_2, y_3, y_4, y_5$, которые удовлетворяют всем перечисленным ниже условиям?

$$(x_1 \rightarrow x_2) \wedge (x_2 \rightarrow x_3) \wedge (x_3 \rightarrow x_4) \wedge (x_4 \rightarrow x_5) = 1$$

$$(y_1 \rightarrow y_2) \wedge (y_2 \rightarrow y_3) \wedge (y_3 \rightarrow y_4) \wedge (y_4 \rightarrow y_5) = 1$$

$$y_1 \rightarrow x_1 = 1$$

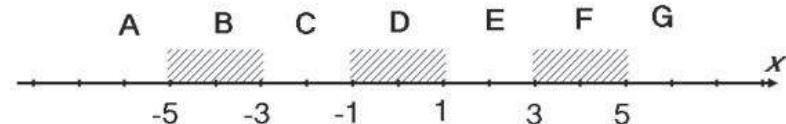
В ответе **не нужно** перечислять все различные наборы значений переменных $x_1, x_2, x_3, x_4, x_5, y_1, y_2, y_3, y_4, y_5$, при которых выполнена данная система равенств. В качестве ответа Вам нужно указать количество таких наборов.

Ответ:

Часть 3

Для записи ответов к заданиям этой части (C1–C4) используйте бланк ответов № 2. Запишите сначала номер задания (C1 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 (включая границы). Программист торопился и написал программу неправильно.



Бейсик	<pre>INPUT x IF x>=-5 OR x<=1 THEN IF x>=-3 AND x<=3 THEN IF x>=-1 AND x<=5 THEN PRINT "принадлежит" ELSE PRINT "не принадлежит" END IF END IF END IF</pre>
---------------	---

Паскаль

```
var x: real;
begin
    readln(x);
    if (x>=-5) or (x<=1) then
        if (x>=-3) and (x<=3) then
            if (x>=-1) and (x<=5) then
                write('принадлежит')
            else
                write('не принадлежит')
        end.
```

Си

```
#include <stdio.h>
void main(){
    float x;
    scanf("%f", &x);
    if (x>=-5 || x<=1)
        if (x>=-3 && x<=3)
            if (x>=-1 && x<=5)
                printf("принадлежит");
            else
                printf("не принадлежит");
}
```

Алгоритмический язык

```
алг
нач
    веш x
    ввод x
    если x>=-5 или x<=1 то
        если x>=-3 и x<=3 то
            если x>=-1 и x<=5 то
                вывод 'принадлежит'
            иначе
                вывод 'не принадлежит'
            все
    все
кон
```

Последовательно выполните следующее.

- Пересвойте и заполните таблицу, которая показывает, как работает программа при аргументе, принадлежащем различным областям (A, B, C, D, E, F, G). Точки, лежащие на границах областей, отдельно не рассматривать.

Область	Условие 1 ($x \geq -5$ или $x \leq 1$)	Условие 2 ($x \geq -3$ и $x \leq 3$)	Условие 3 ($x \geq -1$ и $x \leq 5$)	Программа выведет	Область обрабатывается верно
A					
B					
C					
D					
E					
F					
G					

В столбцах условий укажите «да», если условие выполнится, «нет», если условие не выполнится, «—» (прочерк), если условие не будет проверяться, «не изв.», если программа ведёт 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».

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

C2 Дан массив, содержащий 70 целых чисел. Опишите на одном из языков программирования алгоритм, позволяющий найти и вывести наименьшее содержащееся в массиве положительное число, десятична запись которого не оканчивается цифрой 7. Гарантируется, что в массиве есть хотя бы один положительный элемент, десятичная запись которого не оканчивается цифрой 7.

Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

Паскаль

```
const
    N=70;
var
    a: array [1..N] of integer;
    i, j, m: integer;
begin
    for i:=1 to N do
        readln(a[i]);
    ...
end.
```

Бейсик

```
N=70
DIM A(N) AS INTEGER
DIM I, J, M AS INTEGER
FOR I = 1 TO N
    INPUT A(I)
NEXT I
...
END
```

Си

```
#include <stdio.h>
#define N 70
void main(){
    int a[N];
    int I, j, m;
    for (i=0; i<N; i++)
        scanf("%d", &a[i]);
    ...
}
```

Алгоритмический язык

```
алг
нач
    цел N=70
    целтаб a[1:N]
    цел i, j, m
    нц для i от 1 до N
        ввод a[i]
    кц
    ...
кон
```

В качестве ответа Вам необходимо привести фрагмент программы, 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.

C3

Два игрока, Паша и Вова, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Паша. За один ход игрок может добавить в кучу 1 камень или 10 камней. Например, имея кучу из 7 камней, за один ход можно получить кучу из 8 или 17 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче становится не менее 41. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 41 или больше камней.

В начальный момент в куче было S камней, $1 \leq S \leq 40$.

Будем говорить, что игрок имеет *выигрышную стратегию*, если он может выиграть при любых ходах противника. Описать стратегию игрока – значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника.

Выполните следующие задания. Во всех случаях обосновывайте свой ответ.

1. а) Укажите все такие значения числа S , при которых Паша может выиграть в один ход. Обоснуйте, что найдены все нужные значения S , и укажите выигрышающие ходы.

б) Укажите такое значение S , при котором Паша не может выиграть за один ход, но при любом ходе Паша Вова может выиграть своим первым ходом. Опишите выигрышную стратегию Вовы.

2. Укажите два значения S , при которых у Паши есть выигрышная стратегия, причём Паши не может выиграть за один ход, но может выиграть своим вторым ходом независимо от того, как будет ходить Вова. Для указанных значений S опишите выигрышную стратегию Паши.

3. Укажите значение S , при котором у Вовы есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Паши, однако у Вовы нет стратегии, которая позволит ему гарантированно выиграть первым ходом. Для указанного значения S опишите выигрышную стратегию Вовы. Постройте дерево всех партий, возможных при этой выигрышной стратегии Вовы (в виде рисунка или таблицы). На ребрах дерева указывайте, кто делает ход, в узлах – количество камней в куче.

C4 На плоскости дан набор точек с целочисленными координатами. Необходимо найти такой треугольник наибольшей площади с вершинами в этих точках, у которого нет общих точек с осью Ox , а одна из сторон лежит на оси Oy .

Напишите эффективную, в том числе по памяти, программу, которая будет решать эту задачу. Размер памяти, которую использует Ваша программа, не должен зависеть от количества точек.

Перед текстом программы кратко опишите используемый алгоритм решения задачи и укажите используемый язык программирования и его версию.

Описание входных данных

В первой строке вводится одно целое положительное число – количество точек N .

Каждая из следующих N строк содержит два целых числа – сначала координата x , затем координата y очередной точки. Числа разделены пробелом.

Описание выходных данных

Программа должна вывести одно число – максимальную площадь треугольника, удовлетворяющего условиям задачи. Если такого треугольника не существует, программа должна вывести ноль.

Пример входных данных:

```
8
0 -10
0 2
4 0
3 3
0 7
0 4
5 5
-9 9
```

Пример выходных данных для приведённого выше примера входных данных:

22.5

Диагностическая работа № 1

по ИНФОРМАТИКЕ

14 декабря 2012 года

11 класс

Вариант 3

Информатика. 11 класс. Вариант 3

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 развёрнутый 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 удается 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!

Район. _____

Город (населённый пункт) _____

Школа _____

Класс _____

Фамилия. _____

Имя. _____

Отчество _____

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, $A \vee B$) либо $|$ (например, $A | B$);
- 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 А и В 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, содержащие 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, $\neg A \wedge B \vee C \wedge D$ совпадает с $((\neg A) \wedge B) \vee (C \wedge D)$. 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. Везде в задачах 1 Кбайт = 1024 байт; 1 Мбайт = 1024 Кбайт.

Часть 1

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

A1 Для каждого из перечисленных ниже десятичных чисел построили двоичную запись. Укажите число, двоичная запись которого содержит наибольшее количество единиц.

- 1) 13 2) 14 3) 15 4) 16

A2 Между населёнными пунктами A, B, C, D, E, F, Z построены дороги, протяжённость которых приведена в таблице. (Отсутствие числа в таблице означает, что прямой дороги между пунктами нет.)

	A	B	C	D	E	F	Z
A		4	6				33
B	4		1				
C	6	1		5			27
D			5		4	8	10
E				4		1	8
F				8	1		2
Z	33		27	10	8	2	

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

- 1) 13 2) 17 3) 21 4) 33

A3 Дан фрагмент таблицы истинности выражения F:

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

Каким выражением может быть F?

- 1) $x_1 \wedge \neg x_2 \wedge x_3 \wedge \neg x_4 \wedge x_5 \wedge \neg x_6 \wedge x_7$
 2) $x_1 \vee \neg x_2 \vee x_3 \vee \neg x_4 \vee x_5 \vee \neg x_6 \vee x_7$
 3) $\neg x_1 \vee x_2 \vee \neg x_3 \vee x_4 \vee \neg x_5 \vee x_6 \vee \neg x_7$
 4) $\neg x_1 \wedge x_2 \wedge \neg x_3 \wedge x_4 \wedge \neg x_5 \wedge x_6 \wedge \neg x_7$

A4 Для групповых операций с файлами используются **маски имён файлов**. Мaska 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:

korsten.docx

mikor5.docx

mokkorte.dat

mokkorte.doc

skorcher.doc

x-korvet.doc

Определите, по какой из масок из каталога будет отобрана указанная группа файлов:

mikor5.docx

mokkorte.doc

skorcher.doc

x-korvet.doc

1) *kor?*.d*

3) *?kor?*.do*

2) ?kor*.doc

4) *kor?.doc*

A5 Автомат получает на вход четырёхзначное восьмеричное число. По этому числу строится новое число по следующим правилам.

1. Складываются первая и вторая, а также третья и четвёртая цифры.

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

Пример. Исходное число: 4531. Суммы: $4+5 = 11$; $3+1 = 4$. Результат: 411.

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

1) 117

2) 1213

3) 1511

4) 1517

A6 В фрагменте базы данных представлены сведения о родственных отношениях. На основании приведённых данных определите фамилию и инициалы дедушки Корзуна П.А.

Таблица 1

ID	Фамилия_И.О.	Пол
1072	Онищенко А.Б.	Ж
1028	Онищенко Б.Ф.	М
1099	Онищенко И.Б.	М
1178	Онищенко П.И.	М
1156	Онищенко Т.И.	Ж
1065	Корзун А.И.	Ж
1131	Корзун А.П.	М
1061	Корзун Л.Г.	М
1217	Корзун П.А.	М
1202	Зельдович М.А.	Ж
1027	Витюк Д.К.	Ж
1040	Лемешко В.А.	Ж
1046	Месяц К.Г.	М
1187	Лукина Р.Г.	Ж
1093	Фокс П.А.	Ж
1110	Друк Г.Р.	Ж
...

Таблица 2

ID_Родителя	ID_Ребёнка
1027	1072
1027	1099
1028	1072
1028	1099
1072	1040
1072	1202
1072	1217
1099	1156
1099	1178
1110	1156
1110	1178
1131	1040
1131	1202
1131	1217
1187	1061
1187	1093
...	...

1) Витюк Д.К.

3) Корзун Л.Г.

2) Корзун А.П.

4) Онищенко Б.Ф.

A7 В ячейки диапазона C3:F6 электронной таблицы записаны числа, как показано на рисунке.

	A	B	C	D	E	F
1						
2						
3		1	2	3	4	
4		11	13	15	17	
5		21	24	27	30	
6		31	35	39	43	

В ячейке B2 записали формулу =E\$5-\$D4. После этого ячейку B2 скопировали в ячейку A1. Какое число будет показано в ячейке A1?

Примечание: знак \$ используется для обозначения абсолютной адресации.

- 1) 22 2) 14 3) 12 4) 4

A8 Производилась двухканальная (стерео) звукозапись с частотой дискретизации 16 кГц и 24-битным разрешением. В результате был получен файл размером 60 Мбайт, сжатие данных не производилось. Какая из приведённых ниже величин наиболее близка к времени, в течение которого проводилась запись?

- 1) 2 мин 2) 5 мин 3) 10 мин 4) 15 мин

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. Для букв А, Б, В и Г использовали такие кодовые слова: А – 111, Б – 110, В – 101, Г – 100.

Укажите, каким кодовым словом из перечисленных ниже может быть закодирована буква Д.

Код должен удовлетворять свойству однозначного декодирования. Если можно использовать более одного кодового слова, укажите кратчайшее из них.

- 1) 1 2) 0 3) 01 4) 10

A10 На числовой прямой даны два отрезка: $P = [4, 16]$ и $Q = [9, 18]$.

Выберите такой отрезок A, что формула

$$((x \in A) \rightarrow (x \in P)) \vee (x \in Q)$$

тождественно истинна, то есть принимает значение 1 при любом значении переменной x.

- 1) [1, 11] 2) [5, 15] 3) [11, 21] 4) [15, 25]

A11 При регистрации в компьютерной системе каждому пользователю выдаётся пароль, состоящий из 15 символов и содержащий только символы К, О, М, П, Б, Ю, Т, Е, Р. Каждый такой пароль 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 30 паролей.

- 1) 180 байт 2) 210 байт 3) 240 байт 4) 270 байт

A12 В программе описан одномерный целочисленный массив с индексами от 0 до 10. Ниже представлен записанный на разных языках программирования фрагмент одной и той же программы, обрабатывающей данный массив.

Бейсик	<pre>s = 0 n = 10 FOR i = 0 TO n IF i=n-i THEN s = s+A(i)+A(i+1) END IF NEXT i</pre>
--------	--

Паскаль	<pre>s:=0; n:=10; for i:=0 to n do begin if i = n-i then s:=s+A[i]+A[i+1]; end;</pre>
---------	---

Си	<pre>s = 0; n=10; for (i = 0; i <= n; i++) if (i==n-i) s=s+A[i]+A[i+1];</pre>
----	--

Алгоритмический язык	<pre>s:=0 n:=10 нц для i от 0 до n если i = n-i то s:=s+A[i]+A[i+1] все кц</pre>
----------------------	--

В начале выполнения этого фрагмента в массиве находились числа 0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, т. е. $A[0]=0$, $A[1]=2$ и т. д. Чему будет равно значение переменной s после выполнения данной программы?

- 1) 0 2) 11 3) 22 4) 110

A13 Система команд исполнителя РОБОТ, «живущего» в прямоугольном лабиринте на клетчатой плоскости, состоит из 8 команд. Четыре команды – это команды-приказы:

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

При выполнении любой из этих команд РОБОТ перемещается на одну клетку соответственно: вверх \uparrow , вниз \downarrow , влево \leftarrow , вправо \rightarrow .

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

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

Цикл
ПОКА условие
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)

В конструкциях ПОКА и ЕСЛИ условие может содержать команды проверки, а также слова И, ИЛИ, НЕ, обозначающие логические операции.

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

Сколько клеток лабиринта соответствуют требованию, что, начав движение в данной клетке и выполнив предложенную программу, РОБОТ уцелеет и остановится в закрашенной клетке (клетка F6)?

НАЧАЛО
ПОКА снизу свободно ИЛИ справа свободно
ПОКА снизу свободно

вниз

КОНЕЦ ПОКА

ЕСЛИ справа свободно

ТО

вправо

КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

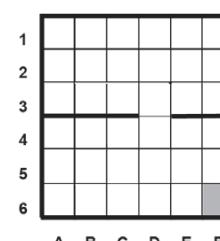
КОНЕЦ

1) 18

2) 24

3) 27

4) 30



Часть 2

Ответом к заданиям этой части (B1–B15) является число, последовательность букв или цифр. Впишите ответы сначала в текст работы, а затем перенесите их в бланк ответов № 1 справа от номера соответствующего задания, начиная с первой клеточки, без пробелов, запятых и других дополнительных символов. Каждую букву или цифру пишите в отдельной клеточке в соответствии с приведёнными в бланке образцами.

B1 У исполнителя Квадратор две команды, которым присвоены номера:

1. **прибавь 1,**
2. **возвели в квадрат.**

Первая из этих команд увеличивает число на экране на 1, вторая – возводит в квадрат. Программа для исполнителя Квадратор – это последовательность номеров команд.

Например, 21211 – это программа

возвели в квадрат

прибавь 1

возвели в квадрат

прибавь 1

прибавь 1

Эта программа преобразует число 2 в число 27.

Запишите программу, которая преобразует **число 2 в число 102** и содержит не более 6 команд. Если таких программ более одной, то запишите любую из них.

Ответ:

B2 Определите значение переменной **c** после выполнения следующего фрагмента программы (записанного ниже на разных языках программирования). Ответ запишите в виде целого числа.

Бейсик	<pre>a = 30 b = 6 a = a * 3 / b IF a < b THEN c = 2*a - 10 * b ELSE c = 2*a + 10 * b END IF</pre>
--------	--

Паскаль	<pre>a := 30; b := 6; a := a * 3 / b; if a < b then c := 2*a - 10 * b else c := 2*a + 10 * b;</pre>
---------	--

Си	<pre>a = 30; b = 6; a = a * 3 / b; if (a < b) c = 2*a - 10 * b; else c = 2*a + 10 * b;</pre>
----	---

Алгоритмический	<pre>a := 30 b := 6 a := a * 3 / b если a < b то с := 2*a - 10 * b иначе с := 2*a + 10 * b все</pre>
-----------------	---

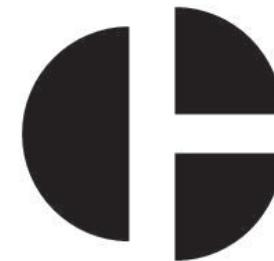
Ответ:

B3 Дан фрагмент электронной таблицы.

	A	B	C
1	5		=A1*2
2	=(B1-A1)/2	=B1-C1	=B2+A1

Какое число должно быть записано в ячейке B1, 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 более пяти** сигналов (точек и тире)?

Ответ:

B5

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

Бейсик

```
DIM N, S AS INTEGER
N = 0
S = 0
WHILE S <= 365
    S = S + 33
    N = N + 10
WEND
PRINT N
```

Паскаль

```
var n, s: integer;
begin
    n := 0;
    s := 0;
    while s <= 365 do
    begin
        s := s + 33;
        n := n + 10
    end;
    write(n)
end.
```

Си

```
#include<stdio.h>
void main()
{
    int n, s;
    n = 0;
    s = 0;
    while (s <= 365)
    {
        s = s + 33;
        n = n + 10;
    }
    printf("%d", n);
}
```

алг
нач

```
цел n, s
n := 0
s := 0
нц пока s <= 365
    s := s + 33
    n := n + 10
кц
вывод n
кон
```

Ответ: **B6**

Алгоритм вычисления значения функции $F(n)$, где n – натуральное число, задан следующими соотношениями:

$$\begin{aligned} F(1) &= 1; \quad F(2)=1; \\ F(n) &= F(n-2) * n \text{ при } n > 2. \end{aligned}$$

Чему равно значение функции $F(7)$? В ответе запишите только натуральное число.

Ответ: **B7**

Десятичное число 59 в некоторой системе счисления записывается как 214. Определите основание системы счисления.

Ответ: **B8**

Ниже на 4 языках программирования 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 3, а потом 7.

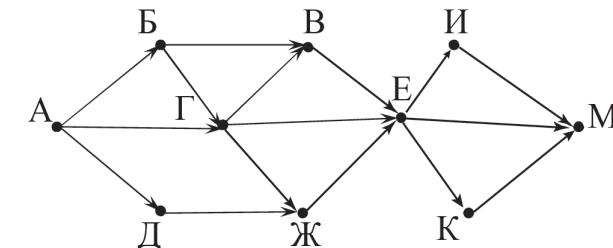
Бейсик

```
DIM X, A, B AS INTEGER
INPUT X
A=0: B=0
WHILE X > 0
    A = A+1
    B = B + (X MOD 10)
    X = X \ 10
WEND
PRINT A
PRINT B
```

Паскаль <pre>var x, a, b: integer; begin readln(x); a:=0; b:=0; while x>0 do begin a:=a + 1; b:=b + (x mod 10); x:=x div 10; end; writeln(a); write(b); end.</pre>
Си <pre>#include<stdio.h> void main() { int x, a, b; scanf("%d", &x); a=0; b=0; while (x>0) { a = a+1; b = b +(x%10); x = x/10; } printf("%d\n%d", a, b); }</pre>
Алгоритмический <pre>алг нач цел x, a, b ввод x a:=0; b:=0 нц пока x>0 a:=a+1 b:=b+mod(x,10) x:=div(x,10) кц вывод a, b кон</pre>

Ответ:

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



Ответ:

- B10** Документ объёмом 15 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% от исходного,
 – время, требуемое на сжатие документа, – 18 секунд, на распаковку – 2 секунды?

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

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

Слова «секунд», «сек.», «с.» к ответу добавлять не нужно.

Ответ:

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

IP-адрес узла: 224.32.249.137

Маска: 255.255.240.0

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

A	B	C	D	E	F	G	H
255	249	240	224	137	32	8	0

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:

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)
Лебедь & (Рак Щука)	320
Лебедь & Рак	200
Лебедь & Рак & Щука	50

Какое количество страниц (в тысячах) будет найдено по запросу
Лебедь & Щука

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 2,
2. умножь на 2.

Первая из них увеличивает на 2 число на экране, вторая удваивает его. Программа для Удвоителя – это последовательность команд. Сколько есть программ, которые число 2 преобразуют в число 42?

