

В экзаменационных заданиях используются следующие соглашения.

1. Обозначения для логических связок (операций):

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

2. Два логических выражения, 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. Обозначения Мбайт и Кбайт используются в традиционном для информатики смысле – как обозначения единиц измерения, соотношение которых с единицей «байт» выражается степенью двойки.

Часть 1

При выполнении заданий этой части (A1–A13) правильный ответ обведите кружком.

A1 Даны 4 целых числа, записанных в двоичной системе: 10001011; 10111000; 10011011; 10110100. Сколько среди них чисел, больших, чем $9A_{16}$?

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

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

	А	В	С	D	E	F	Z
А		5	8	25			39
В	5		1	20			
С	8	1		11			28
D	25	20	11		4	6	10
E				4			8
F				6			2
Z	39		28	10	8	2	

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

- 1) 23 2) 25 3) 35 4) 39

A3 Дан фрагмент таблицы истинности выражения F:

x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	F
0	1	0	1	1	1	0	1	1	1	1
1	0	1	1	0	0	1	1	1	0	0
0	1	0	1	0	1	0	0	1	0	0

Каким выражением может быть F?

- 1) $\neg x1 \wedge x2 \wedge \neg x3 \wedge x4 \wedge x5 \wedge x6 \wedge \neg x7 \wedge x8 \wedge x9 \wedge x10$
- 2) $\neg x1 \vee x2 \vee \neg x3 \vee x4 \vee x5 \vee x6 \vee \neg x7 \vee x8 \vee x9 \vee x10$
- 3) $\neg x1 \vee x2 \vee \neg x3 \vee x4 \vee \neg x5 \vee x6 \vee \neg x7 \vee \neg x8 \vee x9 \vee \neg x10$
- 4) $\neg x1 \wedge x2 \wedge \neg x3 \wedge x4 \wedge \neg x5 \wedge x6 \wedge \neg x7 \wedge \neg x8 \wedge x9 \wedge \neg x10$

A9 По каналу связи передаются сообщения, содержащие только 4 буквы: С, Л, О, Н; для передачи используется двоичный код, допускающий однозначное декодирование. Для букв С, О, Н используются такие кодовые слова: С: 011, О: 00, Н: 11.

Укажите такое кодовое слово для буквы Л, при котором код будет допускать однозначное декодирование. Если таких кодов несколько, укажите тот, у которого меньшая длина.

- 1) 1 2) 10 3) 010 4) 0

A10 На числовой прямой даны два отрезка: P = [3, 33] и Q = [22, 44]. Выберите такой отрезок А, что формула

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

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

- 1) [2, 20] 2) [10, 25] 3) [20, 40] 4) [25, 30]

A11 При регистрации в компьютерной системе каждому пользователю выдаётся пароль, состоящий из 15 символов и содержащий только символы К, Т, А, М, Р, О, Ф, Н, И. 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) 210 байт 2) 240 байт 3) 270 байт 4) 300 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	Паскаль
<pre>n = 10 FOR i = 0 TO n K = A(i) A(K) = 0 NEXT i</pre>	<pre>n:=10; for i:=0 to n do begin K := A[i]; A[K] := 0; end;</pre>
Си	Алгоритмический язык
<pre>n=10; for (i = 0; i <= n; i++){ K = A[i]; A[K] = 0; }</pre>	<pre>n:=10 нц для i от 0 до n K := A[i] A[K] := 0 кц</pre>

В начале выполнения этого фрагмента в массиве находились числа 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, т. е. A[0] = 10, A[1] = 9 и т. д. Сколько элементов массива после выполнения программы будут иметь ненулевые значения?

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

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 этой клетке и выполнив предложенную программу, РОБОТ уцелеет и остановится в закрашенной клетке (клетка F6)?

НАЧАЛО

ПОКА **снизу свободно** ИЛИ **справа свободно**

ПОКА **снизу свободно**

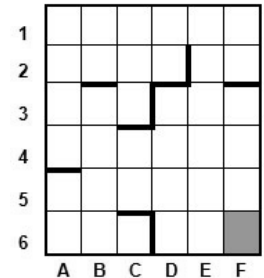
вниз

КОНЕЦ ПОКА

вправо

КОНЕЦ ПОКА

КОНЕЦ



- 1) 10 2) 13 3) 16 4) 20

Часть 2

Ответом к заданиям этой части (В1–В15) является число, последовательность букв или цифр. Впишите ответы в указанном месте без пробелов, запятых и других дополнительных символов.

В1 У исполнителя Троечник две команды, которым присвоены номера:

1. прибавь 3,
2. умножь на 2.

Первая из этих команд увеличивает число на экране на 3, вторая – умножает его на 2. Программа для исполнителя Троечник – это последовательность номеров команд.

Например, 1211 – это программа

- прибавь 3
умножь на 2
прибавь 3
прибавь 3

Эта программа преобразует число 2 в число 16.

Запишите программу, которая преобразует число 12 в число 123 и содержит не более 5 команд. Если таких программ более одной, то запишите любую из них.

Ответ: _____.

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

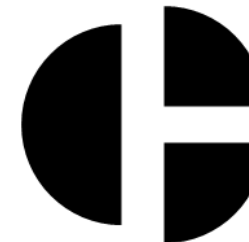
Бейсик	Паскаль
<pre>a = 30 b = 6 a = a * 3 / b IF a < b THEN c = 3*a - 5 * (b+2) ELSE c = 3*a + 5 * (b+2) END IF</pre>	<pre>a := 30; b := 6; a := a * 3 / b; if a < b then c := 3*a - 5 * (b+2) else c := 3*a + 5 * (b+2);</pre>
Си	Алгоритмический язык
<pre>a = 30; b = 6; a = a * 3 / b; if (a < b) c = 3*a - 5 * (b+2); else c = 3*a + 5 * (b+2);</pre>	<pre>a := 30 b := 6 a := a * 3 / b если a < b то c := 3*a - 5*(b+2) иначе c := 3*a + 5*(b+2) все</pre>

Ответ: _____.

В3	А	В	С
1	12		= A1*4
2	= B1/A1	= C1/B1	= B2 + A1/6

Дан фрагмент электронной таблицы (см. выше).

Какое число должно быть записано в ячейке В1, чтобы построенная после выполнения диаграммы по значениям диапазона ячеек А2:С2 соответствовала рисунку?