Ответ:

B14 Определите, какое число будет напечатано в результате выполнения следующего алгоритма (для Вашего удобства алгоритм представлен на четырёх языках):

Бейсик	<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) < R THEN M = T R = F(T) END IF NEXT T PRINT R FUNCTION F(x) F = 16*(9-x)*(9-x)+127 END FUNCTION </pre>
--------	---

Паскаль	<pre> var a,b,t,M,R :integer; Function F(x: integer):integer; begin F := 16*(9-x)*(9-x)+127; end; BEGIN a := -20; b := 20; M := a; R := F(a); for t := a to b do begin if (F(t) < R) then begin M := t; R := F(t); end; end; write(R); END. </pre>
---------	---

Си

```
#include<stdio.h>
int F(int x)
{
    return 16*(9-x)*(9-x)+127;
}
void main()
{
    int a, b, t, M, R;
    a = -20; b = 20;
    M = a; R = F(a);
    for (t=a; t<=b; t++) {
        if (F(t) < R) {
            M = t; R = F(t);
        }
    }
    printf("%d", R);
}
```

Алгоритмический

```
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) < R
            то
                M := t; R := F(t)
        все
    вывод R
кон
алг цел F(цел x)
нач
    знач := 16*(9-x)*(9-x)+127
кон
```

Ответ:

B15 Сколько существует различных наборов значений логических переменных $x_1, x_2, x_3, x_4, x_5, y_1, y_2, y_3, y_4, y_5$, которые удовлетворяют всем перечисленным ниже условиям?

$$(x_1 \rightarrow x_2) \wedge (x_2 \rightarrow x_3) \wedge (x_3 \rightarrow x_4) \wedge (x_4 \rightarrow x_5) = 1$$

$$(y_1 \rightarrow y_2) \wedge (y_2 \rightarrow y_3) \wedge (y_3 \rightarrow y_4) \wedge (y_4 \rightarrow y_5) = 1$$

$$y_1 \rightarrow x_1 = 1$$

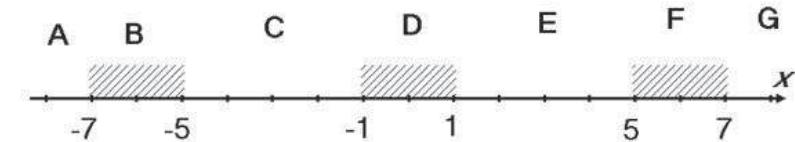
В ответе **не нужно** перечислять все различные наборы значений переменных $x_1, x_2, x_3, x_4, x_5, y_1, y_2, y_3, y_4, y_5$, при которых выполнена данная система равенств. В качестве ответа Вам нужно указать количество таких наборов.

Ответ:

Часть 3

Для записи ответов к заданиям этой части (C1–C4) используйте бланк ответов № 2. Запишите сначала номер задания (C1 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 (включая границы). Программист торопился и написал программу неправильно.



Бейсик	<pre>INPUT x IF x>=-7 OR x<=1 THEN IF x>=-5 AND x<=5 THEN IF x>=-1 AND x<=7 THEN PRINT "принадлежит" ELSE PRINT "не принадлежит" END IF END IF END IF</pre>
---------------	---

Паскаль

```

var x: real;
begin
    readln(x);
    if (x>=-7) or (x<=1) then
        if (x>=-5) and (x<=5) then
            if (x>=-1) and (x<=7) then
                write('принадлежит')
            else
                write('не принадлежит')
        end.
    
```

Си

```

#include <stdio.h>
void main(){
    float x;
    scanf("%f", &x);
    if (x>=-7 || x<=1)
        if (x>=-5 && x<=5)
            if (x>=-1 && x<=7)
                printf("принадлежит");
            else
                printf("не принадлежит");
    }
}

```

Алгоритмический язык

```

алг
нач
    веш x
    ввод x
    если x>=-7 или x<=1 то
        если x>=-5 и x<=5 то
            если x>=-1 и x<=7 то
                вывод 'принадлежит'
            иначе
                вывод 'не принадлежит'
            все
        все
    кон

```

Последовательно выполните следующее.

- Пересвойте и заполните таблицу, которая показывает, как работает программа при аргументе, принадлежащем различным областям (A, B, C, D, E, F, G). Точки, лежащие на границах областей, отдельно не рассматривать.

Область	Условие 1 ($x \geq -7$ или $x \leq 1$)	Условие 2 ($x \geq -5$ и $x \leq 5$)	Условие 3 ($x \geq -1$ и $x \leq 7$)	Программа выведет	Область обрабатывается верно
A					
B					
C					
D					
E					
F					
G					

В столбцах условий укажите «да», если условие выполнится, «нет», если условие не выполнится, «—» (прочерк), если условие не будет проверяться, «не изв.», если программа ведёт 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 массив, содержащий 70 целых чисел. Опишите на одном из языков программирования алгоритм, позволяющий найти и вывести наименьшее содержащееся в массиве положительное число, десятичная запись которого не оканчивается цифрой 7. Гарантируется, что в массиве есть хотя бы один положительный элемент, десятичная запись которого не оканчивается цифрой 7.

Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

Паскаль

```

const
    N=70;
var
    a: array [1..N] of integer;
    i, j, m: integer;
begin
    for i:=1 to N do
        readln(a[i]);
    ...
end.

```

Бейсик

```
N=70
DIM A(N) AS INTEGER
DIM I, J, M AS INTEGER
FOR I = 1 TO N
    INPUT A(I)
NEXT I
...
END
```

Си

```
#include <stdio.h>
#define N 70
void main(){
    int a[N];
    int I, j, m;
    for (i=0; i<N; i++)
        scanf("%d", &a[i]);
    ...
}
```

Алгоритмический язык

```
алг
нач
    цел N=70
    целтаб a[1:N]
    цел i, j, m
    нц для i от 1 до N
        ввод a[i]
    кц
    ...
кон
```

В качестве ответа Вам необходимо привести фрагмент программы, 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.

C3

Два игрока, Паша и Вова, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Паша. За один ход игрок может добавить в кучу 1 камень или 10 камней. Например, имея кучу из 7 камней, за один ход можно получить кучу из 8 или 17 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче становится не менее 31. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 31 или больше камней.

В начальный момент в куче было S камней, $1 \leq S \leq 30$.

Будем говорить, что игрок имеет *выигрышную стратегию*, если он может выиграть при любых ходах противника. Описать стратегию игрока – значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника.

Выполните следующие задания. Во всех случаях обосновывайте свой ответ.

1. а) Укажите все такие значения числа S , при которых Паша может выиграть в один ход. Обоснуйте, что найдены все нужные значения S , и укажите выигрышающие ходы.

б) Укажите такое значение S , при котором Паша не может выиграть за один ход, но при любом ходе Паша Вова может выиграть своим первым ходом. Опишите выигрышную стратегию Вовы.

2. Укажите два значения S , при которых у Паши есть выигрышная стратегия, причём Паши не может выиграть за один ход, но может выиграть своим вторым ходом независимо от того, как будет ходить Вова. Для указанных значений S опишите выигрышную стратегию Паши.

3. Укажите значение S , при котором у Вовы есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Паши, однако у Вовы нет стратегии, которая позволит ему гарантированно выиграть первым ходом. Для указанного значения S опишите выигрышную стратегию Вовы. Постройте дерево всех партий, возможных при этой выигрышной стратегии Вовы (в виде рисунка или таблицы). На ребрах дерева указывайте, кто делает ход, в узлах – количество камней в куче.

C4 На плоскости дан набор точек с целочисленными координатами. Необходимо найти такой треугольник наибольшей площади с вершинами в этих точках, у которого нет общих точек с осью Ox , а одна из сторон лежит на оси Oy .

Напишите эффективную, в том числе по памяти, программу, которая будет решать эту задачу. Размер памяти, которую использует Ваша программа, не должен зависеть от количества точек.

Перед текстом программы кратко опишите используемый алгоритм решения задачи и укажите используемый язык программирования и его версию.

Описание входных данных

В первой строке вводится одно целое положительное число – количество точек N .

Каждая из следующих N строк содержит два целых числа – сначала координата x , затем координата y очередной точки. Числа разделены пробелом.

Описание выходных данных

Программа должна вывести одно число – максимальную площадь треугольника, удовлетворяющего условиям задачи. Если такого треугольника не существует, программа должна вывести ноль.

Пример входных данных:

```
8
0 -10
0 2
4 0
3 3
0 7
0 4
5 5
-9 9
```

Пример выходных данных для приведённого выше примера входных данных:

22.5

Диагностическая работа № 1

по ИНФОРМАТИКЕ

14 декабря 2012 года

11 класс

Вариант 4

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 развёрнутый 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 удается 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!

Район. _____

Город (населённый пункт) _____

Школа _____

Класс _____

Фамилия. _____

Имя. _____

Отчество _____

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, $A \vee B$) либо $|$ (например, $A | B$);
- 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 А и В 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, содержащие 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, $\neg A \wedge B \vee C \wedge D$ совпадает с $((\neg A) \wedge B) \vee (C \wedge D)$. 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. Везде в задачах 1 Кбайт = 1024 байт; 1 Мбайт = 1024 Кбайт.

Часть 1

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

A1 Для каждого из перечисленных ниже десятичных чисел построили двоичную запись. Укажите число, двоичная запись которого содержит наибольшее количество значащих нулей.

- 1) 1 2) 7 3) 11 4) 15

A2 Между населёнными пунктами A, B, C, D, E, F, Z построены дороги, протяжённость которых приведена в таблице. (Отсутствие числа в таблице означает, что прямой дороги между пунктами нет.)

	A	B	C	D	E	F	Z
A		4	6				43
B	4		1				
C	6	1		15			32
D			15		4	6	10
E				4			8
F				6			2
Z	43		32	10	8	2	

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

- 1) 28 2) 38 3) 41 4) 43

A3 Дан фрагмент таблицы истинности выражения F:

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

Каким выражением может быть F?

- 1) $x_1 \wedge x_2 \wedge \neg x_3 \wedge x_4 \wedge x_5 \wedge \neg x_6 \wedge x_7$
 2) $x_1 \vee x_2 \vee \neg x_3 \vee x_4 \vee x_5 \vee \neg x_6 \vee x_7$
 3) $\neg x_1 \vee \neg x_2 \vee x_3 \vee \neg x_4 \vee \neg x_5 \vee x_6 \vee \neg x_7$
 4) $\neg x_1 \wedge \neg x_2 \wedge x_3 \wedge \neg x_4 \wedge \neg x_5 \wedge x_6 \wedge \neg x_7$

A4 Для групповых операций с файлами используются **маски имён файлов**. Мaska 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:

astarta.doc
catarsis.dat
catarsis.doc
plataria.docx
start.doc
tartar.docx

Определите, по какой из масок из каталога будет отобрана указанная группа файлов:

astarta.doc
catarsis.doc
plataria.docx
start.doc

- 1) *tar?*.d*
2) ?tar*.doc

- 3) *?tar?*.do*
4) *tar?.doc*

A5 Автомат получает на вход четырёхзначное восьмеричное число. По этому числу строится новое число по следующим правилам.

1. Складываются первая и вторая, а также третья и четвёртая цифры.
2. Полученные два числа записываются друг за другом в порядке возрастания (без разделителей).

Пример. Исходное число: 4531. Суммы: $4+5=11$; $3+1=4$. Результат: 411.

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

- 1) 811 2) 717 3) 1511 4) 1214

A6 В фрагменте базы данных представлены сведения о родственных отношениях. На основании приведённых данных определите фамилию и инициалы бабушки Гейко П.А.

Таблица 1

ID	Фамилия_И.О.	Пол
940	Карпец Д.К.	Ж
941	Базилевич Б.Ф.	М
953	Лемешко В.А.	Ж
959	Месхи К.Г.	М
974	Гейко Л.А.	Ж
978	Гейко А.И.	Ж
985	Базилевич А.Б.	Ж
1006	Фирс П.А.	Ж
1012	Базилевич И.Б.	М
1023	Фоменко Г.Р.	Ж
1044	Гейко А.П.	М
1069	Базилевич Т.И.	Ж
1091	Базилевич П.И.	М
1100	Мухина Р.Г.	Ж
1115	Звавич М.А	Ж
1130	Гейко П.А.	М
....

Таблица 2

ID_Родителя	ID_Ребёнка
940	985
940	1012
941	985
941	1012
985	953
985	1115
985	1130
1012	1069
1012	1091
1023	1069
1023	1091
1044	953
1044	1115
1044	1130
1100	974
1100	1006
...	...

- 1) Базилевич Б.Ф. 2) Гейко А.И. 3) Гейко Л.А. 4) Карпец Д.К.

A7 В ячейки диапазона C3:F6 электронной таблицы записаны числа, как показано на рисунке.

	A	B	C	D	E	F
1						
2						
3		1	2	3	4	
4		11	22	33	44	
5		22	11	44	33	
6		31	35	39	43	

В ячейке B2 записали формулу =E\$5+\$D4. После этого ячейку B2 скопировали в ячейку A1. Какое число будет показано в ячейке A1?

Примечание: знак \$ используется для обозначения абсолютной адресации.

- 1) 13 2) 23 3) 44 4) 66

A8 Производилась двухканальная (стерео) звукозапись с частотой дискретизации 32 кГц и 24-битным разрешением. В результате был получен файл размером 60 Мбайт, сжатие данных не производилось. Какая из приведённых ниже величин наиболее близка к времени, в течение которого проводилась запись?

- 1) 2 мин 2) 5 мин 3) 10 мин 4) 15 мин

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. Для букв А, Б, В и Г использовали такие кодовые слова: А – 000, Б – 001, В – 010, Г – 011.

Укажите, каким кодовым словом из перечисленных ниже может быть закодирована буква Д.

Код должен удовлетворять свойству однозначного декодирования. Если можно использовать более одного кодового слова, укажите кратчайшее из них.

- 1) 1 2) 0 3) 01 4) 10

A10 На числовой прямой даны два отрезка: $P = [3, 9]$ и $Q = [4, 12]$.

Выберите такой отрезок A, что формула

$$((x \in A) \rightarrow (x \in P)) \vee (x \in Q)$$

тождественно истинна, то есть принимает значение 1 при любом значении переменной x.

- 1) [0, 5] 2) [5, 10] 3) [10, 15] 4) [15, 20]

A11 При регистрации в компьютерной системе каждому пользователю выдаётся пароль, состоящий из 15 символов и содержащий только символы К, О, М, П, Б, Ю, Т, Е, Р. Каждый такой пароль 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 25 паролей.

- 1) 150 байт 2) 200 байт 3) 225 байт 4) 250 байт

A12 В программе описан одномерный целочисленный массив с индексами от 0 до 10. Ниже представлен записанный на разных языках программирования фрагмент одной и той же программы, обрабатывающей данный массив.

Бейсик	<pre>s = 0 n = 10 FOR i = 0 TO n IF i=n-i THEN s = s+A(i)+A(i+1) END IF NEXT i</pre>
--------	--

Паскаль	<pre>s:=0; n:=10; for i:=0 to n do begin if i = n-i then s:=s+A[i]+A[i+1]; end;</pre>
---------	---

Си	<pre>s = 0; n=10; for (i = 0; i <= n; i++) if(i==n-i) s=s+A[i]+A[i+1];</pre>
----	---

Алгоритмический язык	<pre>s:=0 n:=10 нц для i от 0 до n если i = n-i то s:=s+A[i]+A[i+1] все кц</pre>
----------------------	--

В начале выполнения этого фрагмента в массиве находились числа 0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, т. е. $A[0]=0$, $A[1]=10$ и т. д. Чему будет равно значение переменной s после выполнения данной программы?

- 1) 0 2) 100 3) 110 4) 550

A13 Система команд исполнителя РОБОТ, «живущего» в прямоугольном лабиринте на клетчатой плоскости, состоит из 8 команд. Четыре команды – это команды-приказы:

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

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

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

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

Цикл
ПОКА условие
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)

В конструкциях ПОКА и ЕСЛИ условие может содержать команды проверки, а также слова И, ИЛИ, НЕ, обозначающие логические операции.

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

Сколько клеток лабиринта соответствуют требованию, что, начав движение в данной клетке и выполнив предложенную программу, РОБОТ уцелеет и остановится в закрашенной клетке (клетка F6)?

НАЧАЛО

ПОКА снизу свободно ИЛИ справа свободно

ПОКА снизу свободно

вниз

КОНЕЦ ПОКА

ЕСЛИ справа свободно

ТО

вправо

КОНЕЦ ЕСЛИ

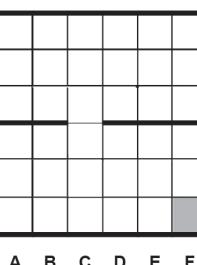
КОНЕЦ ПОКА

КОНЕЦ

1) 18

2) 24

3) 27



4) 30

Часть 2

Ответом к заданиям этой части (B1–B15) является число, последовательность букв или цифр. Впишите ответы сначала в текст работы, а затем перенесите их в бланк ответов № 1 справа от номера соответствующего задания, начиная с первой клеточки, без пробелов, запятых и других дополнительных символов. Каждую букву или цифру пишите в отдельной клеточке в соответствии с приведёнными в бланке образцами.

B1 У исполнителя Квадр две команды, которым присвоены номера:

1. прибавь 1,
2. возвели в квадрат.

Первая из этих команд увеличивает число на экране на 1, вторая – возводит в квадрат. Программа для исполнителя Квадр – это последовательность номеров команд.

Например, 21121 – это программа

возвели в квадрат

прибавь 1

прибавь 1

возвели в квадрат

прибавь 1

Эта программа преобразует число 3 в число 122.

Запишите программу, которая преобразует число 1 в число 101 и содержит не более 6 команд. Если таких программ более одной, то запишите любую из них.

Ответ:

B2 Определите значение переменной **c** после выполнения следующего фрагмента программы (записанного ниже на разных языках программирования). Ответ запишите в виде целого числа.

Бейсик	<pre>a = 40 b = 6 a = a * 3 / b IF a < b THEN c = 2*a - 10 * b ELSE c = 2*a + 10 * b END IF</pre>
--------	--

Паскаль	<pre>a := 40; b := 6; a := a * 3 / b; if a < b then c := 2*a - 10 * b else c := 2*a + 10 * b;</pre>
---------	--

Си	<pre>a = 40; b = 6; a = a * 3 / b; if (a < b) c = 2*a - 10 * b; else c = 2*a + 10 * b;</pre>
----	---

Алгоритмический	<pre>a := 40 b := 6 a := a * 3 / b если a < b то c := 2*a - 10 * b иначе c := 2*a + 10 * b все</pre>
-----------------	---

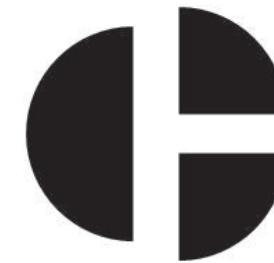
Ответ:

B3 Дан фрагмент электронной таблицы.

	A	B	C
1	2		=A1*2
2	= (B1-A1)/2	=B1-C1	=B2+A1

Какое число должно быть записано в ячейке B1, 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 двух 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

```
DIM N, S AS INTEGER
N = 0
S = 0
WHILE S <= 365
    S = S + 33
    N = N + 5
WEND
PRINT N
```

Паскаль

```
var n, s: integer;
begin
    n := 0;
    s := 0;
    while s <= 365 do
    begin
        s := s + 33;
        n := n + 5
    end;
    write(n)
end.
```

Си

```
#include<stdio.h>
void main()
{
    int n, s;
    n = 0;
    s = 0;
    while (s <= 365)
    {
        s = s + 33;
        n = n + 5;
    }
    printf("%d", n);
}
```

Алгоритмический

```
алг
нач
    цел n, s
    n := 0
    s := 0
    цц пока s <= 365
        s := s + 33
        n := n + 5
    кц
    вывод n
кон
```

Ответ:

B6 Алгоритм вычисления значения функции $F(n)$, где n – натуральное число, задан следующими соотношениями:

$$\begin{aligned} F(1) &= 1; \quad F(2)=1; \\ F(n) &= F(n-2) * n \text{ при } n > 2. \end{aligned}$$

Чему равно значение функции $F(8)$? В ответе запишите только натуральное число.

Ответ:

B7 Десятичное число 57 в некоторой системе счисления записывается как 212. Определите основание системы счисления.

Ответ:

B8 Ниже на 4 языках программирования 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 3, а потом 5.

Бейсик

```
DIM X, A, B AS INTEGER
INPUT X
A=0: B=0
WHILE X > 0
    A = A+1
    B = B +(X MOD 10)
    X = X \ 10
WEND
PRINT A
PRINT B
```

Паскаль

```

var x, a, b: integer;
begin
  readln(x);
  a:=0; b:=0;
  while x>0 do
  begin
    a:=a + 1;
    b:=b + (x mod 10);
    x:=x div 10;
  end;
  writeln(a); write(b);
end.

```

Си

```

#include<stdio.h>
void main()
{
  int x, a, b;
  scanf("%d", &x);
  a=0; b=0;
  while (x>0) {
    a = a+1;
    b = b +(x%10);
    x = x/10;
  }
  printf("%d\n%d", a, b);
}

```

Алгоритмический

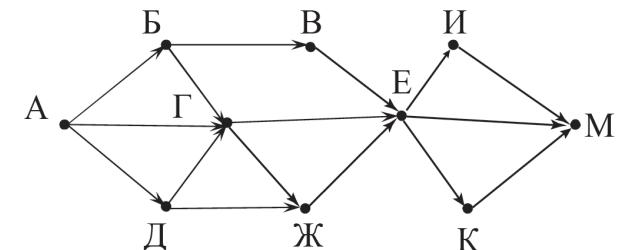
```

алг
нач
  цел x, a, b
  ввод x
  a:=0; b:=0
  нц пока x>0
    a:=a+1
    b:=b+mod(x,10)
    x:=div(x,10)
  кц
  вывод a, b
кон

```

Ответ: **B9**

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

**Ответ:** **B10**

Документ объёмом 8 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 25% от исходного,
- время, требуемое на сжатие документа – 10 секунд, на распаковку – 2 секунды?

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

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

Слова «секунд», «сек.», «с.» к ответу добавлять не нужно.

Ответ:

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

IP-адрес узла: 224.23.251.133

Маска: 255.255.240.0

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

A	B	C	D	E	F	G	H
255	240	252	224	133	23	8	0

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:

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)
Rишиль & (Amos Portos)	320
Rишиль & Amos	220
Rишиль & Amos & Portos	50

Какое количество страниц (в тысячах) будет найдено по запросу

Rишиль & Portos

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 2,
2. умножь на 2.

Первая из них увеличивает на 2 число на экране, вторая удваивает его. Программа для Удвоителя – это последовательность команд. Сколько есть программ, которые число 2 преобразуют в число 40?

Ответ:

B14 Определите, какое число будет напечатано в результате выполнения следующего алгоритма (для Вашего удобства алгоритм представлен на четырёх языках):

Бейсик	<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) < R THEN M = T R = F(T) END IF NEXT T PRINT R FUNCTION F(x) F = 11*(9-x)*(9-x)+131 END FUNCTION </pre>
--------	---

Паскаль	<pre> var a,b,t,M,R :integer; Function F(x: integer):integer; begin F := 11*(9-x)*(9-x)+131; end; BEGIN a := -20; b := 20; M := a; R := F(a); for t := a to b do begin if (F(t) < R) then begin M := t; R := F(t); end; end; write(R); END. </pre>
---------	---

Си

```
#include<stdio.h>
int F(int x)
{
    return 11*(9-x)*(9-x)+131;
}
void main()
{
    int a, b, t, M, R;
    a = -20; b = 20;
    M = a; R = F(a);
    for (t=a; t<=b; t++) {
        if (F(t) < R) {
            M = t; R = F(t);
        }
    }
    printf("%d", R);
}
```

Алгоритмический

```
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) < R
            то
                M := t; R := F(t)
        все
    кон
    вывод R
кон
алг цел F(цел x)
нач
    знач := 11*(9-x)*(9-x)+131
кон
```

Ответ:

B15 Сколько существует различных наборов значений логических переменных $x_1, x_2, x_3, x_4, x_5, y_1, y_2, y_3, y_4, y_5$, которые удовлетворяют всем перечисленным ниже условиям?

$$(x_1 \rightarrow x_2) \wedge (x_2 \rightarrow x_3) \wedge (x_3 \rightarrow x_4) \wedge (x_4 \rightarrow x_5) = 1$$

$$(y_1 \rightarrow y_2) \wedge (y_2 \rightarrow y_3) \wedge (y_3 \rightarrow y_4) \wedge (y_4 \rightarrow y_5) = 1$$

$$x_1 \rightarrow y_1 = 1$$

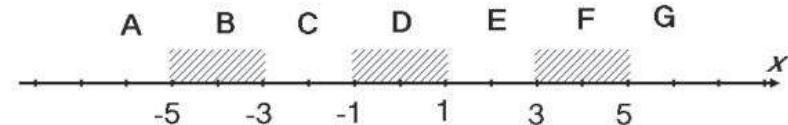
В ответе **не нужно** перечислять все различные наборы значений переменных $x_1, x_2, x_3, x_4, x_5, y_1, y_2, y_3, y_4, y_5$, при которых выполнена данная система равенств. В качестве ответа Вам нужно указать количество таких наборов.

Ответ:

Часть 3

Для записи ответов к заданиям этой части (C1–C4) используйте бланк ответов № 2. Запишите сначала номер задания (C1 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 (включая границы). Программист торопился и написал программу неправильно.



Бейсик	<pre>INPUT x IF x>=-5 OR x<=1 THEN IF x>=-3 AND x<=3 THEN IF x>=-1 AND x<=5 THEN PRINT "принадлежит" ELSE PRINT "не принадлежит" END IF END IF END IF</pre>
---------------	---

Паскаль

```
var x: real;
begin
    readln(x);
    if (x>=-5) or (x<=1) then
        if (x>=-3) and (x<=3) then
            if (x>=-1) and (x<=5) then
                write('принадлежит')
            else
                write('не принадлежит')
        end.
```

Си

```
#include <stdio.h>
void main(){
    float x;
    scanf("%f", &x);
    if (x>=-5 || x<=1)
        if (x>=-3 && x<=3)
            if (x>=-1 && x<=5)
                printf("принадлежит");
            else
                printf("не принадлежит");
}
```

Алгоритмический язык

```
алг
нач
    веш x
    ввод x
    если x>=-5 или x<=1 то
        если x>=-3 и x<=3 то
            если x>=-1 и x<=5 то
                вывод 'принадлежит'
            иначе
                вывод 'не принадлежит'
            все
        все
    все
кон
```

Последовательно выполните следующее.

- Пересвойте и заполните таблицу, которая показывает, как работает программа при аргументе, принадлежащем различным областям (A, B, C, D, E, F, G). Точки, лежащие на границах областей, отдельно не рассматривать.

Область	Условие 1 ($x \geq -5$ или $x \leq 1$)	Условие 2 ($x \geq -3$ и $x \leq 3$)	Условие 3 ($x \geq -1$ и $x \leq 5$)	Программа выведет	Область обрабатывается верно
A					
B					
C					
D					
E					
F					
G					

В столбцах условий укажите «да», если условие выполнится, «нет», если условие не выполнится, «—» (прочерк), если условие не будет проверяться, «не изв.», если программа ведёт 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».

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

C2 Дан массив, содержащий 70 целых чисел. Опишите на одном из языков программирования алгоритм, позволяющий найти и вывести наименьшее содержащееся в массиве положительное число, десятичная запись которого оканчивается цифрой 7. Гарантируется, что в массиве есть хотя бы один положительный элемент, десятичная запись которого оканчивается цифрой 7. Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

Паскаль

```
const
    N=70;
var
    a: array [1..N] of integer;
    i, j, m: integer;
begin
    for i:=1 to N do
        readln(a[i]);
    ...
end.
```

Бейсик

```
N=70
DIM A(N) AS INTEGER
DIM I, J, M AS INTEGER
FOR I = 1 TO N
    INPUT A(I)
NEXT I
...
END
```

Си

```
#include <stdio.h>
#define N 70
void main(){
    int a[N];
    int i, j, m;
    for (i=0; i<N; i++)
        scanf("%d", &a[i]);
    ...
}
```

Алгоритмический язык

```
алг
нач
    цел N=70
    целтаб a[1:N]
    цел i, j, m
    нц для i от 1 до N
        ввод a[i]
    кц
    ...
кон
```

В качестве ответа Вам необходимо привести фрагмент программы, 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.

C3

Два игрока, Паша и Вова, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Паша. За один ход игрок может добавить в кучу 1 камень или 10 камней. Например, имея кучу из 7 камней, за один ход можно получить кучу из 8 или 17 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче становится не менее 41. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 41 или больше камней.

В начальный момент в куче было S камней, $1 \leq S \leq 40$.

Будем говорить, что игрок имеет *выигрышную стратегию*, если он может выиграть при любых ходах противника. Описать стратегию игрока – значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника.

Выполните следующие задания. Во всех случаях обосновывайте свой ответ.

1. а) Укажите все такие значения числа S , при которых Паша может выиграть в один ход. Обоснуйте, что найдены все нужные значения S , и укажите выигрышающие ходы.

б) Укажите такое значение S , при котором Паша не может выиграть за один ход, но при любом ходе Паша Вова может выиграть своим первым ходом. Опишите выигрышную стратегию Вовы.

2. Укажите два значения S , при которых у Паши есть выигрышная стратегия, причём Паши не может выиграть за один ход, но может выиграть своим вторым ходом независимо от того, как будет ходить Вова. Для указанных значений S опишите выигрышную стратегию Паши.

3. Укажите значение S , при котором у Вовы есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Паши, однако у Вовы нет стратегии, которая позволит ему гарантированно выиграть первым ходом. Для указанного значения S опишите выигрышную стратегию Вовы. Постройте дерево всех партий, возможных при этой выигрышной стратегии Вовы (в виде рисунка или таблицы). На ребрах дерева указывайте, кто делает ход, в узлах – количество камней в куче.

C4 На плоскости дан набор точек с целочисленными координатами. Необходимо найти такой треугольник наибольшей площади с вершинами в этих точках, у которого нет общих точек с осью Oy , а одна из сторон лежит на оси Ox .

Напишите эффективную, в том числе по памяти, программу, которая будет решать эту задачу. Размер памяти, которую использует Ваша программа, не должен зависеть от количества точек.

Перед текстом программы кратко опишите используемый алгоритм решения задачи и укажите используемый язык программирования и его версию.

Описание входных данных

В первой строке вводится одно целое положительное число – количество точек N .

Каждая из следующих N строк содержит два целых числа – сначала координата x , затем координата y очередной точки. Числа разделены пробелом.

Описание выходных данных

Программа должна вывести одно число – максимальную площадь треугольника, удовлетворяющего условиям задачи. Если такого треугольника не существует, программа должна вывести ноль.

Пример входных данных:

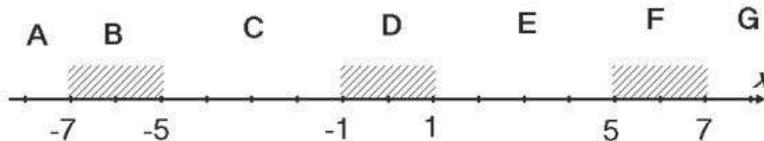
```
8
-10 0
2 0
0 4
3 3
7 0
5 5
4 0
9 -9
```

Пример выходных данных для приведённого выше примера входных данных:

22.5

Критерии оценивания заданий с развёрнутым ответом**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 (включая границы). Программист торопился и написал программу неправильно.



Бейсик	<pre> INPUT x IF x>=-7 OR x<=1 THEN IF x>=-5 AND x<=5 THEN IF x>=-1 AND x<=7 THEN PRINT "принадлежит" ELSE PRINT "не принадлежит" END IF END IF END IF </pre>
Паскаль	<pre> var x: real; begin readln(x); if (x>=-7) or (x<=1) then if (x>=-5) and (x<=5) then if (x>=-1) and (x<=7) then write('принадлежит') else write('не принадлежит') end. </pre>
Си	<pre> #include <stdio.h> void main(){ float x; scanf("%f", &x); if (x>=-7 x<=1) if (x>=-5 && x<=5) if (x>=-1 && x<=7) printf("принадлежит"); else printf("не принадлежит"); } </pre>

Алгоритмический язык	<pre> алг нач веш x ввод x если x>=-7 или x<=1 то если x>=-5 и x<=5 то если x>=-1 и x<=7 то вывод 'принадлежит' иначе вывод 'не принадлежит' все все кон </pre>
-----------------------------	---

Последовательно выполните следующее.

1. Пересуйте и заполните таблицу, которая показывает, как работает программа при аргументе, принадлежащем различным областям (A, B, C, D, E, F, G). Точки, лежащие на границах областей, отдельно не рассматривать.

Область	Условие 1 ($x \geq -7$ или $x \leq 1$)	Условие 2 ($x \geq -5$ и $x \leq 5$)	Условие 3 ($x \geq -1$ и $x \leq 7$)	Программа выведет	Область обрабатывается верно
A					
B					
C					
D					
E					
F					
G					

В столбцах условий укажите «да», если условие выполнится, «нет», если условие не выполнится, «—» (прочерк), если условие не будет проверяться, «не изв.», если программа ведёт 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.)

Элементы ответа:

Правильно заполненная таблица:

Область	Условие 1 ($x \geq -7$ или $x \leq 1$)	Условие 2 ($x \geq -5$ и $x \leq 5$)	Условие 3 ($x \geq -1$ и $x \leq 7$)	Программа выведет	Область обрабатывается верно
A	Да	Нет	—	—	Нет
B	Да	Нет	—	—	Нет
C	Да	Да	Нет	не принадлежит	Да
D	Да	Да	Да	принадлежит	Да
E	Да	Да	Да	принадлежит	Нет
F	Да	Нет	—	—	Нет
G	Да	Нет	—	—	Нет

Возможны различные способы доработки программы. Ниже приведены различные правильные решения, записанные на разных языках. Данные решения не привязаны к языкам: любую из содержащихся в них идей можно применить, используя любой язык программирования.

Пример на алгоритмическом языке

```
если -7<=x<=-5 или -1<=x<=1 или 5<=x<=7 то
    вывод 'принадлежит'
иначе
    вывод 'не принадлежит'
все
```

Пример на Паскале

```
if (abs(x)<=1) or (5<=abs(x)) and (abs(x)<=7) then
    write('принадлежит')
else
    write('не принадлежит');
```

Пример на Бейсике:

```
T = ABS(ABS(x)-3)
IF 2<=T AND T<=4 THEN
    PRINT "принадлежит"
ELSE
    PRINT "не принадлежит"
END IF
```

Возможны и другие способы доработки.

Обратите внимание! В задаче требуется выполнить три действия.

1. Заполнить таблицу.
2. Исправить ошибку в условном операторе.
3. Исправить ошибку, связанную с неправильным набором условий.
Баллы за данное задание начисляются как сумма баллов за верное выполнение каждого действия.

Рассмотрим отдельно каждое действие.

1. Действие по заполнению таблицы считается выполненным, если в таблице нет ошибок или ошибки присутствуют только в одной строке.
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 , при этом программа не стала работать хуже, чем раньше, т. е. для всех точек, для которых программа ранее выдавала верный ответ, доработанная программа также должна выдавать верный ответ.

3. Приведённые ограничения не описывают требуемые области. Первое условие верно для любых x , то есть не позволяет принять никаких решений, второе и третье не захватывают области В и F и не позволяют отделить друг от друга области D и E. Исправлением этой ошибки может быть разбиение области на части и использование дизъюнкции либо использование комбинации каскадных условий. Несколько примеров различных верных решений приведены выше.

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.

Указания по оцениванию	Баллы
Правильно выполнены оба пункта задания. Верно заполнена таблица, исправлены две ошибки. Программа для всех чисел x верно определяет принадлежность точки закрашенной области. В работе (во фрагментах программ) допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора решения.	3
1. Правильно выполнены два действия из трёх (исправлены обе ошибки, но в первом пункте 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). При написании операций сравнения допускается одно неправильное использование строгих/нестрогих неравенств (считается несущественной ошибкой, погрешностью записи). Например, вместо « $x \geq 5$ » используется « $x > 5$ ». 2. Или выполнены все три действия, но при этом в логическом выражении неверно учтены приоритеты логических операций (не расставлены или неправильно расставлены скобки в выражении).	2
Правильно выполнено только одно действие из трёх, то есть 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).	1
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).	0
<i>Максимальный балл</i>	3

C2 Дан массив, содержащий 70 целых чисел. Опишите на одном из языков программирования алгоритм, позволяющий найти и вывести наименьшее содержащееся в массиве положительное число, десятичная запись которого оканчивается цифрой 7. Гарантируется, что в массиве есть хотя бы один положительный элемент, десятичная запись которого оканчивается цифрой 7. Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

Паскаль	<pre> const N=70; var a: array [1..N] of integer; i, j, m: integer; begin for i:=1 to N do readln(a[i]); ... end. </pre>
---------	--

Бейсик	<pre> N=70 DIM A(N) AS INTEGER DIM I, J, M AS INTEGER FOR I = 1 TO N INPUT A(I) NEXT I ... END </pre>
--------	---

Си	<pre> #include <stdio.h> #define N 70 void main(){ int a[N]; int i, j, m; for (i=0; i<N; i++) scanf("%d", &a[i]); ... } </pre>
----	---

Алгоритмический язык	<u>алг</u> <u>нач</u> <u>цел</u> N=70 <u>целтаб</u> a[1:N] <u>цел</u> i, j, m <u>нц</u> для i от 1 до N <u>ввод</u> a[i] <u>кц</u> ... <u>кон</u>
----------------------	---

В качестве ответа Вам необходимо привести фрагмент программы, 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.

Содержание верного ответа и указания по оцениванию

(допускаются иные формулировки ответа, не искажающие его смысла)

В задаче необходимо найти минимальный среди элементов массива, удовлетворяющих заданному в условии ограничению. По сравнению со стандартной задачей поиска минимального среди всех элементов массива дополнительная сложность данной задачи заключается в том, что нельзя брать в качестве первого значения минимума первый элемент массива, так как этот элемент может не удовлетворять заданным ограничениям. Нельзя также принять в качестве первого значения большое число, заведомо превосходящее все возможные значения данных, так как в условии не указан диапазон возможных значений.

Ниже представлены несколько возможных способов решения задачи, проиллюстрированные фрагментами программ на разных языках. Способы решения не привязаны к языкам: любой из этих способов может быть реализован на любом допустимом языке.

Способ 1.

В качестве начального значения минимума принимается значение, заведомо не подходящее под заданные ограничения, например 0.

Пример программы на Паскале

```
m:=0;
for i:=1 to N do begin
  if (a[i]>0) and (a[i] mod 10=7) and ((m=0) or (a[i]<m))
    then m := a[i];
end;
writeln(m);
```

Способ 2.

Вместо проверки специального начального значения используется отдельная переменная, показывающая, был ли уже найден хотя бы один подходящий под ограничения элемент. Для этой переменной следовало бы использовать логический тип, но в условии разрешены только целые переменные, поэтому логическое значение моделируется с помощью целого.

Пример программы на Бейсике

```
M = 0: J = 0
FOR I = 1 TO N
  IF A(I)>0 AND A(i) MOD 10 = 7 AND (J = 0 OR A(I) < M) THEN
    M = A(I)
    J = 1
  END IF
NEXT I
PRINT M
```

Способ 3.

Сначала в массиве ищется первый элемент, соответствующий ограничениям. Затем в оставшейся части массива ищется подходящий наименьший элемент. Этот способ приводит к более длинной (требуется использовать два цикла), но не менее эффективной программе.

Пример программы на алгоритмическом языке

```
i:=1
нц пока не (a[i]>0 и mod(a[i],10)=7)
  i := i+1
кц
m := a[i]
нц для i от i+1 до N
  если a[i]>0 и mod(a[i],10)=7 и a[i]<m
    то m:=a[i]
  все
кц
вывод m
```

Указания по оцениванию	Баллы
<p>Предложен правильный алгоритм, выдающий верное значение.</p> <p>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.</p>	2
<p>В любом варианте решения может присутствовать не более одной ошибки из числа следующих:</p> <ol style="list-style-type: none"> 1) Не инициализируется или неверно инициализируется переменная m. В частности, нельзя инициализировать эту переменную первым элементом массива. Нельзя также инициализировать её каким-то очень большим значением (например, maxInt в Паскале) 2) Неверно проверяется положительность элементов массива 3) Неверно определяется последняя цифра десятичной записи числа 4) В сложном логическом условии простые проверки верны, но условие в целом построено неверно (например, перепутаны операции И и ИЛИ, неверно расставлены скобки в логическом выражении). 5) Вместо значения элемента проверяется его индекс. 6) Отсутствует вывод ответа. 7) Используется переменная, не объявленная в разделе описания переменных. 8) Не указано или неверно указано условие завершения цикла. 9) Индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно. 	1
Ошибок, перечисленных в п. 1–9, две или больше, или алгоритм сформулирован неверно.	0
Максимальный балл	2

C3

Два игрока, Паша и Вова, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Паша. За один ход игрок может добавить в кучу 1 камень или 10 камней. Например, имея кучу из 7 камней, за один ход можно получить кучу из 8 или 17 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче становится не менее 31. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 31 или больше камней.

В начальный момент в куче было S камней, $1 \leq S \leq 30$.

Будем говорить, что игрок имеет *выигрышную стратегию*, если он может выиграть при любых ходах противника. Описать стратегию игрока – значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника.

Выполните следующие задания. Во всех случаях обосновывайте свой ответ.

1. а) Укажите все такие значения числа S , при которых Паша может выиграть в один ход. Обоснуйте, что найдены все нужные значения S , и укажите выигрышающие ходы.

б) Укажите такое значение S , при котором Паша не может выиграть за один ход, но при любом ходе Паша Вова может выиграть своим первым ходом. Опишите выигрышную стратегию Вовы.

2. Укажите два значения S , при которых у Паши есть выигрышная стратегия, причём Паша не может выиграть за один ход, но может выиграть своим вторым ходом независимо от того, как будет ходить Вова. Для указанных значений S опишите выигрышную стратегию Паши.

3. Укажите значение S , при котором у Вовы есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Паши, однако у Вовы нет стратегии, которая позволит ему гарантированно выиграть первым ходом. Для указанного значения S опишите выигрышную стратегию Вовы. Постройте дерево всех партий, возможных при этой выигрышной стратегии Вовы (в виде рисунка или таблицы). На ребрах дерева указывайте, кто делает ход, в узлах – количество камней в куче.

Содержание верного ответа и указания по оцениванию

(допускаются иные формулировки ответа, не искажающие его смысла)

1. а) Паша может выиграть первым ходом, если $S=21, \dots, 30$. Во всех случаях можно добавить в кучу 10 камней. При меньших значениях S за один ход нельзя получить кучу, в которой больше 30 камней.

б) Вова может выиграть первым ходом (как бы ни играл Паша), если исходно в куче будет $S=20$ камней. Тогда после первого хода Паши в куче будет 21 камень или 30 камней. В обоих случаях Вова может добавить в кучу 10 камней и выиграть первым ходом.