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

Ответ: _____.

В4 Сколько слов длины 5 можно составить из букв Е, Г, Э? Каждая буква может входить в слово несколько раз.

Ответ: _____.

В5 Определите, что будет напечатано в результате выполнения программы (записанной ниже на разных языках программирования):

Бейсик	Паскаль
<pre>DIM N, S AS INTEGER N = 1 S = 0 WHILE S <= 365 S = S + 37 N = N * 2 WEND PRINT N</pre>	<pre>var n, s: integer; begin n := 1; s := 0; while s <= 365 do begin s := s + 37; n := n * 2 end; write(n) end.</pre>

Си	Алгоритмический
<pre>#include<stdio.h> void main() { int n, s; n = 1; s = 0; while (s <= 365) { s = s + 37; n = n * 2; } printf("%d", n); }</pre>	<pre>алг нач цел n, s n := 1 s := 0 нц пока s <= 365 s := s + 37 n := n * 2 кц вывод n кон</pre>

Ответ: _____.

- B6** Алгоритм вычисления значений функций $F(n)$ и $G(n)$, где n – натуральное число, задан следующими соотношениями:
 $F(1) = 1; G(1) = 1;$
 $F(n) = F(n-1) - G(n-1), G(n) = F(n-1) + G(n-1)$, при $n > 2$
 Чему равно значение величины $F(5)/G(5)$?
 В ответе запишите только натуральное число.

Ответ: _____.

- B7** Решите уравнение:

$$100_7 + x = 210_5$$

Ответ запишите в шестеричной системе (основание системы счисления в ответе писать не нужно).

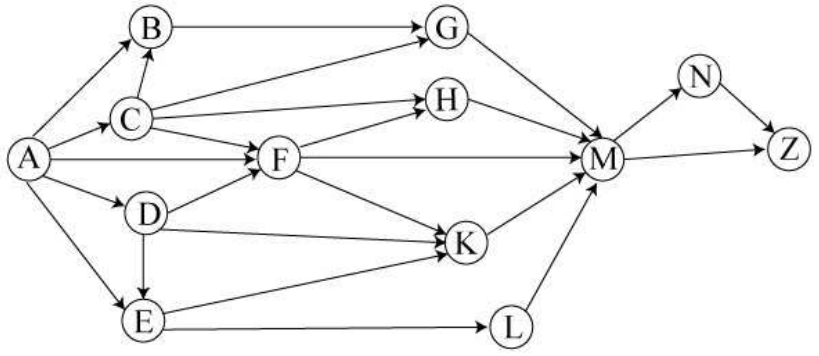
Ответ: _____.

- B8** Ниже на четырёх языках записан алгоритм. Получив на вход число x , этот алгоритм печатает два числа: a и b . Укажите наименьшее из таких чисел x , при вводе которого алгоритм печатает сначала 3, а потом 2.

Бейсик	Паскаль
<pre>DIM X, A, B AS INTEGER INPUT X A=0: B=0 WHILE X > 0 A = A+1 IF B < (X MOD 8) THEN B = X MOD 8 END IF X = X \ 8 WEND PRINT A PRINT B</pre>	<pre>var x, a, b: integer; begin readln(x); a:=0; b:=0; while x>0 do begin a:=a + 1; if b < (x mod 8) then b:=x mod 8; x:=x div 8; end; writeln(a); write(b); end.</pre>
Си	Алгоритмический язык
<pre>#include<stdio.h> void main() { int x, a, b; scanf("%d", &x); a=0; b=0; while (x>0){ a = a+1; if (b < (x%8){ b = x%8; } x = x/8; } printf("%d\n%d", a, b); }</pre>	<pre>алг нач цел x, a, b ввод x а:=0; б:=0 нц пока x>0 а:=а+1 если b < mod(x,8) то б:=mod(x,8) все х:=div(x,10) кц вывод а, б кон</pre>

Ответ: _____.

B9 На рисунке – схема дорог, связывающих города А, В, С, D, E, F, G, H, K, L, M, N, Z. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город Z?



Ответ: _____.

B10 Документ объёмом 10 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 на сколько, если

- средняя скорость передачи данных по каналу связи составляет 2^{23} бит в секунду,
- объём сжатого архиватором документа равен 30% 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. Например, если способ Б быстрее способа А на 23 секунды, в ответе нужно написать Б23.

Слов «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 заданному IP-адресу узла и маске.

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

IP-адрес узла: 224.24.254.134
 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	232	224	134	24	8	0

Пример.

Пусть искомым IP-адрес 192.168.128.0, и дана таблица

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

В этом случае правильный ответ будет записан в виде: HBAF.

Ответ: _____.

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, тыс.
Львов & (Ужгород Мукачево)	275
Львов & Ужгород	123
Львов & Ужгород & Мукачево	41

Какое количество страниц (в тыс.) будет найдено по запросу
 Львов & Мукачево?

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. прибавь 4.

Первая из них увеличивает на 2 число на экране, вторая увеличивает это число на 4.

Программа для Удвоителя – это последовательность команд.

Сколько существует программ, которые число 2 преобразуют в число 22?

Ответ: _____.

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

Бейсик	Паскаль
<pre> DIM A, B, T, N, P AS INTEGER A = -25: B = 25 P = 130 N = 0 FOR T = A TO B IF F(T) > P THEN N = N+1 END IF NEXT T PRINT N FUNCTION F(x) F = 16*(9-x)*(9-x)+127 END FUNCTION </pre>	<pre> var a, b, t, N, P :integer; Function F(x: integer):integer; begin F := 16*(9-x)*(9-x)+127; end; BEGIN a := -25; b := 25; P := 130; N := 0; for t := a to b do begin if (F(t) > P) then begin N := N+1; end; end; write(N); END. </pre>
Си	Алгоритмический язык
<pre> #include<stdio.h> int F(int x) { return 16*(9-x)*(9-x)+127; } void main() { int a, b, t, N, P; a = -25; b = 25; P = 130; N = 0; for (t=a; t<=b; t++){ if (F(t) > P) { N++; } } printf("%d", N); } </pre>	<pre> алг нач цел a, b, t, N, P a := -25; b := 25 P := 130 N := 0 нц для t от a до b если F(t) > P то N := N+1 все кц вывод N кон алг цел F(цел x) нач знач := 16*(9-x)*(9-x)+127 кон </pre>

Ответ: _____.