2. Возможные значения S : 10 и 19. В этих случаях Паша, очевидно, не может выиграть первым ходом. Однако он может получить кучу из 20 камней. Эта позиция разобрана в п. 1б. В ней игрок, который будет ходить (теперь это Вова), выиграть не может, а его противник (то есть Паша) следующим ходом выиграет.

3. Возможное значение S : 18. После первого хода Паши в куче будет 19 или 28 камней. Если в куче станет 28 камней, Вова добавит в кучу 10 камней и выиграет первым ходом. Ситуация, когда в куче 19 камней, разобрана в п. 2. В этой ситуации игрок, который будет ходить (теперь это Вова), выигрывает своим вторым ходом.

В таблице изображено дерево возможных партий при описанной стратегии Вовы. Заключительные позиции (в них выигрывает Вова) подчёркнуты. На рисунке это же дерево изображено в графическом виде (оба способа изображения дерева допустимы).

Исходная позиция	Положения после очередных ходов			
	1-й ход Паши (разобраны все ходы)	1-й ход Вовы (только ход по стратегии)	2-й ход Паши (разобраны все ходы)	2-й ход Вовы (только ход по стратегии)
18	$18+1 = 19$	$19+1 = 20$	$20+1 = 21$	<u>$21+10 = 31$</u>
			$20+10 = 30$	<u>$30+10 = 40$</u>
	$18+10 = 28$	<u>$28+10 = 38$</u>		

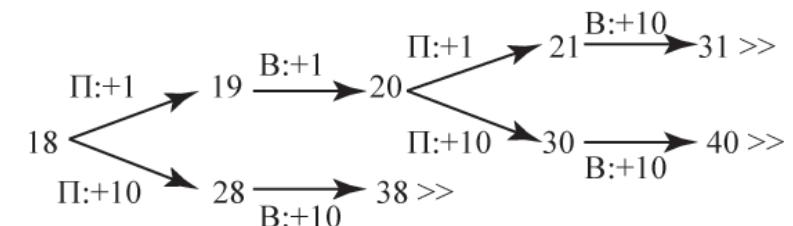


Рис. 1. Дерево всех партий, возможных при Вовиной стратегии.
Знаком >> обозначены позиции, в которых партия заканчивается

В задаче от ученика требуется выполнить 3 задания. Их трудность возрастает. Количество баллов в целом соответствует количеству выполненных заданий (подробнее см. ниже).

Ошибка в решении, не искажающая основного замысла, например арифметическая ошибка при вычислении количества камней в заключительной позиции при оценке решения не учитывается.

Первое задание считается выполненным, если выполнены оба пункта а) и б). Пункт а) считается выполненным, если правильно указаны все позиции, в которых Паша выигрывает первым ходом, и указано, каким должен быть первый ход. Пункт б) считается выполненным, если правильно указана позиция, в которой Вова выигрывает первым ходом, и описана стратегия Вовы, т. е. показано, как Вова может получить кучу, в которой содержится нужное количество камней, при любом ходе Паши.

Пункт а) первого задания считается выполненным частично, если правильно указаны все позиции, в которых Паша выигрывает первым ходом. Пункт б) первого задания считается выполненным частично, если правильно указана позиция, в которой Вова выигрывает первым ходом и явно сказано, что при любом ходе Паши Вова может получить кучу, которая содержит нужное для выигрыша количество камней.

Второе задание выполнено, если правильно указаны обе позиции, выигрышные для Паши, и описана соответствующая стратегия Паши – так, как это написано в примере решения, или другим способом, например с помощью дерева всех возможных партий.

Третье задание выполнено, если правильно указана позиция, выигрышная для Вовы, и построено дерево всех партий, возможных при Вовиной стратегии. Должно быть явно сказано, что в этом дереве в каждой позиции, где должен ходить Паша, разобраны все возможные ходы, а для позиций, где должен ходить Вова, – только ход, соответствующий стратегии, которую выбрал Вова.

Во всех случаях стратегии могут быть описаны так, как это сделано в примере решения, или другим способом.

Указания по оцениванию	Баллы
Выполнены второе и третье задания. Первое задание выполнено полностью или частично. Здесь и далее допускаются арифметические ошибки, которые не искажают сути решения и не приводят к неправильному ответу (см. выше).	3
Не выполнены условия, позволяющие поставить 3 балла, и выполнено одно из следующих условий. 1. Задание 3 выполнено. 2. Первое и второе задания выполнены. 3. Первое задание выполнено полностью или частично; для заданий 2 и 3 указаны правильные значения S .	2
Не выполнены условия, позволяющие поставить 3 или 2 балла, и выполнено одно из следующих условий. 1. Первое задание выполнено. 2. Во втором задании правильно указано одно из двух возможных значений S , и для этого значения указана и обоснована выигрышная стратегия Паши. 3. Первое задание выполнено частично, и для одного из остальных заданий правильно указано значение S . 4. Для второго и третьего задания правильно указаны значения S .	1
Не выполнено ни одно из условий, позволяющих поставить 3, 2 или 1 балл.	0
<i>Максимальный балл</i>	3

C4 На плоскости дан набор точек с целочисленными координатами. Необходимо найти такой треугольник наибольшей площади с вершинами в этих точках, у которого нет общих точек с осью Oy , а одна из сторон лежит на оси Ox .

Напишите эффективную, в том числе по памяти, программу, которая будет решать эту задачу. Размер памяти, которую использует Ваша программа, не должен зависеть от количества точек.

Перед текстом программы кратко опишите используемый алгоритм решения задачи и укажите используемый язык программирования и его версию.

Описание входных данных

В первой строке вводится одно целое положительное число – количество точек N .

Каждая из следующих N строк содержит два целых числа – сначала координата x , затем координата y очередной точки. Числа разделены пробелом.

Описание выходных данных

Программа должна вывести одно число – максимальную площадь треугольника, удовлетворяющего условиям задачи. Если такого треугольника не существует, программа должна вывести ноль.

Пример входных данных:

```
8
-10 0
2 0
0 4
3 3
7 0
5 5
4 0
9 -9
```

Пример выходных данных для приведённого выше примера входных данных:
22.5

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

Треугольник не имеет общих точек с осью Оу (осью ординат), если абсциссы всех его вершин имеют один и тот же знак, т. е. нужно отдельно рассматривать точки с положительными и отрицательными абсциссами. Для каждой из этих групп треугольник АВС, имеющий максимальную площадь, – это треугольник, у которого вершины А и В лежат на оси абсцисс по одну сторону от начала координат, причем у одной из этих точек абсцисса имеет наибольшую абсолютную величину, а у другой – наименьшую. Третья вершина С – это вершина, имеющая наибольшую по абсолютной величине ординату среди точек, лежащих по ту же сторону от оси ординат, что и точки А, В. Из двух таких «максимальных» треугольников (один лежит по одну сторону от оси ординат, другой – по другую) нужно выбрать тот, который имеет большую площадь.

Программа читает исходные данные, не запоминая все точки в массиве.

После обработки очередной точки программа хранит значения следующих величин:

- максимальное значение Pos_xmax абсциссы для всех прочитанных точек, лежащих на оси Ox правее начала координат;
- минимальное значение Pos_xmin абсциссы для всех прочитанных точек, лежащих на оси Ox правее начала координат;
- максимальное значение Pos_ymax абсолютной величины ординаты для всех прочитанных точек, лежащих правее оси ординат;
- максимальное значение Neg_xmax абсциссы для всех прочитанных точек, лежащих на оси Ox левее начала координат;
- минимальное значение Neg_xmin абсциссы для всех прочитанных точек, лежащих на оси Ox левее начала координат;
- максимальное значение Neg_ymax абсолютной величины ординаты для всех прочитанных точек, лежащих левее оси ординат.

После того как все точки прочитаны, вычисляется максимум

$$S = \max\{(Pos_xmax - Pos_xmin)*Pos_ymax/2, (Neg_xmax - Neg_xmin)*Neg_ymax/2\}$$

(если оба значения не определены, полагаем $S=0$).

В качестве ответа выводится число S.

Пример правильной и эффективной программы на языке Паскаль

```
var
  n: integer;
  x, y: integer;

  Pos_xmin, Pos_xmax: integer;
  Pos_xsearch: boolean;
  Pos_ymax: integer;

  Neg_xmin, Neg_xmax: integer;
  Neg_xsearch: boolean;
  Neg_ymax: integer;
  © МИОО 2012 г
```

```
i: integer;
Pos_s, Neg_s: real;

begin
  Pos_xsearch := true;
  Pos_xmin := 0; Pos_xmax := 0;
  Pos_ymax := 0;
  Neg_xsearch := true;
  Neg_xmin := 0; Neg_xmax := 0;
  Neg_ymax := 0;

  readln(n);
  for i:=1 to n do begin
    readln(x,y);
    if x > 0 then begin
      if y=0 then begin
        if Pos_xsearch or (x<Pos_xmin) then Pos_xmin:=x;
        if Pos_xsearch or (x>Pos_xmax) then Pos_xmax:=x;
        Pos_xsearch:=false;
      end
      else begin
        if abs(y)>Pos_ymax then Pos_ymax:=abs(y);
      end;
    end;
    if x < 0 then begin
      if y=0 then begin
        if Neg_xsearch or (x<Neg_xmin) then Neg_xmin:=x;
        if Neg_xsearch or (x>Neg_xmax) then Neg_xmax:=x;
        Neg_xsearch:=false;
      end
      else begin
        if abs(y)>Neg_ymax then Neg_ymax:=abs(y);
      end;
    end;
  end;
  Pos_s := (Pos_xmax-Pos_xmin)*Pos_ymax/2;
  Neg_s := (Neg_xmax-Neg_xmin)*Neg_ymax/2;
  if Pos_s > Neg_s then writeln(Pos_s)
  else writeln(Neg_s);
end.
```

Пример правильной и эффективной программы на языке Бейсик

```

DIM n AS INTEGER
DIM x, y AS INTEGER
DIM Pos_xmin, Pos_xmax AS INTEGER
DIM Pos_xsearch AS INTEGER
DIM Pos_ymax AS INTEGER
DIM Neg_xmin, Neg_xmax AS INTEGER
DIM Neg_xsearch AS INTEGER
DIM Neg_ymax AS INTEGER
DIM i AS INTEGER
DIM Pos_s, Neg_s AS DOUBLE

Pos_xsearch = 1: Neg_xsearch = 1
Pos_ymax = 0: Neg_ymax = 0
Pos_xmin = 0: Pos_xmax = 0
Neg_xmin = 0: Neg_xmax = 0
INPUT n
FOR i = 1 TO n
    INPUT x, y
    IF x > 0 THEN
        IF y = 0 THEN
            IF Pos_xsearch = 1 OR x < Pos_xmin THEN Pos_xmin = x
            IF Pos_xsearch = 1 OR x > Pos_xmax THEN Pos_xmax = x
            Pos_xsearch = 0
        ELSE
            IF ABS(y) > Pos_ymax THEN Pos_ymax = ABS(y)
        END IF
    END IF
    IF x < 0 THEN
        IF y = 0 THEN
            IF Neg_xsearch = 1 OR x < Neg_xmin THEN Neg_xmin = x
            IF Neg_xsearch = 1 OR x > Neg_xmax THEN Neg_xmax = x
            Neg_xsearch = 0
        ELSE
            IF ABS(y) > Neg_ymax THEN Neg_ymax = ABS(y)
        END IF
    END IF
NEXT i
Pos_s = (Pos_xmax - Pos_xmin) * Pos_ymax / 2
Neg_s = (Neg_xmax - Neg_xmin) * Neg_ymax / 2
IF Pos_s > Neg_s THEN
    PRINT Pos_s
ELSE
    PRINT Neg_s
END IF

```

Пример правильной и эффективной программы на алгоритмическом языке

```

алг
нач
    цел  n
    цел  x, y
    цел  Pos_xmin=0, Pos_xmax=0
    лог  Pos_xsearch=да
    цел  Pos_ymax=0
    цел  Neg_xmin=0, Neg_xmax=0
    лог  Neg_xsearch=да
    цел  Neg_ymax=0
    веш  Pos_s, Neg_s
    ввод  n
    нц  n  раз
        ввод  x, y
        если  x > 0  то
            если  y=0
                то
                    если  Pos_xsearch  или  x<Pos_xmin  то  Pos_xmin:=x  все
                    если  Pos_xsearch  или  x>Pos_xmax  то  Pos_xmax:=x  все
                    Pos_xsearch:=нет
                иначе
                    если  iabs(y)>Pos_ymax  то  Pos_ymax:=iabs(y)  все
                    все
                все
                если  x < 0
                    если  y=0
                        то
                            если  Neg_xsearch  или  x<Neg_xmin  то  Neg_xmin:=x  все
                            если  Neg_xsearch  или  x>Neg_xmax  то  Neg_xmax:=x  все
                            Neg_xsearch:=нет
                        иначе
                            если  iabs(y)>Neg_ymax  то  Neg_ymax:=iabs(y)  все
                            все
                        все
                    кон
                    Pos_s:=(Pos_xmax-PoS_xmin)*Pos_ymax/2
                    Neg_s:=(Neg_xmax-Neg_xmin)*Neg_ymax/2
                    если  Pos_s > Neg_s
                        то  вывод  Pos_s
                    иначе  вывод  Neg_s
                    все
                кон

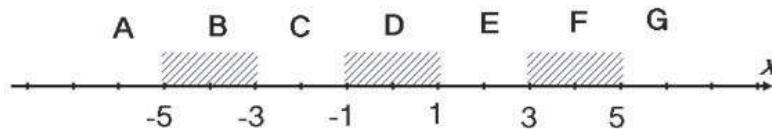
```

Указания по оцениванию	Баллы
Программа правильно работает для любых входных данных произвольного размера и находит ответ, не сохраняя входные данные в массиве. 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). Программа работает верно, но размер используемой памяти зависит от количества исходных данных. Например, входные данные (координаты точек) запоминаются в массиве или другой структуре данных, размер которой соответствует количеству точек. При этом обработка данных происходит с использованием эффективного алгоритма, аналогичного приведённым выше. Допускается одна из следующих ошибок. 1) Поиск минимума или максимума не учитывает, что первый подходящий элемент может оказаться на любом месте в исходных данных или вообще отсутствовать. 2) Перепутаны координаты x и y при поиске основания, ищутся максимальные и минимальные значения y при $x = 0$. 3) Перепутаны координаты x и y при поиске высоты: ищется максимальное значение x . 4) При поиске высоты ищется максимум значения координаты y , а не её модуля. 5) При поиске высоты запоминается не модуль, а значение y , при этом при вычислении площади модуль тоже не берётся, в результате может получиться отрицательная площадь. 6) Все вершины определены правильно, но площадь треугольника определена неверно, например, использована неверная формула. 7) Не учитывается, что вычисленное значение площади может быть нецелым. Например, значение площади присваивается переменной целого типа, при вычислении площади используется операция целочисленного деления (div в Паскале, деление целых величин без приведения типов в Си), при форматном выводе используется формат целого числа, или имеются другие подобные ошибки, приводящие к неверному результату при дробном ответе. 8) Вершины и площади двух треугольников определены верно, но из них выбирается не больший, а меньший. 9) Неверно обрабатывается ситуация, когда искомый треугольник отсутствует. Допускается наличие от одной до трёх синтаксических ошибок, описанных выше.	4
3	

Программа работает в целом верно, эффективно или нет. Возможны переборные решения, при которых все точки хранятся в массиве, из них выбираются подходящие треугольники, вычисляется и сравнивается их площадь. В реализации алгоритма допущено более 1 ошибки из числа перечисленных в предыдущем пункте или допущены другие ошибки, приводящие к неверной работе программы в отдельных случаях. Допускается наличие от одной до пяти синтаксических ошибок, описанных выше. Программа работает в отдельных частных случаях. Один балл также ставится, если программа написана неверно, но из описания алгоритма и общей структуры программы видно, что экзаменуемый в целом правильно представляет путь решения задачи.	2
Не выполнено ни одно из перечисленных выше условий.	0
<i>Максимальный балл</i>	4

Критерии оценивания заданий с развёрнутым ответом**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 (включая границы). Программист торопился и написал программу неправильно.



Бейсик	<pre> INPUT x IF x>=-5 OR x<=1 THEN IF x>=-3 AND x<=3 THEN IF x>=-1 AND x<=5 THEN PRINT "принадлежит" ELSE PRINT "не принадлежит" END IF END IF END IF END </pre>
--------	---

Паскаль	<pre> var x: real; begin readln(x); if (x>=-5) or (x<=1) then if (x>=-3) and (x<=3) then if (x>=-1) and (x<=5) then write('принадлежит') else write('не принадлежит') end. end. </pre>
---------	--

Си	<pre> #include <stdio.h> void main(){ float x; scanf("%f", &x); if (x>=-5 x<=1) if (x>=-3 && x<=3) if (x>=-1 && x<=5) printf("принадлежит"); else printf("не принадлежит"); } </pre>
----	---

Алгоритмический язык	<pre> алг нач веш x ввод x если x>=-5 или x<=1 то если x>=-3 и x<=3 то если x>=-1 и x<=5 то вывод 'принадлежит' иначе вывод 'не принадлежит' все все кон </pre>
-----------------------------	---

Последовательно выполните следующее.

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

Область	Условие 1 ($x \geq -5$ или $x \leq 1$)	Условие 2 ($x \geq -3$ и $x \leq 3$)	Условие 3 ($x \geq -1$ и $x \leq 5$)	Программа выведет	Область обрабатывается верно
A					
B					
C					
D					
E					
F					
G					

В столбцах условий укажите «да», если условие выполнится, «нет», если условие не выполнится, «—» (прочерк), если условие не будет проверяться, «не изв.», если программа ведёт 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.)

Элементы ответа:

Правильно заполненная таблица:

Область	Условие 1 $(x \geq -5 \text{ или } x \leq 1)$	Условие 2 $(x \geq -3 \text{ и } x \leq 3)$	Условие 3 $(x \geq -1 \text{ и } x \leq 5)$	Программа выведет	Область обрабатывается верно
A	Да	Нет	—	—	Нет
B	Да	Нет	—	—	Нет
C	Да	Да	Нет	не принадлежит	Да
D	Да	Да	Да	принадлежит	Да
E	Да	Да	Да	принадлежит	Нет
F	Да	Нет	—	—	Нет
G	Да	Нет	—	—	Нет

Возможны различные способы доработки программы. Ниже приведены различные правильные решения, записанные на разных языках. Данные решения не привязаны к языкам: любую из содержащихся в них идей можно применить, используя любой язык программирования.

Пример на алгоритмическом языке

```
если  $-5 \leq x \leq -3$  или  $-1 \leq x \leq 1$  или  $3 \leq x \leq 5$  то
    вывод 'принадлежит'
иначе
    вывод 'не принадлежит'
все
```

Пример на Паскале

```
if (abs(x) <= 1) or (3 <= abs(x)) and (abs(x) <= 5) then
    write('принадлежит')
else
    write('не принадлежит');
```

Пример на Бейсике

```
T = ABS(ABS(x) - 2)
IF 1 <= T AND T <= 3 THEN
    PRINT "принадлежит"
ELSE
    PRINT "не принадлежит"
END IF
```

Возможны и другие способы доработки.

Обратите внимание! В задаче требуется выполнить **три** действия.

1. Заполнить таблицу.
2. Исправить ошибку в условном операторе.
3. Исправить ошибку, связанную с неправильным набором условий.

Баллы за данное задание начисляются как сумма баллов за верное выполнение каждого действия. Рассмотрим отдельно каждое действие.

1. Действие по заполнению таблицы считается выполненным, если в таблице нет ошибок или ошибки присутствуют только в одной строке.

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 , при этом программа не стала работать хуже, чем раньше, т. е. для всех точек, для которых программа ранее выдавала верный ответ, доработанная программа также должна выдавать верный ответ.

3. Приведённые ограничения не описывают требуемые области. Первое условие верно для любых x , то есть не позволяет принять никаких решений, второе и третье не захватывают области В и F и не позволяют отделить друг от друга области D и E. Исправлением этой ошибки может быть разбиение области на части и использование дизъюнкции либо использование комбинации каскадных условий. Несколько примеров различных верных решений приведены выше.

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.

Указания по оцениванию	Баллы
Правильно выполнены оба пункта задания. Верно заполнена таблица, исправлены две ошибки. Программа для всех чисел x верно определяет принадлежность точки закрашенной области. В работе (во фрагментах программ) допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора решения.	3
1. Правильно выполнены два действия из трёх (исправлены обе ошибки, но в первом пункте 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). При написании операций сравнения допускается одно неправильное использование строгих/нестрогих неравенств (считается несущественной ошибкой, погрешностью записи). Например, вместо $\langle x \rangle \geq 5$ используется $\langle x \rangle > 5$. 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).	2
Правильно выполнено только одно действие из трёх, то есть 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 программы на точках границ областей (вместо нестрогих неравенств в решении были использованы строгие неравенства или наоборот).	1
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).	0
<i>Максимальный балл</i>	3

C2 Дан массив, содержащий 70 целых чисел. Опишите на одном из языков программирования алгоритм, позволяющий найти и вывести наименьшее содержащееся в массиве положительное число, десятичная запись которого не оканчивается цифрой 7. Гарантируется, что в массиве есть хотя бы один положительный элемент, десятичная запись которого не оканчивается цифрой 7.

Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

Паскаль	<pre> const N=70; var a: array [1..N] of integer; i, j, m: integer; begin for i:=1 to N do readln(a[i]); ... end. </pre>
---------	--

Бейсик	<pre> N=70 DIM A(N) AS INTEGER DIM I, J, M AS INTEGER FOR I = 1 TO N INPUT A(I) NEXT I ... END </pre>
--------	---

Си	<pre> #include <stdio.h> #define N 70 void main(){ int a[N]; int I, j, m; for (i=0; i<N; i++) scanf("%d", &a[i]); ... } </pre>
----	---

Алгоритмический язык	<u>алг</u> <u>нач</u> <u>цел</u> N=70 <u>целтаб</u> a[1:N] <u>цел</u> i, j, m <u>нц</u> <u>для</u> i <u>от</u> 1 <u>до</u> N <u>ввод</u> a[i] <u>кц</u> <u>...</u> <u>кон</u>
-----------------------------	--

В качестве ответа Вам необходимо привести фрагмент программы, 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.

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

В задаче необходимо найти минимальный среди элементов массива, удовлетворяющих заданному в условии ограничению. По сравнению со стандартной задачей поиска минимального среди всех элементов массива дополнительная сложность данной задачи заключается в том, что нельзя брать в качестве первого значения минимума первый элемент массива, так как этот элемент может не удовлетворять заданным ограничениям. Нельзя также принять в качестве первого значения большое число, заведомо превосходящее все возможные значения данных, так как в условии не указан диапазон возможных значений.

Ниже представлены несколько возможных способов решения задачи, проиллюстрированные фрагментами программ на разных языках. Способы решения не привязаны к языкам: любой из этих способов может быть реализован на любом допустимом языке.

Способ 1.

В качестве начального значения минимума принимается значение, заведомо не подходящее под заданные ограничения, например 0.

Пример программы на Паскале

```

m:=0;
for i:=1 to N do begin
  if (a[i]>0) and (a[i] mod 10<>7) and ((m=0) or (a[i]<m))
    then m := a[i];
end;
writeln(m);

```

Способ 2.

Вместо проверки специального начального значения используется отдельная переменная, показывающая, был ли уже найден хотя бы один подходящий под ограничения элемент. Для этой переменной следовало бы использовать логический тип, но в условии разрешены только целые переменные, поэтому логическое значение моделируется с помощью целого.

Пример программы на Бейсике

```
M = 0: J = 0
FOR I = 1 TO N
    IF A(I)>0 AND A(i) MOD 10 <> 7 AND (J=0 OR A(I)<M) THEN
        M = A(I)
        J = 1
    END IF
NEXT I
PRINT M
```

Способ 3.

Сначала в массиве ищется первый элемент, соответствующий ограничениям. Затем в оставшейся части массива ищется подходящий наименьший элемент. Этот способ приводит к более длинной (требуется использовать два цикла), но не менее эффективной программе.

Пример программы на алгоритмическом языке

```
i:=1
нц пока не (a[i]>0 и mod(a[i],10)<>7)
    i := i+1
кц
m := a[i]
нц для i от i+1 до N
    если a[i]>0 и mod(a[i],10)<>7 и a[i]<m
        то m:=a[i]
    все
кц
вывод m
```

Указания по оцениванию	Баллы
<p>Предложен правильный алгоритм, выдающий верное значение. Допускается запись алгоритма на другом языке, использующая аналогичные переменные. В случае если язык программирования использует типизированные переменные, описания переменных должны быть аналогичны описаниям переменных на естественном языке. Использование нетипизированных или необъявленных переменных возможно только в случае, если это допускается 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.</p>	2
<p>В любом варианте решения может присутствовать не более одной ошибки из числа следующих:</p> <ol style="list-style-type: none"> 1. Не инициализируется или неверно инициализируется переменная т. В частности, нельзя инициализировать эту переменную первым элементом массива. Нельзя также инициализировать её каким-то очень большим значением (например, maxInt в Паскале) 2. Неверно проверяется положительность элементов массива 3. Неверно определяется последняя цифра десятичной записи числа 4. В сложном логическом условии простые проверки верны, но условие в целом построено неверно (например, перепутаны операции И и ИЛИ, неверно расставлены скобки в логическом выражении). 5. Вместо значения элемента проверяется его индекс. 6. Отсутствует вывод ответа. 7. Используется переменная, не объявленная в разделе описания переменных. 8. Не указано или неверно указано условие завершения цикла. 9. Индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно. 	1
Ошибок, перечисленных в п. 1–9, две или больше, или алгоритм сформулирован неверно.	0
<i>Максимальный балл</i>	2

- C3** Два игрока, Паша и Вова, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Паша. За один ход игрок может добавить в кучу 1 камень или 10 камней. Например, имея кучу из 7 камней, за один ход можно получить кучу из 8 или 17 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней. Игра завершается в тот момент, когда количество камней в куче становится не менее 41. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 41 или больше камней.

В начальный момент в куче было S камней, $1 \leq S \leq 40$.

Будем говорить, что игрок имеет *выигрышную стратегию*, если он может выиграть при любых ходах противника. Описать стратегию игрока – значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника.

Выполните следующие задания. Во всех случаях обосновывайте свой ответ.

1. а) Укажите все такие значения числа S , при которых Паша может выиграть в один ход. Обоснуйте, что найдены все нужные значения S , и укажите выигрышающие ходы.

б) Укажите такое значение S , при котором Паша не может выиграть за один ход, но при любом ходе Паша Вова может выиграть своим первым ходом. Опишите выигрышную стратегию Вовы.

2. Укажите два значения S , при которых у Паши есть выигрышная стратегия, причём Паши не может выиграть за один ход, но может выиграть своим вторым ходом независимо от того, как будет ходить Вова. Для указанных значений S опишите выигрышную стратегию Паши.

3. Укажите значение S , при котором у Вовы есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Паши, однако у Вовы нет стратегии, которая позволит ему гарантированно выиграть первым ходом. Для указанного значения S опишите выигрышную стратегию Вовы. Постройте дерево всех партий, возможных при этой выигрышной стратегии Вовы (в виде рисунка или таблицы). На ребрах дерева указывайте, кто делает ход, в узлах – количество камней в куче.

Содержание верного ответа и указания по оцениванию

(допускаются иные формулировки ответа, не искажающие его смысла)

1. а) Паши может выиграть первым ходом, если $S = 31, \dots, 40$. Во всех случаях можно добавить в кучу 10 камней. При меньших значениях S за один ход нельзя получить кучу, в которой больше 40 камней.
- б) Вова может выиграть первым ходом (как бы ни играл Паши), если исходно в куче будет $S=30$ камней. Тогда после первого хода Паши в куче будет 31 камень или 40 камней. В обоих случаях Вова может добавить в кучу 10 камней и выиграть первым ходом.
2. Возможные значения S : 20 и 29. В этих случаях Паши, очевидно, не может выиграть первым ходом. Однако он может получить кучу из 30 камней. Эта позиция разобрана в п. 1б. В ней игрок, который будет ходить (теперь это Вова), выиграть не может, а его противник (то есть Паши) следующим ходом выиграет.

3. Возможное значение S : 28. После первого хода Паши в куче будет 29 или 38 камней. Если в куче станет 38 камней, Вова добавит в кучу 10 камней и выиграет первым ходом. Ситуация, когда в куче 29 камней, разобрана в п. 2. В этой ситуации игрок, который будет ходить (теперь это Вова), выигрывает своим вторым ходом.

В таблице изображено дерево возможных партий при описанной стратегии Вовы. Заключительные позиции (в них выигрывает Вова) подчёркнуты. На рисунке это же дерево изображено в графическом виде (оба способа изображения дерева допустимы).

Положения после очередных ходов				
Исходная позиция	1-й ход Паши (разобраны все ходы)	1-й ход Вовы (только ход по стратегии)	2-й ход Паши (разобраны все ходы)	2-й ход Вовы (только ход по стратегии)
28	$28+1=29$	$29+1=30$	$30+1=31$	$\underline{31+10=41}$
			$30+10=40$	$\underline{40+10=50}$
	$28+10=38$	$38+10=48$		

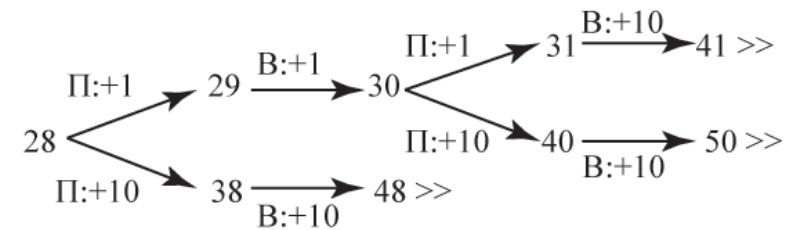


Рис. 1. Дерево всех партий, возможных при Вовиной стратегии. Знаком >> обозначены позиции, в которых партия заканчивается

В задаче от ученика требуется выполнить 3 задания. Их трудность возрастает. Количество баллов в целом соответствует количеству выполненных заданий (подробнее см. ниже).

Ошибка в решении, не искажающая основного замысла, например арифметическая ошибка при вычислении количества камней в заключительной позиции, при оценке решения не учитывается.

Первое задание считается выполненным, если выполнены оба пункта а) и б). Пункт а) считается выполненным, если правильно указаны все позиции, в которых Паши выигрывает первым ходом, и указано, каким должен быть первый ход. Пункт б) считается выполненным, если правильно указана позиция, в которой Вова выигрывает первым ходом, и описана стратегия Вовы, т. е. показано, как Вова может получить кучу, в которой содержится нужное количество камней, при любом ходе Паши.

Пункт а) первого задания считается выполненным частично, если правильно указаны все позиции, в которых Паши выигрывает первым ходом. Пункт б) первого задания считается выполненным частично, если правильно указана позиция, в которой Вова выигрывает первым ходом, и явно сказано, что при любом ходе Паши Вова может получить кучу, которая содержит нужное для выигрыша количество камней.

Второе задание выполнено, если правильно указаны обе позиции, выигрышные для Паши, и описана соответствующая стратегия Паши – так, как это написано в примере решения, или другим способом, например с помощью дерева всех возможных партий. Третье задание выполнено, если правильно указана позиция, выигрышная для Вовы, и построено дерево всех партий, возможных при Вовиной стратегии. Должно быть явно сказано, что в этом дереве в каждой позиции, где должен ходить Паша, разобраны все возможные ходы, а для позиций, где должен ходить Вова, – только ход, соответствующий стратегии, которую выбрал Вова. Во всех случаях стратегии могут быть описаны так, как это сделано в примере решения, или другим способом.

Указания по оцениванию	Баллы
Выполнены второе и третье задания. Первое задание выполнено полностью или частично. Здесь и далее допускаются арифметические ошибки, которые не искажают сути решения и не приводят к неправильному ответу (см. выше).	3
Не выполнены условия, позволяющие поставить 3 балла, и выполнено одно из следующих условий. 1. Задание 3 выполнено. 2. Первое и второе задания выполнены. 3. Первое задание выполнено полностью или частично; для заданий 2 и 3 указаны правильные значения S .	2
Не выполнены условия, позволяющие поставить 3 или 2 балла, и выполнено одно из следующих условий. 1. Первое задание выполнено. 2. Во втором задании правильно указано одно из двух возможных значений S , и для этого значения указана и обоснована выигрышная стратегия Паши. 3. Первое задание выполнено частично, и для одного из остальных заданий правильно указано значение S . 4. Для второго и третьего задания правильно указаны значения S .	1
Не выполнено ни одно из условий, позволяющих поставить 3, 2 или 1 балл.	0
<i>Максимальный балл</i>	3

C4 На плоскости дан набор точек с целочисленными координатами. Необходимо найти такой треугольник наибольшей площади с вершинами в этих точках, у которого нет общих точек с осью Ox , а одна из сторон лежит на оси Oy .

Напишите эффективную, в том числе по памяти, программу, которая будет решать эту задачу. Размер памяти, которую использует Ваша программа, не должен зависеть от количества точек.

Перед текстом программы кратко опишите используемый алгоритм решения задачи и укажите используемый язык программирования и его версию.

Описание входных данных

В первой строке вводится одно целое положительное число – количество точек N .

Каждая из следующих N строк содержит два целых числа – сначала координата x , затем координата y очередной точки. Числа разделены пробелом.

Описание выходных данных

Программа должна вывести одно число – максимальную площадь треугольника, удовлетворяющего условиям задачи. Если такого треугольника не существует, программа должна вывести ноль.

Пример входных данных:

```
8
0 -10
0 2
4 0
3 3
0 7
0 4
5 5
-9 9
```

Пример выходных данных для приведённого выше примера входных данных:
22.5

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

Треугольник не имеет общих точек с осью Ox (осью абсцисс), если ординаты всех его вершин имеют один и тот же знак, т. е. нужно отдельно рассматривать точки с положительными и отрицательными ординатами. Для каждой из этих групп треугольник ABC , имеющий максимальную площадь, – это треугольник, у которого вершины A и B лежат на оси ординат по одну сторону от начала координат, причем у одной из этих точек ордината имеет наибольшую абсолютную величину, а у другой – наименьшую. Третья вершина C – это вершина, имеющая наибольшую по абсолютной величине абсциссу среди точек, лежащих по ту же сторону от оси абсцисс, что и точки A , B . Из двух таких «максимальных» треугольников (один лежит по одну сторону от оси абсцисс, другой – по другую) нужно выбрать тот, который имеет большую площадь.

Программа читает исходные данные, не запоминая все точки в массиве.

После обработки очередной точки программа хранит значения следующих величин:

- максимальное значение Pos_ymax ординаты для всех прочитанных точек, лежащих на оси Oy выше начала координат;
- минимальное значение Pos_ymin ординаты для всех прочитанных точек, лежащих на оси Oy выше начала координат;
- максимальное значение Pos_xmax абсолютной величины абсциссы для всех прочитанных точек, лежащих выше оси абсцисс;
- максимальное значение Neg_ymax ординаты для всех прочитанных точек, лежащих на оси Oy ниже начала координат;
- минимальное значение Neg_ymin ординаты для всех прочитанных точек, лежащих на оси Oy ниже начала координат;
- максимальное значение Neg_xmax абсолютной величины абсциссы для всех прочитанных точек, лежащих ниже оси абсцисс.

После того, как все точки прочитаны, вычисляется максимум

$$S = \max\{(Pos_ymax - Pos_ymin) * Pos_xmax / 2, (Neg_ymax - Neg_ymin) * Neg_xmax / 2\}$$

(если оба значения не определены, полагаем $S=0$).

В качестве ответа выводится число S .

Пример правильной и эффективной программы на языке Паскаль

```
var
  n: integer;
  x, y: integer;
  Pos_ymin, Pos_ymax: integer;
  Pos_ysearch: boolean;
  Pos_xmax: integer;
  Neg_ymin, Neg_ymax: integer;
  Neg_ysearch: boolean;
  Neg_xmax: integer;
```

```
i: integer;
Pos_s, Neg_s: real;
begin
  Pos_ysearch := true; Neg_ysearch := true;
  Pos_ymin := 0; Pos_ymax := 0;
  Neg_ymin := 0; Neg_ymax := 0;
  Pos_xmax := 0; Neg_xmax := 0;

  readln(n);
  for i:=1 to n do begin
    readln(x, y);
    if y > 0 then begin
      if x=0 then begin
        if Pos_ysearch or (y<Pos_ymin) then Pos_ymin:=y;
        if Pos_ysearch or (y>Pos_ymax) then Pos_ymax:=y;
        Pos_ysearch:=false;
      end
      else begin
        if abs(x)>Pos_xmax then Pos_xmax:=abs(x);
      end;
    end;
    if y < 0 then begin
      if x=0 then begin
        if Neg_ysearch or (y<Neg_ymin) then Neg_ymin:=y;
        if Neg_ysearch or (y>Neg_ymax) then Neg_ymax:=y;
        Neg_ysearch:=false;
      end
      else begin
        if abs(x)>Neg_xmax then Neg_xmax:=abs(x);
      end;
    end;
  end;
  Pos_s := (Pos_ymax-Pos_ymin)*Pos_xmax/2;
  Neg_s := (Neg_ymax-Neg_ymin)*Neg_xmax/2;
  if Pos_s > Neg_s then writeln(Pos_s)
  else writeln(Neg_s);
end.
```

Пример правильной и эффективной программы на языке Бейсик

```

DIM n AS INTEGER
DIM x, y AS INTEGER
DIM Pos_ymin, Pos_ymax AS INTEGER
DIM Pos_ysearch AS INTEGER
DIM Pos_xmax AS INTEGER
DIM Neg_ymin, Neg_ymax AS INTEGER
DIM Neg_ysearch AS INTEGER
DIM Neg_xmax AS INTEGER
DIM i AS INTEGER
DIM Pos_s, Neg_s AS DOUBLE

Pos_ysearch = 1: Neg_ysearch = 1
Pos_ymin = 0: Pos_ymax = 0
Neg_ymin = 0: Neg_ymax = 0
Pos_xmax = 0: Neg_xmax = 0
INPUT n
FOR i = 1 TO n
    INPUT x, y
    IF y > 0 THEN
        IF x = 0 THEN
            IF Pos_ysearch = 1 OR y < Pos_ymin THEN Pos_ymin = y
            IF Pos_ysearch = 1 OR y > Pos_ymax THEN Pos_ymax = y
            Pos_ysearch = 0
        ELSE
            IF ABS(x) > Pos_xmax THEN Pos_xmax = ABS(x)
        END IF
    END IF
    IF y < 0 THEN
        IF x = 0 THEN
            IF Neg_ysearch = 1 OR y < Neg_ymin THEN Neg_ymin = y
            IF Neg_ysearch = 1 OR y > Neg_ymax THEN Neg_ymax = y
            Neg_ysearch = 0
        ELSE
            IF ABS(x) > Neg_xmax THEN Neg_xmax = ABS(x)
        END IF
    END IF
NEXT i
Pos_s = (Pos_ymax - Pos_ymin) * Pos_xmax / 2
Neg_s = (Neg_ymax - Neg_ymin) * Neg_xmax / 2
IF Pos_s > Neg_s THEN
    PRINT Pos_s
ELSE
    PRINT Neg_s
END IF

```

Пример правильной и эффективной программы на алгоритмическом языке

```

алг
нач
    цел n
    цел x, y
    цел Pos_ymin=0, Pos_ymax=0
    лог Pos_ysearch=да
    цел Pos_xmax=0
    цел Neg_ymin=0, Neg_ymax=0
    лог Neg_ysearch=да
    цел Neg_xmax=0
    веш Pos_s, Neg_s
    ввод n
    нц n раз
        ввод x, y
        если y > 0 то
            если x=0
                то
                    если Pos_ysearch или y<Pos_ymin то Pos_ymin:=y все
                    если Pos_ysearch или y>Pos_ymax то Pos_ymax:=y все
                    Pos_ysearch:=нет
                    иначе
                        если iabs(x)>Pos_xmax то Pos_xmax:=iabs(x) все
                    все
                    если y < 0 то
                        если x=0
                            то
                                если Neg_ysearch или y<Neg_ymin то Neg_ymin:=y все
                                если Neg_ysearch или y>Neg_ymax то Neg_ymax:=y все
                                Neg_ysearch:=нет
                                иначе
                                    если iabs(x)>Neg_xmax то Neg_xmax:=iabs(x) все
                                все
                            кон
                            Pos_s:=(Pos_ymax-Pos_ymin)*Pos_xmax/2
                            Neg_s:=(Neg_ymax-Neg_ymin)*Neg_xmax/2
                            если Pos_s > Neg_s
                                то вывод Pos_s
                            иначе вывод Neg_s
                        все
                    кон
                кон
            кон
        кон
    кон
кон

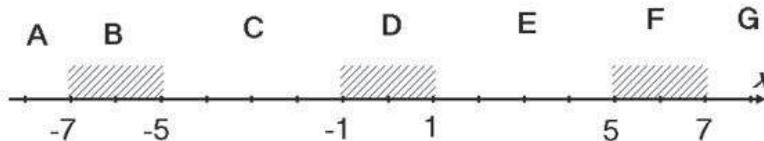
```

Указания по оцениванию	Баллы
<p>Программа правильно работает для любых входных данных произвольного размера и находит ответ, не сохраняя входные данные в массиве. Допускается наличие в тексте программы одной синтаксической ошибки: пропущен или неверно указан знак пунктуации, неверно написано или пропущено зарезервированное слово языка программирования, не описана или неверно описана переменная, применяется операция, недопустимая для соответствующего типа данных (если одна и та же ошибка встречается несколько раз, то это считается за одну ошибку).</p> <p>Программа работает верно, но размер используемой памяти зависит от количества исходных данных. Например, входные данные (координаты точек) запоминаются в массиве или другой структуре данных, размер которой соответствует количеству точек. При этом обработка данных происходит с использованием эффективного алгоритма, аналогичного приведённым выше.</p> <p>Допускается одна из следующих ошибок.</p> <ol style="list-style-type: none"> 1) Поиск минимума или максимума не учитывает, что первый подходящий элемент может оказаться на любом месте в исходных данных или вообще отсутствовать. 2) Перепутаны координаты y и x при поиске основания: ищутся максимальные и минимальные значения x при $y = 0$ 3) Перепутаны координаты y и x при поиске высоты: ищется максимальное значение y 4) При поиске высоты ищется максимум значения координаты x, а не её модуля 5) При поиске высоты запоминается не модуль, а значение x, при этом при вычислении площади модуль тоже не берётся, в результате может получиться отрицательная площадь 6) Все вершины определены правильно, но площадь треугольника определена неверно, например, использована неверная формула. 7) Не учитывается, что вычисленное значение площади может быть нецелым. Например, значение площади присваивается переменной целого типа, при вычислении площади используется операция целочисленного деления (div в Паскале, деление целых величин без приведения типов в Си), при форматном выводе используется формат целого числа или допущены другие подобные ошибки, приводящие к неверному результату при дробном ответе 8) Вершины и площади двух треугольников определены верно, но из них выбирается не больший, а меньший 9) Неверно обрабатывается ситуация, когда искомый треугольник отсутствует <p>Допускается наличие от одной до трёх синтаксических ошибок, описанных выше</p>	4
<p>Программа работает в целом верно, эффективно или нет. Возможны переборные решения, при которых все точки хранятся в массиве, из них выбираются подходящие треугольники, вычисляется и сравнивается их площадь</p> <p>В реализации алгоритма допущено более 1 ошибки из числа перечисленных в предыдущем пункте или допущены другие ошибки, приводящие к неверной работе программы в отдельных случаях.</p> <p>Допускается наличие от одной до пяти синтаксических ошибок, описанных выше.</p> <p>Программа работает в отдельных частных случаях.</p> <p>Один балл также ставится, если программа написана неверно, но из описания алгоритма и общей структуры программы видно, что экзаменуемый в целом правильно представляет путь решения задачи.</p> <p>Не выполнено ни одно из перечисленных выше условий.</p>	2
	Максимальный балл

<p>Программа работает в целом верно, эффективно или нет. Возможны переборные решения, при которых все точки хранятся в массиве, из них выбираются подходящие треугольники, вычисляется и сравнивается их площадь</p> <p>В реализации алгоритма допущено более 1 ошибки из числа перечисленных в предыдущем пункте или допущены другие ошибки, приводящие к неверной работе программы в отдельных случаях.</p> <p>Допускается наличие от одной до пяти синтаксических ошибок, описанных выше.</p> <p>Программа работает в отдельных частных случаях.</p> <p>Один балл также ставится, если программа написана неверно, но из описания алгоритма и общей структуры программы видно, что экзаменуемый в целом правильно представляет путь решения задачи.</p> <p>Не выполнено ни одно из перечисленных выше условий.</p>	1
	Максимальный балл

Критерии оценивания заданий с развёрнутым ответом**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 (включая границы). Программист торопился и написал программу неправильно.