B15 Сколько существует различных наборов значений логических переменных $x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4, z_1, z_2, z_3, z_4$, которые удовлетворяют всем перечисленным ниже условиям?

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

$$(\neg x_1 \wedge y_1 \wedge z_1) \vee (x_1 \wedge \neg y_1 \wedge z_1) \vee (x_1 \wedge y_1 \wedge \neg z_1) = 1$$

$$(\neg x_2 \wedge y_2 \wedge z_2) \vee (x_2 \wedge \neg y_2 \wedge z_2) \vee (x_2 \wedge y_2 \wedge \neg z_2) = 1$$

$$(\neg x_3 \wedge y_3 \wedge z_3) \vee (x_3 \wedge \neg y_3 \wedge z_3) \vee (x_3 \wedge y_3 \wedge \neg z_3) = 1$$

$$(\neg x_4 \wedge y_4 \wedge z_4) \vee (x_4 \wedge \neg y_4 \wedge z_4) \vee (x_4 \wedge y_4 \wedge \neg z_4) = 1$$

В ответе не нужно перечислять все различные наборы значений переменных $x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4, z_1, z_2, z_3, z_4$, при которых выполнена данная система равенств. В качестве ответа Вам нужно указать количество таких наборов.

Ответ: _____.

Часть 3

Для записи ответов на задания этой части (C1–C4) используйте отдельный лист. Запишите сначала номер задания (C1, C2 и т. д.), а затем – полное решение. Ответы записывайте чётко и разборчиво.

C1 Требовалось написать программу, при выполнении которой с клавиатуры вводится натуральное число, не превосходящее 10^8 , и выводится его первая (старшая) цифра. Ученик написал такую программу:

Бейсик	Паскаль
<pre> DIM N AS LONG INPUT N WHILE N>10 N = N MOD 10 WEND PRINT N END </pre>	<pre> var n: longint; begin read(n); while n>10 do begin n := n mod 10 end; write(n); end. </pre>
Си	Алгоритмический язык
<pre> #include <stdio.h> void main(){ long int n; scanf("%ld",&n); while (n>10) { n = n%10; } printf("%ld", n); } </pre>	<pre> алг нач цел n ввод n нц пока n>10 n := mod(n,10) кц вывод n кон </pre>

Последователно изпълните следните задачи.

1. Напишите, что выведет эта программа при вводе числа 1984.
2. Приведите пример числа, при вводе которого программа выдаст верный ответ.
3. Найдите в программе все ошибки (их может быть одна или несколько). Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.

Обратите внимание: вам нужно исправить приведённую программу, а не написать свою. Вы можете только заменять ошибочные строки, но не можете удалять строки или добавлять новые. Заменять следует только ошибочные строки: за исправления, внесённые в строки, не содержащие ошибок, баллы будут снижаться.

C2 Дан массив, содержащий 2014 положительных целых чисел. Напишите на одном из языков программирования программу, которая находит в этом массиве количество элементов, значение которых более чем в два раза превосходит значение предшествующего элемента. Например, для массива из 6 элементов, содержащего числа 2, 5, 10, 15, 40, 100, программа должна выдать ответ 3 (условию соответствуют элементы со значениями 5, 40 и 100). Программа должна вывести общее количество подходящих элементов, значения элементов выводить не нужно. Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из описанных переменных.

Паскаль	Бейсик
<pre>const N=2014; var a: array [1..N] of integer; i, j, k: integer; begin for i:=1 to N do readln(a[i]); ... end.</pre>	<pre>N=2014 DIM A(N) AS INTEGER DIM I, J, K AS INTEGER FOR I = 1 TO N INPUT A(I) NEXT I ... END</pre>
Cи	Алгоритмический язык
<pre>#include <stdio.h> #define N 2014 void main(){ int a[N]; int i, j, k; for (i=0; i<N; i++) scanf("%d", &a[i]); ... }</pre>	<pre>алг нач цел N=2014 целтаб a[1:N] цел i, j, k 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). В этом случае вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии.

C3 Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в кучу **один** или **два** камня или увеличить количество камней в куче в **три** раза. Например, имея кучу из 15 камней, за один ход можно получить кучу из 16, 17 или 45 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней. Игра завершается в тот момент, когда количество камней в куче становится не менее 64. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 64 или больше камней. В начальный момент в куче было S камней, $1 \leq S \leq 63$.

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) При каких значениях числа S Петя может выиграть в один ход? Укажите все такие значения.
 б) Укажите такое значение S , при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом. Опишите выигрышную стратегию Вани.
2. Укажите три таких значения S , при которых у Пети есть выигрышная стратегия, причём
 – Петя не может выиграть за один ход, но
 – Петя может выиграть своим вторым ходом, независимо от того, как будет ходить Ваня.
 Для каждого из указанных значений S опишите выигрышную стратегию Пети.
3. Укажите значение S , при котором у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, однако у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом.
 Для указанного значения S опишите выигрышную стратегию Вани. Постройте дерево всех партий, возможных при этой выигрышной стратегии Вани (в виде рисунка или таблицы). На рёбрах дерева указывайте, кто делает ход, в узлах – количество камней в позиции.

Дед Мороз и Снегурочка приходят на детские утренники с мешком конфет. Дед Мороз делит конфеты поровну между всеми присутствующими детьми (детей на утреннике никогда не бывает больше 100), а оставшиеся конфеты отдает Снегурочке. Снегурочка каждый раз записывает в блокнот количество полученных конфет. Если конфеты разделились между всеми детьми без остатка, Снегурочка ничего не получает и ничего не записывает. Когда утренники закончились, Деду Морозу стало интересно, какое число чаще всего записывала Снегурочка. Дед Мороз и Снегурочка – волшебные, поэтому число утренников N , на которых они побывали, может быть очень большим.

Напишите программу, которая будет решать эту задачу. Перед текстом программы кратко опишите алгоритм решения задачи и укажите используемый язык программирования и его версию.

Желательно, чтобы программа была эффективной как по времени работы, так и по используемой памяти. Программу будем считать эффективной по памяти, если используемая память не зависит от размера входных данных (то есть числа утренников). Программу будем считать эффективной по времени, если при увеличении размера входных данных N в t раз (t – любое число) время её работы увеличивается не более чем в t раз.

Описание входных данных

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

Каждая из следующих N строк содержит два целых числа: сначала D – количество пришедших на очередной утренник детей, а затем K – количество конфет в мешке Деда Мороза на этом утреннике. Гарантируется выполнение следующих соотношений:

$$1 \leq N \leq 10000$$

$$1 \leq D \leq 100 \text{ (для каждого } D)$$

$$D \leq K \leq 1000 \text{ (для каждой пары } D, K)$$

Описание выходных данных

Программа должна вывести одно число – то, которое Снегурочка записывала чаще всего. Если несколько чисел записывались одинаково часто, надо вывести большее из них. Если Снегурочка ни разу ничего не записывала, надо вывести ноль.

Пример входных данных:

```
7
10 58
15 315
20 408
100 1000
32 63
32 63
11 121
```

Пример выходных данных для приведённого выше примера входных данных:

```
31
```

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) *импликация* (следование) обозначается \rightarrow (например, $A \rightarrow B$);
- e) *тождество* обозначается \equiv (например, $A \equiv B$); выражение $A \equiv B$ истинно тогда и только тогда, когда значения A и B совпадают (либо они оба истинны, либо они оба ложны);
- f) символ 1 используется для обозначения истины (истинного высказывания); символ 0 – для обозначения лжи (ложного высказывания).

2. Два логических выражения, 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. Обозначения Мбайт и Кбайт используются в традиционном для информатики смысле – как обозначения единиц измерения, соотношение которых с единицей «байт» выражается степенью двойки.

Часть 1

При выполнении заданий этой части (A1–A13) правильный ответ обведите кружком.

A1 Даны 4 целых числа, записанных в двоичной системе.: 10101011; 10011100; 11000111; 10110100. Сколько среди них чисел, меньших, чем BC_{16} ?

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

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	9	26			37
B	4		3	21			
C	9	3		13			27
D	26	21	13		4	7	10
E				4			8
F				7			2
Z	37		27	10	8	2	

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

- 1) 26
- 2) 29
- 3) 36
- 4) 37

A3 Дан фрагмент таблицы истинности выражения F:

x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	F
0	1	0	1	1	1	0	1	1	1	0
1	0	1	1	0	0	1	1	1	0	1
0	1	0	1	0	1	0	0	1	0	0

Каким выражением может быть F?

- 1) $x1 \wedge \neg x2 \wedge x3 \wedge \neg x4 \wedge x5 \wedge \neg x6 \wedge x7 \wedge x8 \wedge \neg x9 \wedge x10$
- 2) $x1 \vee \neg x2 \vee x3 \vee \neg x4 \vee x5 \vee \neg x6 \vee x7 \vee x8 \vee \neg x9 \vee x10$
- 3) $x1 \vee \neg x2 \vee x3 \vee x4 \vee \neg x5 \vee \neg x6 \vee x7 \vee x8 \vee x9 \vee \neg x10$
- 4) $x1 \wedge \neg x2 \wedge x3 \wedge x4 \wedge \neg x5 \wedge \neg x6 \wedge x7 \wedge x8 \wedge x9 \wedge \neg x10$

A9 По каналу связи передаются сообщения, содержащие только 4 буквы: А, Т, О, М; для передачи используется двоичный код, допускающий однозначное декодирование. Для букв Т, О, М используются такие кодовые слова: Т: 100, О: 00, М: 11.

Укажите такое кодовое слово для буквы А, при котором код будет допускать однозначное декодирование. Если таких кодов несколько, укажите тот, у которого меньшая длина.

- 1) 1 2) 0 3) 01 4) 101

A10 На числовой прямой даны два отрезка: P = [3, 33] и Q = [22, 44]. Выберите такой отрезок А, что формула

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

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

- 1) [31, 45] 2) [21, 35] 3) [11, 25] 4) [1, 15]

A11 При регистрации в компьютерной системе каждому пользователю выдаётся пароль, состоящий из 15 символов и содержащий только символы И, Н, Ф, О, Р, М, А, Т, К. 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) 175 байт 3) 200 байт 4) 225 байт

A12 В программе описан одномерный целочисленный массив с индексами от 0 до 9. Ниже представлен записанный на разных языках программирования фрагмент одной и той же программы, обрабатывающей данный массив.

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

В начале выполнения этого фрагмента в массиве находились числа 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, т. е. A[0] = 9, A[1] = 8 и т. д. Сколько элементов массива после выполнения программы будут иметь ненулевые значения?

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

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*

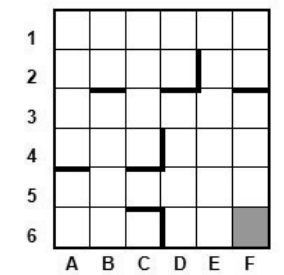
КОНЕЦ ЕСЛИ

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 этой клетке и выполнив предложенную программу, РОБОТ уцелеет и остановится в закрашенной клетке (клетка F6)?

НАЧАЛО
ПОКА **снизу свободно ИЛИ справа свободно**
ПОКА **справа свободно**
вправо
КОНЕЦ ПОКА
вниз
КОНЕЦ ПОКА
КОНЕЦ



- 1) 6 2) 10 3) 13 4) 16

Ответом к заданиям этой части (В1–В15) является число, последовательность букв или цифр. Впишите ответы в указанном месте без пробелов, запятых и других дополнительных символов.

В1 У исполнителя Троечник две команды, которым присвоены номера:

1. прибавь 3,
2. умножь на 2.

Первая из этих команд увеличивает число на экране на 3, вторая – умножает его на 2. Программа для исполнителя Троечник – это последовательность номеров команд.

Например, 1211 – это программа

- прибавь 3
умножь на 2
прибавь 3
прибавь 3

Эта программа преобразует число 2 в число 16.

Запишите программу, которая преобразует число 11 в число 103 и содержит не более 5 команд. Если таких программ более одной, то запишите любую из них.

Ответ: _____.