Бейсик	<pre> INPUT x IF x>=-7 OR x<=1 THEN IF x>=-5 AND x<=5 THEN IF x>=-1 AND x<=7 THEN PRINT "принадлежит" ELSE PRINT "не принадлежит" END IF END IF END IF </pre>
Паскаль	<pre> var x: real; begin readln(x); if (x>=-7) or (x<=1) then if (x>=-5) and (x<=5) then if (x>=-1) and (x<=7) then write('принадлежит') else write('не принадлежит') end. </pre>
Си	<pre> #include <stdio.h> void main(){ float x; scanf("%f", &x); if (x>=-7 x<=1) if (x>=-5 && x<=5) if (x>=-1 && x<=7) printf("принадлежит"); else printf("не принадлежит"); } </pre>

Алгоритмический язык	<pre> алг нач веш x ввод x если x>=-7 или x<=1 то если x>=-5 и x<=5 то если x>=-1 и x<=7 то вывод 'принадлежит' иначе вывод 'не принадлежит' все все кон </pre>
-----------------------------	---

Последовательно выполните следующее.

1. Пересуйте и заполните таблицу, которая показывает, как работает программа при аргументе, принадлежащем различным областям (A, B, C, D, E, F, G). Точки, лежащие на границах областей, отдельно не рассматривать.

Область	Условие 1 ($x \geq -7$ или $x \leq 1$)	Условие 2 ($x \geq -5$ и $x \leq 5$)	Условие 3 ($x \geq -1$ и $x \leq 7$)	Программа выведет	Область обрабатывается верно
A					
B					
C					
D					
E					
F					
G					

В столбцах условий укажите «да», если условие выполнится, «нет», если условие не выполнится, «—» (прочерк), если условие не будет проверяться, «не изв.», если программа ведёт 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.)

Элементы ответа:

Правильно заполненная таблица:

Область	Условие 1 ($x \geq -7$ или $x \leq 1$)	Условие 2 ($x \geq -5$ и $x \leq 5$)	Условие 3 ($x \geq -1$ и $x \leq 7$)	Программа выведет	Область обрабатывается верно
A	Да	Нет	—	—	Нет
B	Да	Нет	—	—	Нет
C	Да	Да	Нет	не принадлежит	Да
D	Да	Да	Да	принадлежит	Да
E	Да	Да	Да	принадлежит	Нет
F	Да	Нет	—	—	Нет
G	Да	Нет	—	—	Нет

Возможны различные способы доработки программы. Ниже приведены различные правильные решения, записанные на разных языках. Данные решения не привязаны к языкам: любую из содержащихся в них идей можно применить, используя любой язык программирования.

Пример на алгоритмическом языке

```
если -7<=x<=-5 или -1<=x<=1 или 5<=x<=7 то
    вывод 'принадлежит'
иначе
    вывод 'не принадлежит'
все
```

Пример на Паскале

```
if (abs(x)<=1) or (5<=abs(x)) and (abs(x)<=7) then
    write('принадлежит')
else
    write('не принадлежит');
```

Пример на Бейсике:

```
T = ABS(ABS(x)-3)
IF 2<=T AND T<=4 THEN
    PRINT "принадлежит"
ELSE
    PRINT "не принадлежит"
END IF
```

Возможны и другие способы доработки.

Обратите внимание! В задаче требуется выполнить три действия.

1. Заполнить таблицу.
2. Исправить ошибку в условном операторе.
3. Исправить ошибку, связанную с неправильным набором условий.
Баллы за данное задание начисляются как сумма баллов за верное выполнение каждого действия.

Рассмотрим отдельно каждое действие.

1. Действие по заполнению таблицы считается выполненным, если в таблице нет ошибок или ошибки присутствуют только в одной строке.
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 , при этом программа не стала работать хуже, чем раньше, т. е. для всех точек, для которых программа ранее выдавала верный ответ, доработанная программа также должна выдавать верный ответ.

3. Приведённые ограничения не описывают требуемые области. Первое условие верно для любых x , то есть не позволяет принять никаких решений, второе и третье не захватывают области В и F и не позволяют отделить друг от друга области D и E. Исправлением этой ошибки может быть разбиение области на части и использование дизъюнкции либо использование комбинации каскадных условий. Несколько примеров различных верных решений приведены выше.

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.

Указания по оцениванию	Баллы
Правильно выполнены оба пункта задания. Верно заполнена таблица, исправлены две ошибки. Программа для всех чисел x верно определяет принадлежность точки закрашенной области. В работе (во фрагментах программ) допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора решения.	3
1. Правильно выполнены два действия из трёх (исправлены обе ошибки, но в первом пункте 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). При написании операций сравнения допускается одно неправильное использование строгих/нестрогих неравенств (считается несущественной ошибкой, погрешностью записи). Например, вместо « $x \geq 5$ » используется « $x > 5$ ». 2. Или выполнены все три действия, но при этом в логическом выражении неверно учтены приоритеты логических операций (не расставлены или неправильно расставлены скобки в выражении).	2
Правильно выполнено только одно действие из трёх, то есть 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).	1
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).	0
<i>Максимальный балл</i>	3

C2

Дан массив, содержащий 70 целых чисел. Опишите на одном из языков программирования алгоритм, позволяющий найти и вывести наименьшее содержащееся в массиве положительное число, десятичная запись которого не оканчивается цифрой 7. Гарантируется, что в массиве есть хотя бы один положительный элемент, десятичная запись которого не оканчивается цифрой 7.

Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

Паскаль	<pre>const N=70; var a: array [1..N] of integer; i, j, m: integer; begin for i:=1 to N do readln(a[i]); ... end.</pre>
---------	--

Бейсик	<pre>N=70 DIM A(N) AS INTEGER DIM I, J, M AS INTEGER FOR I = 1 TO N INPUT A(I) NEXT I ... END</pre>
--------	---

Си	<pre>#include <stdio.h> #define N 70 void main(){ int a[N]; int I, j, m; for (i=0; i<N; i++) scanf("%d", &a[i]); ... }</pre>
----	---

Алгоритмический язык	<pre> алг нач цел N=70 целтаб a[1:N] цел i, j, m нц для i от 1 до N ввод a[i] кц ... кон </pre>
-----------------------------	---

В качестве ответа Вам необходимо привести фрагмент программы, 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.

Содержание верного ответа и указания по оцениванию

(допускаются иные формулировки ответа, не искажающие его смысла)

В задаче необходимо найти минимальный среди элементов массива, удовлетворяющих заданному в условии ограничению. По сравнению со стандартной задачей поиска минимального среди всех элементов массива дополнительная сложность данной задачи заключается в том, что нельзя брать в качестве первого значения минимума первый элемент массива, так как этот элемент может не удовлетворять заданным ограничениям. Нельзя также принять в качестве первого значения большое число, заведомо превосходящее все возможные значения данных, так как в условии не указан диапазон возможных значений.

Ниже представлены несколько возможных способов решения задачи, проиллюстрированные фрагментами программ на разных языках. Способы решения не привязаны к языкам: любой из этих способов может быть реализован на любом допустимом языке.

Способ 1.

В качестве начального значения минимума принимается значение, заведомо не подходящее под заданные ограничения, например 0.

Пример программы на Паскале

```

m:=0;
for i:=1 to N do begin
    if (a[i]>0) and (a[i] mod 10<>7) and ((m=0) or (a[i]<m))
        then m := a[i];
end;
writeln(m);

```

Способ 2.

Вместо проверки специального начального значения используется отдельная переменная, показывающая, был ли уже найден хотя бы один подходящий под ограничения элемент. Для этой переменной следовало бы использовать логический тип, но в условии разрешены только целые переменные, поэтому логическое значение моделируется с помощью целого.

Пример программы на Бейсике

```

M = 0: J = 0
FOR I = 1 TO N
    IF A(I)>0 AND A(i) MOD 10 <> 7 AND (J=0 OR A(I)<M) THEN
        M = A(I)
        J = 1
    END IF
NEXT I
PRINT M

```

Способ 3.

Сначала в массиве ищется первый элемент, соответствующий ограничениям. Затем в оставшейся части массива ищется подходящий наименьший элемент. Этот способ приводит к более длинной (требуется использовать два цикла), но не менее эффективной программе.

Пример программы на алгоритмическом языке

```

i:=1
нц пока не (a[i]>0 и mod(a[i],10)<>7)
    i := i+1
кц
m := a[i]
нц для i от i+1 до N
    если a[i]>0 и mod(a[i],10)<>7 и a[i]<m
        то m:=a[i]
    все
кц
вывод m

```

Указания по оцениванию	Баллы
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: 1. Не инициализируется или неверно инициализируется переменная <i>t</i> . В частности, нельзя инициализировать эту переменную первым элементом массива. Нельзя также инициализировать её каким-то очень большим значением (например, <i>maxInt</i> в Паскале) 2. Неверно проверяется положительность элементов массива 3. Неверно определяется последняя цифра десятичной записи числа 4. В сложном логическом условии простые проверки верны, но условие в целом построено неверно (например, перепутаны операции И и ИЛИ, неверно расставлены скобки в логическом выражении). 5. Вместо значения элемента проверяется его индекс. 6. Отсутствует вывод ответа. 7. Используется переменная, не объявленная в разделе описания переменных. 8. Не указано или неверно указано условие завершения цикла. 9. Индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно. Ошибок, перечисленных в п. 1–9, две или больше, или алгоритм сформулирован неверно.	1
Максимальный балл	0
	2

C3 Два игрока, Паша и Вова, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Паша. За один ход игрок может добавить в кучу 1 камень или 10 камней. Например, имея кучу из 7 камней, за один ход можно получить кучу из 8 или 17 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче становится не менее 31. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 31 или больше камней.

В начальный момент в куче было S камней, $1 \leq S \leq 30$.

Будем говорить, что игрок имеет *выигрышную стратегию*, если он может выиграть при любых ходах противника. Описать стратегию игрока – значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника.

Выполните следующие задания. Во всех случаях обосновывайте свой ответ.

1. а) Укажите все такие значения числа S , при которых Паша может выиграть в один ход. Обоснуйте, что найдены все нужные значения S , и укажите выигрышающие ходы.

б) Укажите такое значение S , при котором Паша не может выиграть за один ход, но при любом ходе Паши Вова может выиграть своим первым ходом. Опишите выигрышную стратегию Вовы.

2. Укажите два значения S , при которых у Паши есть выигрышная стратегия, причём Паши не может выиграть за один ход, но может выиграть своим вторым ходом независимо от того, как будет ходить Вова. Для указанных значений S опишите выигрышную стратегию Паши.

3. Укажите значение S , при котором у Вовы есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Паши, однако у Вовы нет стратегии, которая позволит ему гарантированно выиграть первым ходом. Для указанного значения S опишите выигрышную стратегию Вовы. Постройте дерево всех партий, возможных при этой выигрышной стратегии Вовы (в виде рисунка или таблицы). На ребрах дерева указывайте, кто делает ход, в узлах – количество камней в куче.

Содержание верного ответа и указания по оцениванию

(допускаются иные формулировки ответа, не искажающие его смысла)

1. а) Паши может выиграть первым ходом, если $S = 21, \dots, 30$. Во всех случаях можно добавить в кучу 10 камней. При меньших значениях S за один ход нельзя получить кучу, в которой больше 30 камней.

б) Вова может выиграть первым ходом (как бы ни играл Паши), если исходно в куче будет $S=20$ камней. Тогда после первого хода Паши в куче будет 21 камень или 30 камней. В обоих случаях Вова может добавить в кучу 10 камней и выиграть первым ходом.

2. Возможные значения S : 10 и 19. В этих случаях Паша, очевидно, не может выиграть первым ходом. Однако он может получить кучу из 20 камней. Эта позиция разобрана в п. 1б. В ней игрок, который будет ходить (теперь это Вова), выиграть не может, а его противник (то есть Паша) следующим ходом выиграет.

3. Возможное значение S : 18. После первого хода Паши в куче будет 19 или 28 камней. Если в куче станет 28 камней, Вова добавит в кучу 10 камней и выиграет первым ходом. Ситуация, когда в куче 19 камней, разобрана в п. 2. В этой ситуации игрок, который будет ходить (теперь это Вова), выигрывает своим вторым ходом.

В таблице изображено дерево возможных партий при описанной стратегии Вовы. Заключительные позиции (в них выигрывает Вова) подчёркнуты. На рисунке это же дерево изображено в графическом виде (оба способа изображения дерева допустимы).

Исходная позиция	Положения после очередных ходов			
	1-й ход Паши (разобраны все ходы)	1-й ход Вовы (только ход по стратегии)	2-й ход Паши (разобраны все ходы)	2-й ход Вовы (только ход по стратегии)
18	$18+1=19$	$19+1=\underline{\underline{20}}$	$20+1=\underline{21}$ $20+10=\underline{30}$	$\underline{21}+10=\underline{31}$ $30+10=\underline{40}$
	$18+10=\underline{28}$	$28+10=\underline{38}$		

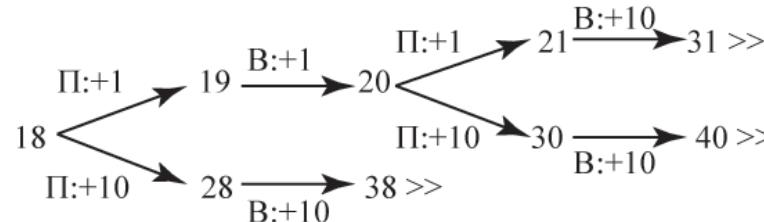


Рис. 1. Дерево всех партий, возможных при Вовиной стратегии. Знаком >> обозначены позиции, в которых партия заканчивается

В задаче от ученика требуется выполнить 3 задания. Их трудность возрастает. Количество баллов в целом соответствует количеству выполненных заданий (подробнее см. ниже).

Ошибка в решении, не искажающая основного замысла, например арифметическая ошибка при вычислении количества камней в заключительной позиции при оценке решения не учитывается.

Первое задание считается выполненным, если выполнены оба пункта а) и б). Пункт а) считается выполненным, если правильно указаны все позиции, в которых Паша выигрывает первым ходом, и указано, каким должен быть первый ход. Пункт б) считается выполненным, если правильно указана позиция, в которой Вова выигрывает первым ходом, и описана стратегия Вовы, т. е. показано, как Вова может получить кучу, в которой содержится нужное количество камней, при любом ходе Паши.

Пункт а) первого задания считается выполненным частично, если правильно указаны все позиции, в которых Паша выигрывает первым ходом. Пункт б) первого задания считается выполненным частично, если правильно указана позиция, в которой Вова выигрывает первым ходом и явно сказано, что при любом ходе Паши Вова может получить кучу, которая содержит нужное для выигрыша количество камней.

Второе задание выполнено, если правильно указаны обе позиции, выигрышные для Паши, и описана соответствующая стратегия Паши – так, как это написано в примере решения, или другим способом, например с помощью дерева всех возможных партий.

Третье задание выполнено, если правильно указана позиция, выигрышная для Вовы, и построено дерево всех партий, возможных при Вовиной стратегии. Должно быть явно сказано, что в этом дереве в каждой позиции, где должен ходить Паша, разобраны все возможные ходы, а для позиций, где должен ходить Вова, – только ход, соответствующий стратегии, которую выбрал Вова.

Во всех случаях стратегии могут быть описаны так, как это сделано в примере решения, или другим способом.

Указания по оцениванию	Баллы
Выполнены второе и третье задания. Первое задание выполнено полностью или частично. Здесь и далее допускаются арифметические ошибки, которые не искажают сути решения и не приводят к неправильному ответу (см. выше).	3
Не выполнены условия, позволяющие поставить 3 балла, и выполнено одно из следующих условий. 1. Задание 3 выполнено. 2. Первое и второе задания выполнены. 3. Первое задание выполнено полностью или частично; для заданий 2 и 3 указаны правильные значения S .	2
Не выполнены условия, позволяющие поставить 3 или 2 балла, и выполнено одно из следующих условий. 1. Первое задание выполнено. 2. Во втором задании правильно указано одно из двух возможных значений S , и для этого значения указана и обоснована выигрышная стратегия Паши. 3. Первое задание выполнено частично, и для одного из остальных заданий правильно указано значение S . 4. Для второго и третьего задания правильно указаны значения S .	1
Не выполнено ни одно из условий, позволяющих поставить 3, 2 или 1 балл.	0
<i>Максимальный балл</i>	3

C4

На плоскости дан набор точек с целочисленными координатами. Необходимо найти такой треугольник наибольшей площади с вершинами в этих точках, у которого нет общих точек с осью Ox , а одна из сторон лежит на оси Oy .

Напишите эффективную, в том числе по памяти, программу, которая будет решать эту задачу. Размер памяти, которую использует Ваша программа, не должен зависеть от количества точек.

Перед текстом программы кратко опишите используемый алгоритм решения задачи и укажите используемый язык программирования и его версию.

Описание входных данных

В первой строке вводится одно целое положительное число – количество точек N .

Каждая из следующих N строк содержит два целых числа – сначала координата x , затем координата y очередной точки. Числа разделены пробелом.

Описание выходных данных

Программа должна вывести одно число – максимальную площадь треугольника, удовлетворяющего условиям задачи. Если такого треугольника не существует, программа должна вывести ноль.

Пример входных данных:

```
8
0 -10
0 2
4 0
3 3
0 7
0 4
5 5
-9 9
```

Пример выходных данных для приведённого выше примера входных данных:
22.5

Содержание верного ответа и указания по оцениванию

(допускаются иные формулировки ответа, не исказжающие его смысла)

Треугольник не имеет общих точек с осью Ox (осью абсцисс), если ординаты всех его вершин имеют один и тот же знак, т. е. нужно отдельно рассматривать точки с положительными и отрицательными ординатами. Для каждой из этих групп треугольник ABC, имеющий максимальную площадь, – это треугольник, у которого вершины A и B лежат на оси ординат по одну сторону от начала координат, причем у одной из этих точек ордината имеет наибольшую абсолютную величину, а у другой – наименьшую. Третья вершина C – это вершина, имеющая наибольшую по абсолютной величине абсциссу среди точек, лежащих по ту же сторону от оси абсцисс, что и точки A, B. Из двух таких «максимальных» треугольников (один лежит по одну сторону от оси абсцисс, другой – по другую) нужно выбрать тот, который имеет большую площадь.

Программа читает исходные данные, не запоминая все точки в массиве.

После обработки очередной точки программа хранит значения следующих величин:

- максимальное значение Pos_ymax ординаты для всех прочитанных точек, лежащих на оси Oy выше начала координат;
- минимальное значение Pos_ymin ординаты для всех прочитанных точек, лежащих на оси Oy выше начала координат;
- максимальное значение Pos_xmax абсолютной величины абсциссы для всех прочитанных точек, лежащих выше оси абсцисс;
- максимальное значение Neg_ymax ординаты для всех прочитанных точек, лежащих на оси Oy ниже начала координат;
- минимальное значение Neg_ymin ординаты для всех прочитанных точек, лежащих на оси Oy ниже начала координат;
- максимальное значение Neg_xmax абсолютной величины абсциссы для всех прочитанных точек, лежащих ниже оси абсцисс.

После того, как все точки прочитаны, вычисляется максимум

$$S = \max\{(Pos_ymax - Pos_ymin)*Pos_xmax/2, (Neg_ymax - Neg_ymin)*Neg_xmax/2\}$$

(если оба значения не определены, полагаем $S=0$).

В качестве ответа выводится число S.