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

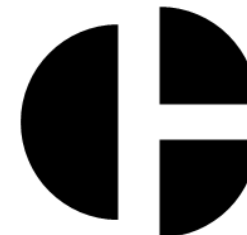
Бейсик	Паскаль
<pre>a = 30 b = 6 a = a * 3 / b IF a < b THEN c = 3*a - 5 * (b+3) ELSE c = 3*a + 5 * (b+3) END IF</pre>	<pre>a := 30; b := 6; a := a * 3 / b; if a < b then c := 3*a - 5 *(b+3) else c := 3*a + 5 * (b+3);</pre>
Си	Алгоритмический язык
<pre>a = 30; b = 6; a = a * 3 / b; if (a < b) c = 3*a - 5 * (b+3); else c = 3*a + 5 * (b+3);</pre>	<pre>а := 30 b := 6 а := а * 3 / b если а < b то с := 3*а - 5*(b+3) иначе с := 3*а + 5*(b+3) все</pre>

Ответ: _____.

В3	A	B	C
1	15		=A1*25
2	=B1/A1	=C1/B1	=B2 + A1/3

Дан фрагмент электронной таблицы (см. выше).

Какое число должно быть записано в ячейке В1, чтобы построенная после выполнения вычислений диаграмма по значениям диапазона ячеек А2:С2 соответствовала рисунку?



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

Ответ: _____.

В4 Сколько слов длины 6 можно составить из букв Е, Г, Э? Каждая буква может входить в слово несколько раз.

Ответ: _____.

В5 Определите, что будет напечатано в результате выполнения программы (записанной ниже на разных языках программирования).

Бейсик	Паскаль
<pre>DIM N, S AS INTEGER N = 1 S = 0 WHILE S <= 365 S = S + 36 N = N * 2 WEND PRINT N</pre>	<pre>var n, s: integer; begin n := 1; s := 0; while s <= 365 do begin s := s + 36; n := n * 2 end; write(n) end.</pre>

Си	Алгоритмический язык
<pre>#include<stdio.h> void main() { int n, s; n = 1; s = 0; while (s <= 365) { s = s + 36; n = n * 2; } printf("%d", n); }</pre>	<pre><u>алг</u> <u>нач</u> <u>цел</u> n, s n := 1 s := 0 <u>нц пока</u> s <= 365 s := s + 36 n := n * 2 <u>кц</u> <u>вывод</u> n <u>кон</u></pre>

Ответ: _____.

B6 Алгоритм вычисления значений функций $F(n)$ и $G(n)$, где n – натуральное число, задан следующими соотношениями:

$$F(1) = 1; G(1) = 1;$$

$$F(n) = F(n-1) - G(n-1), G(n) = F(n-1) + G(n-1), \text{ при } n > 2$$

Чему равно значение величины $G(5)/F(5)$?

В ответе запишите только натуральное число.

Ответ: _____.

B7 Решите уравнение:

$$60_8 + x = 60_9$$

Ответ запишите в шестеричной системе (основание системы счисления в ответе писать не нужно).

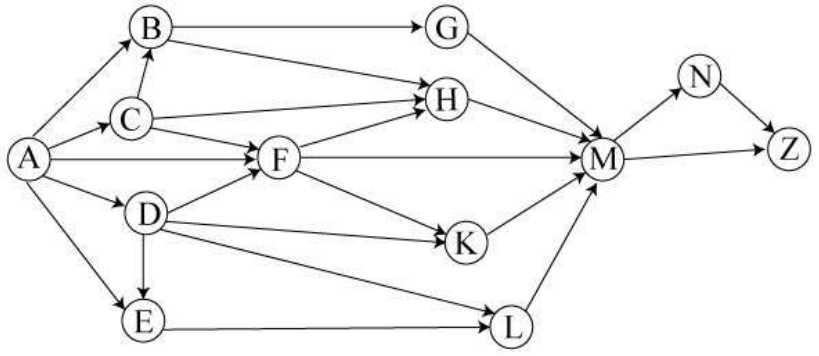
Ответ: _____.

B8 Ниже на четырёх языках записан алгоритм. Получив на вход число x , этот алгоритм печатает два числа: a и b . Укажите наименьшее из таких чисел x , при вводе которого алгоритм печатает сначала 3, а потом 4.

Бейсик	Паскаль
<pre>DIM X, A, B AS INTEGER INPUT X A=0: B=0 WHILE X > 0 A = A+1 IF B < (X MOD 8) THEN B = X MOD 8 END IF X = X \ 8 WEND PRINT A PRINT B</pre>	<pre>var x, a, b: integer; begin readln(x); a:=0; b:=0; while x>0 do begin a:=a + 1; if b < (x mod 8) then b:=x mod 8; x:=x div 8; end; writeln(a); write(b); end.</pre>
Си	Алгоритмический язык
<pre>#include<stdio.h> void main() { int x, a, b; scanf("%d", &x); a=0; b=0; while (x>0){ a = a+1; if (b < (x%8)){ b = x%8; } x = x/8; } printf("%d\n%d", a, b); }</pre>	<pre><u>алг</u> <u>нач</u> <u>цел</u> x, a, b <u>ввод</u> x a:=0; b:=0 <u>нц пока</u> x>0 a:=a+1 <u>если</u> b < mod(x,8) <u>то</u> b:=mod(x,8) <u>все</u> x:=div(x,8) <u>кц</u> <u>вывод</u> a, <u>нс</u>, b <u>кон</u></pre>

Ответ: _____.

B9 На рисунке – схема дорог, связывающих города А, В, С, D, E, F, G, H, K, L, M, N, Z. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город Z?



Ответ: _____.

B10 Документ объёмом 20 Мбайт можно передать с одного компьютера на другой двумя способами:

- A.** Сжать архиватором, передать архив по каналу связи, распаковать.
 - Б.** Передать по каналу связи без использования архиватора.
- Какой способ быстрее и на сколько, если
- средняя скорость передачи данных по каналу связи составляет 2^{23} 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. Например, если способ Б быстрее способа А на 23 секунды, в ответе нужно написать Б23.

Слов «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 заданному IP-адресу узла и маске.

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

IP-адрес узла: 224.37.249.37
 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	37	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, тыс.
Львов & (Ужгород Мукачево)	269
Львов & Мукачево	147
Львов & Ужгород & Мукачево	56

Какое количество страниц (в тысячах) будет найдено по запросу
 Львов & Ужгород

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. прибавь 4.