Пример правильной и эффективной программы на языке Паскаль

```
var
  n: integer;
  x, y: integer;
  Pos_ymin, Pos_ymax: integer;
  Pos_ysearch: boolean;
  Pos_xmax: integer;
  Neg_ymin, Neg_ymax: integer;
  Neg_ysearch: boolean;
  Neg_xmax: integer;
```

```

i: integer;
Pos_s, Neg_s: real;
begin
  Pos_ysearch := true; Neg_ysearch := true;
  Pos_ymin := 0; Pos_ymax := 0;
  Neg_ymin := 0; Neg_ymax := 0;
  Pos_xmax := 0; Neg_xmax := 0;

  readln(n);
  for i:=1 to n do begin
    readln(x, y);
    if y > 0 then begin
      if x=0 then begin
        if Pos_ysearch or (y<Pos_ymin) then Pos_ymin:=y;
        if Pos_ysearch or (y>Pos_ymax) then Pos_ymax:=y;
        Pos_ysearch:=false;
      end
      else begin
        if abs(x)>Pos_xmax then Pos_xmax:=abs(x);
      end;
    end;
    if y < 0 then begin
      if x=0 then begin
        if Neg_ysearch or (y<Neg_ymin) then Neg_ymin:=y;
        if Neg_ysearch or (y>Neg_ymax) then Neg_ymax:=y;
        Neg_ysearch:=false;
      end
      else begin
        if abs(x)>Neg_xmax then Neg_xmax:=abs(x);
      end;
    end;
    Pos_s := (Pos_ymax-Pos_ymin)*Pos_xmax/2;
    Neg_s := (Neg_ymax-Neg_ymin)*Neg_xmax/2;
    if Pos_s > Neg_s then writeln(Pos_s)
    else writeln(Neg_s);
  end.
end.

```

Пример правильной и эффективной программы на языке Бейсик

```

DIM n AS INTEGER
DIM x, y AS INTEGER
DIM Pos_ymin, Pos_ymax AS INTEGER
DIM Pos_ysearch AS INTEGER
DIM Pos_xmax AS INTEGER
DIM Neg_ymin, Neg_ymax AS INTEGER
DIM Neg_ysearch AS INTEGER
DIM Neg_xmax AS INTEGER
DIM i AS INTEGER
DIM Pos_s, Neg_s AS DOUBLE

Pos_ysearch = 1: Neg_ysearch = 1
Pos_ymin = 0: Pos_ymax = 0
Neg_ymin = 0: Neg_ymax = 0
Pos_xmax = 0: Neg_xmax = 0
INPUT n
FOR i = 1 TO n
  INPUT x, y
  IF y > 0 THEN
    IF x = 0 THEN
      IF Pos_ysearch = 1 OR y < Pos_ymin THEN Pos_ymin = y
      IF Pos_ysearch = 1 OR y > Pos_ymax THEN Pos_ymax = y
      Pos_ysearch = 0
    ELSE
      IF ABS(x) > Pos_xmax THEN Pos_xmax = ABS(x)
    END IF
  END IF
  IF y < 0 THEN
    IF x = 0 THEN
      IF Neg_ysearch = 1 OR y < Neg_ymin THEN Neg_ymin = y
      IF Neg_ysearch = 1 OR y > Neg_ymax THEN Neg_ymax = y
      Neg_ysearch = 0
    ELSE
      IF ABS(x) > Neg_xmax THEN Neg_xmax = ABS(x)
    END IF
  END IF
NEXT i
Pos_s = (Pos_ymax - Pos_ymin) * Pos_xmax / 2
Neg_s = (Neg_ymax - Neg_ymin) * Neg_xmax / 2
IF Pos_s > Neg_s THEN
  PRINT Pos_s
ELSE
  PRINT Neg_s
END IF

```

Пример правильной и эффективной программы на алгоритмическом языке

```

алг
нач
цел п
цел х, у
цел Pos_ymin=0, Pos_ymax=0
лог Pos_ysearch=да
цел Pos_xmax=0
цел Neg_ymin=0, Neg_ymax=0
лог Neg_ysearch=да
цел Neg_xmax=0
вещ Pos_s, Neg_s
ввод п
нц п раз
    ввод х, у
    если у > 0 то
        если x=0
            то
                если Pos_ysearch или y<Pos_ymin то Pos_ymin:=y все
                если Pos_ysearch или y>Pos_ymax то Pos_ymax:=y все
                Pos_ysearch:=нет
            иначе
                если iabs(x)>Pos_xmax то Pos_xmax:=iabs(x) все
            все
        все
        если у < 0 то
            если x=0
                то
                    если Neg_ysearch или y<Neg_ymin то Neg_ymin:=y все
                    если Neg_ysearch или y>Neg_ymax то Neg_ymax:=y все
                    Neg_ysearch:=нет
                иначе
                    если iabs(x)>Neg_xmax то Neg_xmax:=iabs(x) все
                все
            все
        кон
    кон
    Pos_s:=(Pos_ymax-Pos_ymin)*Pos_xmax/2
    Neg_s:=(Neg_ymax-Neg_ymin)*Neg_xmax/2
    если Pos_s > Neg_s
        то вывод Pos_s
    иначе вывод Neg_s
    все
кон

```

Указания по оцениванию	Баллы
<p>Программа правильно работает для любых входных данных произвольного размера и находит ответ, не сохраняя входные данные в массиве. Допускается наличие в тексте программы одной синтаксической ошибки: пропущен или неверно указан знак пунктуации, неверно написано или пропущено зарезервированное слово языка программирования, не описана или неверно описана переменная, применяется операция, недопустимая для соответствующего типа данных (если одна и та же ошибка встречается несколько раз, то это считается за одну ошибку).</p> <p>Программа работает верно, но размер используемой памяти зависит от количества исходных данных. Например, входные данные (координаты точек) запоминаются в массиве или другой структуре данных, размер которой соответствует количеству точек. При этом обработка данных происходит с использованием эффективного алгоритма, аналогичного приведённым выше.</p> <p>Допускается одна из следующих ошибок.</p> <ol style="list-style-type: none"> 1) Поиск минимума или максимума не учитывает, что первый подходящий элемент может оказаться на любом месте в исходных данных или вообще отсутствовать. 2) Перепутаны координаты y и x при поиске основания: ищутся максимальные и минимальные значения x при $y = 0$. 3) Перепутаны координаты y и x при поиске высоты: ищется максимальное значение y. 4) При поиске высоты ищется максимум значения координаты x, а не её модуля. 5) При поиске высоты запоминается не модуль, а значение x, при этом при вычислении площади модуль тоже не берётся, в результате может получиться отрицательная площадь. 6) Все вершины определены правильно, но площадь треугольника определена неверно, например, использована неверная формула. 7) Не учитывается, что вычисленное значение площади может быть нецелым. Например, значение площади присваивается переменной целого типа, при вычислении площади используется операция целочисленного деления (div в Паскале, деление целых величин без приведения типов в Си), при форматном выводе используется формат целого числа или допущены другие подобные ошибки, приводящие к неверному результату при дробном ответе. 8) Вершины и площади двух треугольников определены верно, но из них выбирается не больший, а меньший. 9) Неверно обрабатывается ситуация, когда искомый треугольник отсутствует. <p>Допускается наличие от одной до трёх синтаксических ошибок, описанных выше</p>	4
<p>Программа работает верно, но размер используемой памяти зависит от количества исходных данных. Например, входные данные (координаты точек) запоминаются в массиве или другой структуре данных, размер которой соответствует количеству точек. При этом обработка данных происходит с использованием эффективного алгоритма, аналогичного приведённым выше.</p> <p>Допускается одна из следующих ошибок.</p> <ol style="list-style-type: none"> 1) Поиск минимума или максимума не учитывает, что первый подходящий элемент может оказаться на любом месте в исходных данных или вообще отсутствовать. 2) Перепутаны координаты y и x при поиске основания: ищутся максимальные и минимальные значения x при $y = 0$. 3) Перепутаны координаты y и x при поиске высоты: ищется максимальное значение y. 4) При поиске высоты ищется максимум значения координаты x, а не её модуля. 5) При поиске высоты запоминается не модуль, а значение x, при этом при вычислении площади модуль тоже не берётся, в результате может получиться отрицательная площадь. 6) Все вершины определены правильно, но площадь треугольника определена неверно, например, использована неверная формула. 7) Не учитывается, что вычисленное значение площади может быть нецелым. Например, значение площади присваивается переменной целого типа, при вычислении площади используется операция целочисленного деления (div в Паскале, деление целых величин без приведения типов в Си), при форматном выводе используется формат целого числа или допущены другие подобные ошибки, приводящие к неверному результату при дробном ответе. 8) Вершины и площади двух треугольников определены верно, но из них выбирается не больший, а меньший. 9) Неверно обрабатывается ситуация, когда искомый треугольник отсутствует. <p>Допускается наличие от одной до трёх синтаксических ошибок, описанных выше</p>	3

Программа работает в целом верно, эффективно или нет. Возможны переборные решения, при которых все точки хранятся в массиве, из них выбираются подходящие треугольники, вычисляется и сравнивается их площадь.

В реализации алгоритма допущено более 1 ошибки из числа перечисленных в предыдущем пункте или допущены другие ошибки, приводящие к неверной работе программы в отдельных случаях.

Допускается наличие от одной до пяти синтаксических ошибок, описанных выше.

Программа работает в отдельных частных случаях.

Один балл также ставится, если программа написана неверно, но из описания алгоритма и общей структуры программы видно, что экзаменуемый в целом правильно представляет путь решения задачи.

Не выполнено ни одно из перечисленных выше условий.

2

1

0

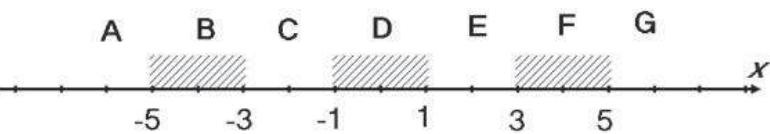
Максимальный балл

4

Критерии оценивания заданий с развёрнутым ответом

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 (включая границы). Программист торопился и написал программу неправильно.



Бейсик	<pre> INPUT x IF x>=-5 OR x<=1 THEN IF x>=-3 AND x<=3 THEN IF x>=-1 AND x<=5 THEN PRINT "принадлежит" ELSE PRINT "не принадлежит" END IF END IF END IF END </pre>
---------------	---

Паскаль	<pre> var x: real; begin readln(x); if (x>=-5) or (x<=1) then if (x>=-3) and (x<=3) then if (x>=-1) and (x<=5) then write('принадлежит') else write('не принадлежит') end. </pre>
----------------	---

Си	<pre> #include <stdio.h> void main(){ float x; scanf("%f", &x); if (x>=-5 x<=1) if (x>=-3 && x<=3) if (x>=-1 && x<=5) printf("принадлежит"); else printf("не принадлежит"); } </pre>
-----------	---

Алгоритмический язык	алг
	нач
	вещ <i>x</i>
	ввод <i>x</i>
	если <i>x</i> $\geq -5 или x\leq 1 то$
	если <i>x</i> $\geq -3 и x\leq 3 то$
	если <i>x</i> $\geq -1 и x\leq 5 то$
	вывод 'принадлежит'
	иначе
	вывод 'не принадлежит'
	все
	все
	кон

Последовательно выполните следующее.

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

Область	Условие 1 ($x \geq -5$ или $x \leq 1$)	Условие 2 ($x \geq -3$ и $x \leq 3$)	Условие 3 ($x \geq -1$ и $x \leq 5$)	Программа выведет	Область обрабатывается верно
A					
B					
C					
D					
E					
F					
G					

В столбцах условий укажите «да», если условие выполнится, «нет», если условие не выполнится, «—» (прочерк), если условие не будет проверяться, «не изв.», если программа ведёт 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.)

Элементы ответа:

Правильно заполненная таблица:

Область	Условие 1 ($x \geq -5$ или $x \leq 1$)	Условие 2 ($x \geq -3$ и $x \leq 3$)	Условие 3 ($x \geq -1$ и $x \leq 5$)	Программа выведет	Область обрабатывается верно
A	Да	Нет	—	—	Нет
B	Да	Нет	—	—	Нет
C	Да	Да	Нет	не принадлежит	Да
D	Да	Да	Да	принадлежит	Да
E	Да	Да	Да	принадлежит	Нет
F	Да	Нет	—	—	Нет
G	Да	Нет	—	—	Нет

Возможны различные способы доработки программы. Ниже приведены различные правильные решения, записанные на разных языках. Данные решения не привязаны к языкам: любую из содержащихся в них идей можно применить, используя любой язык программирования.

Пример на алгоритмическом языке

```
если  $-5 \leq x \leq -3$  или  $-1 \leq x \leq 1$  или  $3 \leq x \leq 5$  то
    вывод 'принадлежит'
```

```
иначе
    вывод 'не принадлежит'
все
```

Пример на Паскале

```
if (abs(x) <= 1) or (3 <= abs(x)) and (abs(x) <= 5) then
    write('принадлежит')
else
    write('не принадлежит');
```

Пример на Бейсике

```
T = ABS(ABS(x)-2)
IF 1 <= T AND T <= 3 THEN
    PRINT "принадлежит"
ELSE
    PRINT "не принадлежит"
END IF
```

Возможны и другие способы доработки.

Обратите внимание! В задаче требуется выполнить **три** действия.

1. Заполнить таблицу.
 2. Исправить ошибку в условном операторе.
 3. Исправить ошибку, связанную с неправильным набором условий.
- Баллы за данное задание начисляются как сумма баллов за верное выполнение каждого действия. Рассмотрим отдельно каждое действие.
1. Действие по заполнению таблицы считается выполненным, если в таблице нет ошибок или ошибки присутствуют только в одной строке.

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 конъюнкций.

В сложных случаях это действие считается выполненным, если программа выдаёт одно из двух сообщений: «принадлежит» или «не принадлежит» – для любых чисел x , при этом программа не стала работать хуже, чем раньше, т. е. для всех точек, для которых программа ранее выдавала верный ответ, доработанная программа также должна выдавать верный ответ.

3. Приведённые ограничения не описывают требуемые области. Первое условие верно для любых x , то есть не позволяет принять никаких решений, второе и третье не захватывают области В и F и не позволяют отделить друг от друга области D и E. Исправлением этой ошибки может быть разбиение области на части и использование дизъюнкции либо использование комбинации каскадных условий. Несколько примеров различных верных решений приведены выше.

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.

Указания по оцениванию	Баллы
Правильно выполнены оба пункта задания. Верно заполнена таблица, исправлены две ошибки. Программа для всех чисел x верно определяет принадлежность точки закрашенной области. В работе (во фрагментах программ) допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора решения.	3
1. Правильно выполнены два действия из трёх (исправлены обе ошибки, но в первом пункте 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). При написании операций сравнения допускается одно неправильное использование строгих/нестрогих неравенств (считается несущественной ошибкой, погрешностью записи). Например, вместо $\langle x \geq 5 \rangle$ используется $\langle x > 5 \rangle$. 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).	2
Правильно выполнено только одно действие из трёх, то есть 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 программы на точках границ областей (вместо нестрогих неравенств в решении были использованы строгие неравенства или наоборот).	1
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).	0
<i>Максимальный балл</i>	3

C2 Дан массив, содержащий 70 целых чисел. Опишите на одном из языков программирования алгоритм, позволяющий найти и вывести наименьшее содержащееся в массиве положительное число, десятичная запись которого оканчивается цифрой 7. Гарантируется, что в массиве есть хотя бы один положительный элемент, десятичная запись которого оканчивается цифрой 7. Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

Паскаль	<pre> const N=70; var a: array [1..N] of integer; i, j, m: integer; begin for i:=1 to N do readln(a[i]); ... end. </pre>
---------	--

Бейсик

```
N=70
DIM A(N) AS INTEGER
DIM I, J, M AS INTEGER
FOR I = 1 TO N
    INPUT A(I)
NEXT I
...
END
```

Си

```
#include <stdio.h>
#define N 70
void main(){
    int a[N];
    int i, j, m;
    for (i=0; i<N; i++)
        scanf("%d", &a[i]);
    ...
}
```

Алгоритмический язык

```
алг
нач
    цел N=70
    целтаб а[1:N]
    цел i, j, m
    нц для i от 1 до N
        ввод а[i]
    кц
    ...
кон
```

В качестве ответа Вам необходимо привести фрагмент программы, 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.

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

В задаче необходимо найти минимальный среди элементов массива, удовлетворяющих заданному в условии ограничению. По сравнению со стандартной задачей поиска минимального среди всех элементов массива дополнительная сложность данной задачи заключается в том, что нельзя брать в качестве первого значения минимума первый элемент массива, так как этот элемент может не удовлетворять заданным ограничениям. Нельзя также принять в качестве первого значения большое число, заведомо превосходящее все возможные значения данных, так как в условии не указан диапазон возможных значений.

Ниже представлены несколько возможных способов решения задачи, проиллюстрированные фрагментами программ на разных языках. Способы решения не привязаны к языкам: любой из этих способов может быть реализован на любом допустимом языке.

Способ 1.

В качестве начального значения минимума принимается значение, заведомо не подходящее под заданные ограничения, например 0.

Пример программы на Паскале

```
m:=0;
for i:=1 to N do begin
    if (a[i]>0) and (a[i] mod 10=7) and ((m=0) or (a[i]<m))
        then m := a[i];
end;
writeln(m);
```

Способ 2.

Вместо проверки специального начального значения используется отдельная переменная, показывающая, был ли уже найден хотя бы один подходящий под ограничения элемент. Для этой переменной следовало бы использовать логический тип, но в условии разрешены только целые переменные, поэтому логическое значение моделируется с помощью целого.

Пример программы на Бейсике

```
M = 0: J = 0
FOR I = 1 TO N
    IF A(I)>0 AND A(i) MOD 10 = 7 AND (J = 0 OR A(I) < M) THEN
        M = A(I)
        J = 1
    END IF
NEXT I
PRINT M
```

Способ 3.

Сначала в массиве ищется первый элемент, соответствующий ограничениям. Затем в оставшейся части массива ищется подходящий наименьший элемент. Этот способ приводит к более длинной (требуется использовать два цикла), но не менее эффективной программе.

Пример программы на алгоритмическом языке

```

i:=1
нц пока не (a[i]>0 и mod(a[i],10)=7)
    i := i+1
кц
m := a[i]
нц для i от i+1 до N
    если a[i]>0 и mod(a[i],10)=7 и a[i]<m
        то m:=a[i]
    все
кц
вывод m

```

Указания по оцениванию	Баллы
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: 1) Не инициализируется или неверно инициализируется переменная m. В частности, нельзя инициализировать эту переменную первым элементом массива. Нельзя также инициализировать её каким-то очень большим значением (например, maxInt в Паскале) 2) Неверно проверяется положительность элементов массива 3) Неверно определяется последняя цифра десятичной записи числа 4) В сложном логическом условии простые проверки верны, но условие в целом построено неверно (например, перепутаны операции И и ИЛИ, неверно расставлены скобки в логическом выражении). 5) Вместо значения элемента проверяется его индекс. 6) Отсутствует вывод ответа. 7) Используется переменная, не объявленная в разделе описания переменных. 8) Не указано или неверно указано условие завершения цикла. 9) Индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно.	1
Ошибок, перечисленных в п. 1–9, две или больше, или алгоритм сформулирован неверно.	0
Максимальный балл	2

C3

Два игрока, Паша и Вова, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Паша. За один ход игрок может добавить в кучу 1 камень или 10 камней. Например, имея кучу из 7 камней, за один ход можно получить кучу из 8 или 17 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче становится не менее 41. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 41 или больше камней.

В начальный момент в куче было S камней, $1 \leq S \leq 40$.

Будем говорить, что игрок имеет *выигрышную стратегию*, если он может выиграть при любых ходах противника. Описать стратегию игрока – значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника.

Выполните следующие задания. Во всех случаях обосновывайте свой ответ.

1. а) Укажите все такие значения числа S , при которых Паша может выиграть в один ход. Обоснуйте, что найдены все нужные значения S , и укажите выигравшие ходы.
- б) Укажите такое значение S , при котором Паша не может выиграть за один ход, но при любом ходе Паши Вова может выиграть своим первым ходом. Опишите выигрышную стратегию Вовы.
2. Укажите два значения S , при которых у Паши есть выигрышная стратегия, причём Паша не может выиграть за один ход, но может выиграть своим вторым ходом независимо от того, как будет ходить Вова. Для указанных значений S опишите выигрышную стратегию Паши.
3. Укажите значение S , при котором у Вовы есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Паши, однако у Вовы нет стратегии, которая позволит ему гарантированно выиграть первым ходом. Для указанного значения S опишите выигрышную стратегию Вовы. Постройте дерево всех партий, возможных при этой выигрышной стратегии Вовы (в виде рисунка или таблицы). На ребрах дерева указывайте, кто делает ход, в узлах – количество камней в куче.

Содержание верного ответа и указания по оцениванию

(допускаются иные формулировки ответа, не искажающие его смысла)

1. а) Паша может выиграть первым ходом, если $S = 31, \dots, 40$. Во всех случаях можно добавить в кучу 10 камней. При меньших значениях S за один ход нельзя получить кучу, в которой больше 40 камней.
- б) Вова может выиграть первым ходом (как бы ни играл Паша), если исходно в куче будет $S=30$ камней. Тогда после первого хода Паши в куче будет 31 камень или 40 камней. В обоих случаях Вова может добавить в кучу 10 камней и выиграть первым ходом.
2. Возможные значения S : 20 и 29. В этих случаях Паша, очевидно, не может выиграть первым ходом. Однако он может получить кучу из 30 камней. Эта позиция разобрана в п. 1б. В ней игрок, который будет ходить (теперь это Вова), выиграть не может, а его противник (то есть Паша) следующим ходом выиграет.