Первая из них увеличивает на 2 число на экране, вторая увеличивает это число на 4.

Программа для Удвоителя – это последовательность команд.

Сколько существует программ, которые число 4 преобразуют в число 22?

Ответ: _____.

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

Бейсик	Паскаль
<pre> DIM A, B, T, N, P AS INTEGER A = -20: B = 20 P = 130 N = 0 FOR T = A TO B IF F(T) > P THEN N = N+1 END IF NEXT T PRINT N FUNCTION F(x) F = 16*(9+x)*(9+x)+127 END FUNCTION </pre>	<pre> var a,b,t,N,P :integer; Function F(x: integer):integer; begin F := 16*(9+x)*(9+x)+127; end; BEGIN a := -20; b := 20; P := 130; N := 0; for t := a to b do begin if (F(t) > P) then begin N := N+1; end; end; write(N); END. </pre>
Си	Алгоритмический язык
<pre> #include<stdio.h> int F(int x) { return 16*(9+x)*(9+x)+127; } void main() { int a, b, t, N, P; a = -20; b = 20; P = 130; N = 0; for (t=a; t<=b; t++){ if (F(t) > P) { N++; } } printf("%d", N); } </pre>	<pre> алг нач цел a, b, t, P, N a := -20; b := 20 P := 130 N := 0 нц для t от a до b если F(t) > P то N := N+1 все кц вывод N кон алг цел F(цел x) нач знач := 16*(9+x)*(9+x)+127 кон </pre>

Ответ: _____.

B15 Сколько существует различных наборов значений логических переменных $x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4, z_1, z_2, z_3, z_4$, которые удовлетворяют всем перечисленным ниже условиям?

$$(z_1 \rightarrow z_2) \wedge (z_2 \rightarrow z_3) \wedge (z_3 \rightarrow z_4) = 1$$

$$(\neg x_1 \wedge y_1 \wedge z_1) \vee (x_1 \wedge \neg y_1 \wedge z_1) \vee (x_1 \wedge y_1 \wedge \neg z_1) = 1$$

$$(\neg x_2 \wedge y_2 \wedge z_2) \vee (x_2 \wedge \neg y_2 \wedge z_2) \vee (x_2 \wedge y_2 \wedge \neg z_2) = 1$$

$$(\neg x_3 \wedge y_3 \wedge z_3) \vee (x_3 \wedge \neg y_3 \wedge z_3) \vee (x_3 \wedge y_3 \wedge \neg z_3) = 1$$

$$(\neg x_4 \wedge y_4 \wedge z_4) \vee (x_4 \wedge \neg y_4 \wedge z_4) \vee (x_4 \wedge y_4 \wedge \neg z_4) = 1$$

В ответе не нужно перечислять все различные наборы значений переменных $x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4, z_1, z_2, z_3, z_4$, при которых выполнена данная система равенств. В качестве ответа Вам нужно указать количество таких наборов.

Ответ: _____.

Часть 3

Для записи ответов на задания этой части (C1–C4) используйте отдельный лист. Запишите сначала номер задания (C1, C2 и т. д.), а затем – полное решение. Ответы записывайте чётко и разборчиво.

C1 Требовалось написать программу, при выполнении которой с клавиатуры вводится натуральное число N (гарантируется, что $10 \leq N \leq 10^8$) и выводится двузначное число, образованное двумя его первыми (старшими) цифрами. Например, при $N = 2014$ надо вывести 20. Ученик написал такую программу:

Бейсик	Паскаль
<pre> DIM N AS LONG INPUT N WHILE N>100 N = N MOD 100 WEND PRINT N END </pre>	<pre> var n: longint; begin read(n); while n>100 do begin n := n mod 100 end; write(n); end. </pre>
Си	Алгоритмический язык
<pre> #include <stdio.h> void main(){ long int n; scanf("%ld",&n); while (n>100) { n = n % 100; } printf("%ld", n); } </pre>	<pre> алг нач цел n ввод n нц пока n>100 n := mod(n,100) кц вывод n кон </pre>

Последовательное выполнение следующей.

1. Напишите, что выведет эта программа при вводе числа 2014.
2. Приведите пример числа, при вводе которого программа выдаст верный ответ.
3. Найдите в программе все ошибки (их может быть одна или несколько). Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.

Обратите внимание: вам нужно исправить приведённую программу, а не написать свою. Вы можете только заменять ошибочные строки, но не можете удалять строки или добавлять новые. Заменять следует только ошибочные строки: за исправления, внесённые в строки, не содержащие ошибок, баллы будут снижаться.

C2

Дан массив, содержащий 2014 положительных целых чисел. Напишите на одном из языков программирования программу, которая находит в этом массиве количество элементов, значение которых более чем в два раза превосходит значение следующего элемента. Например, для массива из 6 элементов, содержащего числа 100, 32, 15, 10, 4, 2, программа должна выдать ответ 3 (условию соответствуют элементы со значениями 100, 32 и 10). Программа должна вывести общее количество подходящих элементов, значения элементов выводить не нужно. Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из описанных переменных.

Паскаль	Бейсик
<pre>const N=2014; var a: array [1..N] of integer; i, j, k: integer; begin for i:=1 to N do readln(a[i]); ... end.</pre>	<pre>N=2014 DIM A(N) AS INTEGER DIM I, J, K AS INTEGER FOR I = 1 TO N INPUT A(I) NEXT I ... END</pre>
Cп	Алгоритмический язык
<pre>#include <stdio.h> #define N 2014 void main(){ int a[N]; int i, j, k; for (i=0; i<N; i++) scanf("%d", &a[i]); ... }</pre>	<pre>алг нач цел N=2014 целтаб a[1:N] цел i, j, k 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). В этом случае вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии.

C3

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в кучу **один** или **два** камня или увеличить количество камней в куче в **три** раза. Например, имея кучу из 15 камней, за один ход можно получить кучу из 16, 17 или 45 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче становится не менее 75. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 75 или больше камней.

В начальный момент в куче было S камней, $1 \leq S \leq 74$.

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) При каких значениях числа S Петя может выиграть в один ход? Укажите все такие значения.

б) Укажите такое значение S , при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом. Опишите выигрышную стратегию Вани.

2. Укажите три таких значения S , при которых у Пети есть выигрышная стратегия, причём

– Петя не может выиграть за один ход, но

– Петя может выиграть своим вторым ходом, независимо от того, как будет ходить Ваня.

Для каждого из указанных значений S опишите выигрышную стратегию Пети.

3. Укажите значение S , при котором у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, однако у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом.

Для указанного значения S опишите выигрышную стратегию Вани. Постройте дерево всех партий, возможных при этой выигрышной стратегии Вани (в виде рисунка или таблицы). На рёбрах дерева указывайте, кто делает ход, в узлах – количество камней в позиции.

Дед Мороз и Снегурочка приходят на детские утренники с мешком конфет. Дед Мороз делит конфеты поровну между всеми присутствующими детьми (детей на утреннике никогда не бывает больше 100), а оставшиеся конфеты отдает Снегурочке. Снегурочка каждый раз записывает в блокнот количество полученных конфет. Если конфеты разделились между всеми детьми без остатка, Снегурочка ничего не получает и ничего не записывает. Когда утренники закончились, Деду Морозу стало интересно, сколько различных чисел встречается в записях Снегурочки. Дед Мороз и Снегурочка – волшебные, поэтому число утренников N , на которых они побывали, может быть очень большим.

Напишите программу, которая будет решать эту задачу. Перед текстом программы кратко опишите алгоритм решения задачи и укажите используемый язык программирования и его версию.

Желательно, чтобы программа была эффективной как по времени работы, так и по используемой памяти. Программу будем считать эффективной по памяти, если используемая память не зависит от размера входных данных (то есть числа утренников). Программу будем считать эффективной по времени, если при увеличении размера входных данных N в t раз (t – любое число) время её работы увеличивается не более чем в t раз.

Описание входных данных

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

Каждая из следующих N строк содержит два целых числа: сначала D – количество пришедших на очередной утренник детей, а затем K – количество конфет в мешке Деда Мороза на этом утреннике. Гарантируется выполнение следующих соотношений:

$$1 \leq N \leq 10000$$

$$1 \leq D \leq 100 \text{ (для каждого } D\text{)}$$

$$D \leq K \leq 1000 \text{ (для каждой пары } D, K\text{)}$$

Описание выходных данных

Программа должна вывести одно число – количество различных чисел в записях Снегурочки. Если Снегурочка ни разу ничего не записывала, надо вывести ноль.

Пример входных данных:

```
7
10 58
15 315
20 408
100 1000
32 63
32 63
11 121
```

Пример выходных данных для приведённого выше примера входных данных:

2

Ответы к заданиям с выбором ответа

№ задания	Ответ
A1	3
A2	2
A3	1
A4	3
A5	1
A6	2
A7	2

№ задания	Ответ
A8	2
A9	2
A10	3
A11	2
A12	4
A13	2

Ответы к заданиям с кратким ответом

№ задания	Ответ
B1	12221
B2	85
B3	24
B4	243
B5	1024
B6	1
B7	10
B8	66

№ задания	Ответ
B9	36
B10	B13
B11	DFBH
B12	193
B13	89
B14	50
B15	31

Ответы к заданиям с выбором ответа

№ задания	Ответ
A1	3
A2	2
A3	4
A4	3
A5	2
A6	2
A7	3

№ задания	Ответ
A8	4
A9	3
A10	2
A11	3
A12	4
A13	4

Ответы к заданиям с кратким ответом

№ задания	Ответ
B1	21221
B2	90
B3	75
B4	729
B5	2048
B6	1
B7	10
B8	68

№ задания	Ответ
B9	36
B10	B4
B11	DECH
B12	178
B13	55
B14	40
B15	31

C1 Требовалось написать программу, при выполнении которой с клавиатуры вводится натуральное число, не превосходящее 10^8 , и выводится его первая (старшая) цифра. Ученик написал такую программу:

Бейсик	Паскаль
<pre>DIM N AS LONG INPUT N WHILE N>10 N = N MOD 10 WEND PRINT N END</pre>	<pre>var n: longint; begin read(n); while n>10 do begin n := n mod 10 end; write(n); end.</pre>
Си	Алгоритмический язык
<pre>#include <stdio.h> void main(){ long int n; scanf("%ld",&n); while (n>10) { n = n%10; } printf("%ld", n); }</pre>	<pre><u>алг</u> нач <u>цел</u> n <u>ввод</u> n <u>нц пока</u> n>10 n := mod(n,10) <u>кц</u> <u>вывод</u> n <u>кон</u></pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 1984.
2. Приведите пример числа, при вводе которого программа выдаст верный ответ.
3. Найдите в программе все ошибки (их может быть одна или несколько). Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.

Обратите внимание: вам нужно исправить приведённую программу, а не написать свою. Вы можете только заменять ошибочные строки, но не можете удалять строки или добавлять новые. Заменять следует только ошибочные строки: за исправления, внесённые в строки, не содержащие ошибок, баллы будут снижаться.

Содержание верного ответа

(допускаются иные формулировки ответа, не искажающие его смысла)

1. При вводе числа 1984 программа выведет число 4.
Комментарий для экспертов. Приведённая программа выводит ответ 10 для $n = 10$ и последнюю цифру для любого другого значения n .
2. Пример числа, для которого программа даёт верный ответ: 2012.
Комментарий для экспертов. Программа даст верный ответ для любого числа, у которого совпадают первая и последняя цифры. В частности, для любого однозначного числа.
3. Ошибки содержатся в двух строках программы:
 - 1) Неверное условие цикла: неравенство должно быть нестрогим, иначе можно в качестве ответа получить 10.
 - 2) Неверная операция отбрасывания последней цифры: вместо нахождения остатка нужно использовать целочисленное деление.**Пример исправления для языка Паскаль:**
 Первая строка с ошибкой:

```
while n>10 do begin
```

 Исправленная строка:

```
while n>=10 do begin
```

 Другой способ исправления:

```
while n>9 do begin
```

 Вторая строка с ошибкой:

```
n := n mod 10
```

 Исправленная строка:

```
n := n div 10
```

 В программах на других языках ошибочные строки и их исправления аналогичны.
 Допустимы избыточные скобки, не изменяющие правильный порядок действий. Незначительной опiskой, не влияющей на оценку, следует считать отсутствие при исправлении первой ошибки слов `do begin` в программе на Паскале и фигурной скобки в программе на Си.