3. Возможное значение S : 28. После первого хода Паши в куче будет 29 или 38 камней. Если в куче станет 38 камней, Вова добавит в кучу 10 камней и выиграет первым ходом. Ситуация, когда в куче 29 камней, разобрана в п. 2. В этой ситуации игрок, который будет ходить (теперь это Вова), выигрывает своим вторым ходом.

В таблице изображено дерево возможных партий при описанной стратегии Вовы. Заключительные позиции (в них выигрывает Вова) подчёркнуты. На рисунке это же дерево изображено в графическом виде (оба способа изображения дерева допустимы).

		Положения после очередных ходов			
Исходная позиция		1-й ход Паши (разобраны все ходы)	1-й ход Вовы (только ход по стратегии)	2-й ход Паши (разобраны все ходы)	2-й ход Вовы (только ход по стратегии)
28	1-й ход Паши (разобраны все ходы)	$28+1=29$	$29+1=30$	<u>$30+1=31$</u>	<u>$31+10=41$</u>
	2-й ход Вовы (только ход по стратегии)			<u>$30+10=40$</u>	<u>$40+10=50$</u>
		<u>$28+10=38$</u>	<u>$38+10=48$</u>		

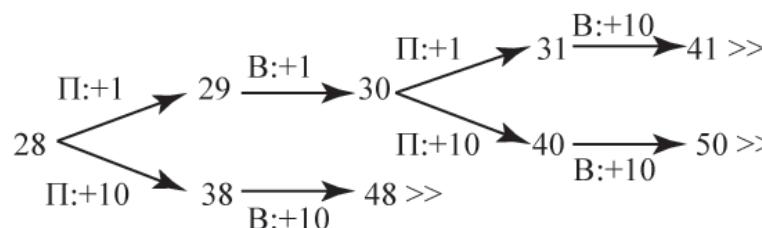


Рис. 1. Дерево всех партий, возможных при Вовиной стратегии.
Знаком >> обозначены позиции, в которых партия заканчивается

В задаче от ученика требуется выполнить 3 задания. Их трудность возрастает. Количество баллов в целом соответствует количеству выполненных заданий (подробнее см. ниже).

Ошибка в решении, не искажающая основного замысла, например арифметическая ошибка при вычислении количества камней в заключительной позиции, при оценке решения не учитывается.

Первое задание считается выполненным, если выполнены оба пункта а) и б). Пункт а) считается выполненным, если правильно указаны все позиции, в которых Паша выигрывает первым ходом, и указано, каким должен быть первый ход. Пункт б) считается выполненным, если правильно указана позиция, в которой Вова выигрывает первым ходом, и описана стратегия Вовы, т. е. показано, как Вова может получить кучу, в которой содержится нужное количество камней, при любом ходе Паши.

Пункт а) первого задания считается выполненным частично, если правильно указаны все позиции, в которых Паша выигрывает первым ходом. Пункт б) первого задания считается выполненным частично, если правильно указана позиция, в которой Вова выигрывает первым ходом, и явно сказано, что при любом ходе Паши Вова может получить кучу, которая содержит нужное для выигрыша количество камней.

Второе задание выполнено, если правильно указаны обе позиции, выигрышные для Паши, и описана соответствующая стратегия Паши – так, как это написано в примере решения, или другим способом, например с помощью дерева всех возможных партий. Третье задание выполнено, если правильно указана позиция, выигрышная для Вовы, и построено дерево всех партий, возможных при Вовиной стратегии. Должно быть явно сказано, что в этом дереве в каждой позиции, где должен ходить Паша, разобраны все возможные ходы, а для позиций, где должен ходить Вова, – только ход, соответствующий стратегии, которую выбрал Вова.

Во всех случаях стратегии могут быть описаны так, как это сделано в примере решения, или другим способом.

Указания по оцениванию	Баллы
Выполнены второе и третье задания. Первое задание выполнено полностью или частично. Здесь и далее допускаются арифметические ошибки, которые не искажают сути решения и не приводят к неправильному ответу (см. выше).	3
Не выполнены условия, позволяющие поставить 3 балла, и выполнено одно из следующих условий. 1. Задание 3 выполнено.	2
2. Первое и второе задания выполнены.	
3. Первое задание выполнено полностью или частично; для заданий 2 и 3 указаны правильные значения S .	
Не выполнены условия, позволяющие поставить 3 или 2 балла, и выполнено одно из следующих условий. 1. Первое задание выполнено.	1
2. Во втором задании правильно указано одно из двух возможных значений S , и для этого значения указана и обоснована выигрышная стратегия Паши.	
3. Первое задание выполнено частично, и для одного из остальных заданий правильно указано значение S .	
4. Для второго и третьего задания правильно указаны значения S .	
Не выполнено ни одно из условий, позволяющих поставить 3, 2 или 1 балл.	0
<i>Максимальный балл</i>	3

C4 На плоскости дан набор точек с целочисленными координатами. Необходимо найти такой треугольник наибольшей площади с вершинами в этих точках, у которого нет общих точек с осью Oy , а одна из сторон лежит на оси Ox .

Напишите эффективную, в том числе по памяти, программу, которая будет решать эту задачу. Размер памяти, которую использует Ваша программа, не должен зависеть от количества точек.

Перед текстом программы кратко опишите используемый алгоритм решения задачи и укажите используемый язык программирования и его версию.

Описание входных данных

В первой строке вводится одно целое положительное число – количество точек N .

Каждая из следующих N строк содержит два целых числа – сначала координата x , затем координата y очередной точки. Числа разделены пробелом.

Описание выходных данных

Программа должна вывести одно число – максимальную площадь треугольника, удовлетворяющего условиям задачи. Если такого треугольника не существует, программа должна вывести ноль.

Пример входных данных:

```
8
-10 0
2 0
0 4
3 3
7 0
5 5
4 0
9 -9
```

Пример выходных данных для приведённого выше примера входных данных:

22.5

Содержание верного ответа и указания по оцениванию

(допускаются иные формулировки ответа, не искажающие его смысла)

Треугольник не имеет общих точек с осью Oy (осью ординат), если абсциссы всех его вершин имеют один и тот же знак, т. е. нужно отдельно рассматривать точки с положительными и отрицательными абсциссами. Для каждой из этих групп треугольник АВС, имеющий максимальную площадь, – это треугольник, у которого вершины А и В лежат на оси абсцисс по одну сторону от начала координат, причем у одной из этих точек абсцисса имеет наибольшую абсолютную величину, а у другой – наименьшую. Третья вершина С – это вершина, имеющая наибольшую по абсолютной величине ординату среди точек, лежащих по ту же сторону от оси ординат, что и точки А, В. Из двух таких «максимальных» треугольников (один лежит по одну сторону от оси ординат, другой – по другую) нужно выбрать тот, который имеет большую площадь.

Программа читает исходные данные, не запоминая все точки в массиве.

После обработки очередной точки программа хранит значения следующих величин:

- максимальное значение Pos_xmax абсциссы для всех прочитанных точек, лежащих на оси Ox правее начала координат;
- минимальное значение Pos_xmin абсциссы для всех прочитанных точек, лежащих на оси Ox правее начала координат;
- максимальное значение Pos_ymax абсолютной величины ординаты для всех прочитанных точек, лежащих правее оси ординат;
- максимальное значение Neg_xmax абсциссы для всех прочитанных точек, лежащих на оси Ox левее начала координат;
- минимальное значение Neg_xmin абсциссы для всех прочитанных точек, лежащих на оси Ox левее начала координат;
- максимальное значение Neg_ymax абсолютной величины ординаты для всех прочитанных точек, лежащих левее оси ординат.

После того как все точки прочитаны, вычисляется максимум

$$S = \max\{(Pos_xmax - Pos_xmin)*Pos_ymax/2, (Neg_xmax - Neg_xmin)*Neg_ymax/2\}$$

(если оба значения не определены, полагаем $S=0$).

В качестве ответа выводится число S .

Пример правильной и эффективной программы на языке Паскаль

```
var
  n: integer;
  x, y: integer;

  Pos_xmin, Pos_xmax: integer;
  Pos_xsearch: boolean;
  Pos_ymax: integer;

  Neg_xmin, Neg_xmax: integer;
  Neg_xsearch: boolean;
  Neg_ymax: integer;
```

```

i: integer;
Pos_s, Neg_s: real;

begin
  Pos_xsearch := true;
  Pos_xmin := 0; Pos_xmax := 0;
  Pos_ymax := 0;
  Neg_xsearch := true;
  Neg_xmin := 0; Neg_xmax := 0;
  Neg_ymax := 0;

  readln(n);
  for i:=1 to n do begin
    readln(x,y);
    if x >0 then begin
      if y=0 then begin
        if Pos_xsearch or (x<Pos_xmin) then Pos_xmin:=x;
        if Pos_xsearch or (x>Pos_xmax) then Pos_xmax:=x;
        Pos_xsearch:=false;
      end
      else begin
        if abs(y)>Pos_ymax then Pos_ymax:=abs(y);
      end;
    end;
    if x < 0 then begin
      if y=0 then begin
        if Neg_xsearch or (x<Neg_xmin) then Neg_xmin:=x;
        if Neg_xsearch or (x>Neg_xmax) then Neg_xmax:=x;
        Neg_xsearch:=false;
      end
      else begin
        if abs(y)>Neg_ymax then Neg_ymax:=abs(y);
      end;
    end;
  end;
  Pos_s := (Pos_xmax-Pos_xmin)*Pos_ymax/2;
  Neg_s := (Neg_xmax-Neg_xmin)*Neg_ymax/2;
  if Pos_s > Neg_s then writeln(Pos_s)
  else writeln(Neg_s);
end.

```

Пример правильной и эффективной программы на языке Бейсик

```

DIM n AS INTEGER
DIM x, y AS INTEGER
DIM Pos_xmin, Pos_xmax AS INTEGER
DIM Pos_xsearch AS INTEGER
DIM Pos_ymax AS INTEGER
DIM Neg_xmin, Neg_xmax AS INTEGER
DIM Neg_xsearch AS INTEGER
DIM Neg_ymax AS INTEGER
DIM i AS INTEGER
DIM Pos_s, Neg_s AS DOUBLE

Pos_xsearch = 1: Neg_xsearch = 1
Pos_ymax = 0: Neg_ymax = 0
Pos_xmin = 0: Pos_xmax = 0
Neg_xmin = 0: Neg_xmax = 0
INPUT n
FOR i = 1 TO n
  INPUT x, y
  IF x > 0 THEN
    IF y = 0 THEN
      IF Pos_xsearch = 1 OR x < Pos_xmin THEN Pos_xmin = x
      IF Pos_xsearch = 1 OR x > Pos_xmax THEN Pos_xmax = x
      Pos_xsearch = 0
    ELSE
      IF ABS(y) > Pos_ymax THEN Pos_ymax = ABS(y)
    END IF
  END IF
  IF x < 0 THEN
    IF y = 0 THEN
      IF Neg_xsearch = 1 OR x < Neg_xmin THEN Neg_xmin = x
      IF Neg_xsearch = 1 OR x > Neg_xmax THEN Neg_xmax = x
      Neg_xsearch = 0
    ELSE
      IF ABS(y) > Neg_ymax THEN Neg_ymax = ABS(y)
    END IF
  END IF
NEXT i
Pos_s = (Pos_xmax - Pos_xmin) * Pos_ymax / 2
Neg_s = (Neg_xmax - Neg_xmin) * Neg_ymax / 2
IF Pos_s > Neg_s THEN
  PRINT Pos_s
ELSE
  PRINT Neg_s
END IF

```

Пример правильной и эффективной программы на алгоритмическом языке

```

алг
нач
    цел п
    цел х, у
    цел Pos_xmin=0, Pos_xmax=0
    лог Pos_xsearch=да
    цел Pos_ymax=0
    цел Neg_xmin=0, Neg_xmax=0
    лог Neg_xsearch=да
    цел Neg_ymax=0
    вещ Pos_s, Neg_s
    ввод п
    нц п раз
        ввод х, у
        если х > 0 то
            если у=0
                то
                    если Pos_xsearch или х<Pos_xmin то Pos_xmin:=х все
                    если Pos_xsearch или х>Pos_xmax то Pos_xmax:=х все
                    Pos_xsearch:=нет
                иначе
                    если iabs(у)>Pos_ymax то Pos_ymax:=iabs(у) все
                все
            все
        если х < 0
            если у=0
                то
                    если Neg_xsearch или х<Neg_xmin то Neg_xmin:=х все
                    если Neg_xsearch или х>Neg_xmax то Neg_xmax:=х все
                    Neg_xsearch:=нет
                иначе
                    если iabs(у)>Neg_ymax то Neg_ymax:=iabs(у) все
                все
            все
    кон
    Pos_s:=(Pos_xmax-Pos_xmin)*Pos_ymax/2
    Neg_s:=(Neg_xmax-Neg_xmin)*Neg_ymax/2
    если Pos_s > Neg_s
        то вывод Pos_s
    иначе вывод Neg_s
    все
кон

```

Указания по оцениванию	Баллы
<p>Программа правильно работает для любых входных данных произвольного размера и находит ответ, не сохраняя входные данные в массиве. Допускается наличие в тексте программы одной синтаксической ошибки: пропущен или неверно указан знак пунктуации, неверно написано или пропущено зарезервированное слово языка программирования, не описана или неверно описана переменная, применяется операция, недопустимая для соответствующего типа данных (если одна и та же ошибка встречается несколько раз, то это считается за одну ошибку).</p> <p>Программа работает верно, но размер используемой памяти зависит от количества исходных данных. Например, входные данные (координаты точек) запоминаются в массиве или другой структуре данных, размер которой соответствует количеству точек. При этом обработка данных происходит с использованием эффективного алгоритма, аналогичного приведённым выше.</p> <p>Допускается одна из следующих ошибок.</p> <ol style="list-style-type: none"> Поиск минимума или максимума не учитывает, что первый подходящий элемент может оказаться на любом месте в исходных данных или вообще отсутствовать. Перепутаны координаты x и y при поиске основания, ищутся максимальные и минимальные значения y при $x = 0$. Перепутаны координаты x и y при поиске высоты: ищется максимальное значение x. При поиске высоты ищется максимум значения координаты y, а не её модуля. При поиске высоты запоминается не модуль, а значение y, при этом при вычислении площади модуль тоже не берётся, в результате может получиться отрицательная площадь. Все вершины определены правильно, но площадь треугольника определена неверно, например, использована неверная формула. Не учитывается, что вычисленное значение площади может быть нецелым. Например, значение площади присваивается переменной целого типа, при вычислении площади используется операция целочисленного деления (div в Паскале, деление целых величин без приведения типов в Си), при форматном выводе используется формат целого числа, или имеются другие подобные ошибки, приводящие к неверному результату при дробном ответе. Вершины и площади двух треугольников определены верно, но из них выбирается не больший, а меньший. Неверно обрабатывается ситуация, когда искомый треугольник отсутствует. <p>Допускается наличие от одной до трёх синтаксических ошибок, описанных выше.</p>	4
<p>Программа работает верно, но размер используемой памяти зависит от количества исходных данных. Например, входные данные (координаты точек) запоминаются в массиве или другой структуре данных, размер которой соответствует количеству точек. При этом обработка данных происходит с использованием эффективного алгоритма, аналогичного приведённым выше.</p> <p>Допускается одна из следующих ошибок.</p> <ol style="list-style-type: none"> Поиск минимума или максимума не учитывает, что первый подходящий элемент может оказаться на любом месте в исходных данных или вообще отсутствовать. Перепутаны координаты x и y при поиске основания, ищутся максимальные и минимальные значения y при $x = 0$. Перепутаны координаты x и y при поиске высоты: ищется максимальное значение x. При поиске высоты ищется максимум значения координаты y, а не её модуля. При поиске высоты запоминается не модуль, а значение y, при этом при вычислении площади модуль тоже не берётся, в результате может получиться отрицательная площадь. Все вершины определены правильно, но площадь треугольника определена неверно, например, использована неверная формула. Не учитывается, что вычисленное значение площади может быть нецелым. Например, значение площади присваивается переменной целого типа, при вычислении площади используется операция целочисленного деления (div в Паскале, деление целых величин без приведения типов в Си), при форматном выводе используется формат целого числа, или имеются другие подобные ошибки, приводящие к неверному результату при дробном ответе. Вершины и площади двух треугольников определены верно, но из них выбирается не больший, а меньший. Неверно обрабатывается ситуация, когда искомый треугольник отсутствует. <p>Допускается наличие от одной до трёх синтаксических ошибок, описанных выше.</p>	3

Программа работает в целом верно, эффективно или нет. Возможны переборные решения, при которых все точки хранятся в массиве, из них выбираются подходящие треугольники, вычисляется и сравнивается их площадь.	2
В реализации алгоритма допущено более 1 ошибки из числа перечисленных в предыдущем пункте или допущены другие ошибки, приводящие к неверной работе программы в отдельных случаях.	
Допускается наличие от одной до пяти синтаксических ошибок, описанных выше.	
Программа работает в отдельных частных случаях.	
Один балл также ставится, если программа написана неверно, но из описания алгоритма и общей структуры программы видно, что экзаменуемый в целом правильно представляет путь решения задачи.	1
Не выполнено ни одно из перечисленных выше условий.	0
<i>Максимальный балл</i>	4

Указания по оцениванию	Баллы
<p>Программа правильно работает для любых входных данных произвольного размера и находит ответ, не сохраняя входные данные в массиве. Допускается наличие в тексте программы одной синтаксической ошибки: пропущен или неверно указан знак пунктуации, неверно написано или пропущено зарезервированное слово языка программирования, не описана или неверно описана переменная, применяется операция, недопустимая для соответствующего типа данных (если одна и та же ошибка встречается несколько раз, то это считается за одну ошибку).</p> <p>Программа работает верно, но размер используемой памяти зависит от количества исходных данных. Например, входные данные (координаты точек) запоминаются в массиве или другой структуре данных, размер которой соответствует количеству точек. При этом обработка данных происходит с использованием эффективного алгоритма, аналогичного приведённым выше.</p> <p>Допускается одна из следующих ошибок.</p> <ol style="list-style-type: none"> 1) Поиск минимума или максимума не учитывает, что первый подходящий элемент может оказаться на любом месте в исходных данных или вообще отсутствовать 2) Перепутаны координаты y и x при поиске основания: ищутся максимальные и минимальные значения x при $y = 0$. 3) Перепутаны координаты y и x при поиске высоты: ищется максимальное значение y. 4) При поиске высоты ищется максимум значения координаты x, а не её модуля 5) При поиске высоты запоминается не модуль, а значение x, при этом при вычислении площади модуль тоже не берётся, в результате может получиться отрицательная площадь 6) Все вершины определены правильно, но площадь треугольника определена неверно, например, использована неверная формула. 7) Не учитывается, что вычисленное значение площади может быть нецелым. Например, значение площади присваивается переменной целого типа, при вычислении площади используется операция целочисленного деления (div в Паскале, деление целых величин без приведения типов в Си), при форматном выводе используется формат целого числа или допущены другие подобные ошибки, приводящие к неверному результату при дробном ответе 8) Вершины и площади двух треугольников определены верно, но из них выбирается не больший, а меньший 9) Неверно обрабатывается ситуация, когда искомый треугольник отсутствует <p>Допускается наличие от одной до трёх синтаксических ошибок, описанных выше.</p>	4
<p>Программа работает в целом верно, эффективно или нет. Возможны переборные решения, при которых все точки хранятся в массиве, из них выбираются подходящие треугольники, вычисляется и сравнивается их площадь</p> <p>В реализации алгоритма допущено более 1 ошибки из числа перечисленных в предыдущем пункте или допущены другие ошибки, приводящие к неверной работе программы в отдельных случаях.</p> <p>Допускается наличие от одной до пяти синтаксических ошибок, описанных выше.</p> <p>Программа работает в отдельных частных случаях.</p> <p>Один балл также ставится, если программа написана неверно, но из описания алгоритма и общей структуры программы видно, что экзаменуемый в целом правильно представляет путь решения задачи.</p> <p>Не выполнено ни одно из перечисленных выше условий.</p>	2
	Максимальный балл

<p>Программа работает в целом верно, эффективно или нет. Возможны переборные решения, при которых все точки хранятся в массиве, из них выбираются подходящие треугольники, вычисляется и сравнивается их площадь</p> <p>В реализации алгоритма допущено более 1 ошибки из числа перечисленных в предыдущем пункте или допущены другие ошибки, приводящие к неверной работе программы в отдельных случаях.</p> <p>Допускается наличие от одной до пяти синтаксических ошибок, описанных выше.</p> <p>Программа работает в отдельных частных случаях.</p> <p>Один балл также ставится, если программа написана неверно, но из описания алгоритма и общей структуры программы видно, что экзаменуемый в целом правильно представляет путь решения задачи.</p> <p>Не выполнено ни одно из перечисленных выше условий.</p>	1
	Максимальный балл

Bap	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15
1	3	1	2	3	2	4	1	2	2	2	3	3	4	112121	90	6	62	60	105	5	106	24	A76	DFBH	170	60	127	31
2	3	2	3	3	4	4	1	3	1	2	2	3	3	121211	100	15	60	120	192	5	104	24	A36	DFCH	150	60	131	31
3	3	2	2	3	2	4	1	3	2	2	3	3	4	121211	90	15	62	120	105	5	106	24	A76	DFCH	170	60	127	31
4	3	1	3	3	4	4	1	2	1	2	2	3	3	112121	100	6	60	60	192	5	104	24	A36	DFBH	150	60	131	31