Указания по оцениванию	Баллы
<p>В задаче требуется выполнить три действия.</p> <p>1. Указать результат программы при данном вводе. Это действие считается выполненным, если указан верный результат работы программы при заданных входных данных. Экзаменуемый не обязан объяснять, как получен этот результат, достаточно указать верное число.</p> <p>2. Указать пример ввода, при котором программа выводит верный ответ. Это действие считается выполненным, если указан пример числа, при котором программа выдаёт верный ответ. Экзаменуемый не обязан описывать все ситуации, в которых программа выдаёт верный ответ, ему достаточно указать пример ввода, при котором это происходит. Если экзаменуемый приводит несколько примеров, действие считается выполненным только в том случае, если программа выдаёт верный ответ для всех приведённых примеров.</p> <p>3. Найти и исправить ошибки в программе. Это действие считается выполненным, если верно указаны обе строки с ошибкой и предложены верные варианты исправления; при этом никакие строки, не содержащие ошибок, не указаны в качестве строк, требующих внесения исправлений. В исправленной строке допускаются незначительные синтаксические ошибки (лишние или пропущенные знаки препинания, неточные написания служебных слов языка).</p>	
<p>Правильно выполнены все действия:</p> <p>1) указан верный результат для приведённого примера входных данных;</p> <p>2) дан пример числа, для которого программа с ошибками выдаёт тот же результат, что и правильная программа;</p> <p>3) указаны и исправлены две ошибки в программе;</p> <p>4) не указаны в качестве ошибочных никакие другие строки программы.</p>	3

<p>Не выполнены условия, позволяющие поставить 3 балла, и имеет место один из следующих случаев:</p> <ol style="list-style-type: none"> 1. Выполнены два первых действия (верный результат при указанных данных, верный пример числа, для которого программа выдаёт правильный результат), найдена и исправлена одна ошибка в программе, ни одна верная строка не названа ошибочной. 2. Выполнены два первых действия (верный результат при указанных данных, верный пример числа, для которого программа выдаёт правильный результат), найдены и исправлены две ошибки в программе, одна верная строка названа ошибочной. 3. Выполнено одно из первых двух действий (верный результат при указанных данных или верный пример числа, для которого программа выдаёт правильный результат), найдены и исправлены две ошибки в программе, ни одна верная строка не названа ошибочной. 	2
<p>Не выполнены условия, позволяющие поставить 2 или 3 балла, и имеет место один из следующих случаев:</p> <ol style="list-style-type: none"> 1. Выполнены два первых действия (верный результат при указанных данных, верный пример числа, для которого программа выдаёт правильный результат). При этом не существенно, насколько правильно выполнено третье действие. 2. Найдены и исправлены две ошибки в программе, не более чем одна верная строка названа ошибочной. При этом не существенно, насколько правильно выполнены первое и второе действия. 3. Найдена и исправлена одна ошибка в программе, ни одна верная строка не названа ошибочной. При этом не существенно, насколько правильно выполнены первое и второе действия. 	1
<p>Не выполнены условия, позволяющие поставить 1, 2 или 3 балла.</p>	0
<p><i>Максимальный балл</i></p>	3

C2

Дан массив, содержащий 2014 положительных целых чисел. Напишите на одном из языков программирования программу, которая находит в этом массиве количество элементов, значение которых более чем в два раза превосходит значение предшествующего элемента. Например, для массива из 6 элементов, содержащего числа 2, 5, 10, 15, 40, 100, программа должна выдать ответ 3 (условию соответствуют элементы со значениями 5, 40 и 100). Программа должна вывести общее количество подходящих элементов, значения элементов выводить не нужно. Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из описанных переменных.

Паскаль	Бейсик
<pre>const N=2014; var a: array [1..N] of integer; i, j, k: integer; begin for i:=1 to N do readln(a[i]); ... end.</pre>	<pre>N=2014 DIM A(N) AS INTEGER DIM I, J, K AS INTEGER FOR I = 1 TO N INPUT A(I) NEXT I ... END</pre>
Си	Алгоритмический язык
<pre>#include <stdio.h> #define N 2014 void main(){ int a[N]; int i, j, k; for (i=0; i<N; i++) scanf("%d", &a[i]); ... }</pre>	<pre><u>алг</u> <u>нач</u> цел N=2014 <u>целтаб</u> a[1:N] цел i, j, k нц для i от 1 до N <u>ввод</u> a[i] <u>кц</u> ... <u>кон</u></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). В этом случае вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии.

Содержание верного ответа

(допускаются иные формулировки ответа, не искажающие его смысла)

Программа просматривает все пары соседних чисел в массиве и подсчитывает количество таких пар, в которых второй элемент более чем вдвое превышает первый.

Пример фрагмента программы на Паскале

```
k:=0;
for i:=2 to N do begin
  if a[i] > 2*a[i-1] then k:=k+1;
end;
writeln(k);
```

При сравнении элементов можно вместо умножения использовать деление, но при этом следует позаботиться о том, чтобы при делении получался вещественный результат, а не округлённое целое значение. Можно, например, записать условие проверки так:

```
if a[i] / a[i-1] > 2
```

Следующая запись недопустима, так как при таком делении происходит округление до меньшего целого, в результате, например, пара (2, 5) не будет определена как допустимая:

```
if a[i] div a[i-1] > 2
```

Если проверка выполняется с помощью деления, следует быть особенно внимательным при использовании языка Си: в этом языке нет специальной операции целочисленного деления, деление целых чисел с помощью операции/по умолчанию выполняется как целочисленное.

Замена в предыдущем неверном решении строгого сравнения на нестрогое не делает программу правильной, так как в этом случае будет, например, ошибочно учтена не соответствующая условиям пара (5, 10).

Ещё один пример неверного решения (приводится только тело цикла):

```
j := a[i] / a[i-1]
if j > 2 then k:=k+1;
```

Здесь целому числу присваивается вещественное значение. Такое присваивание либо (в зависимости от языка программирования) приведёт к ошибке трансляции, либо даст неверный результат при выполнении, как при целочисленном делении.

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 языках, использованных в задании. 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
Предложено в целом верное решение, содержащее не более одной ошибки из числа следующих (если одинаковая ошибка повторяется несколько раз, она считается за одну ошибку): 1) Отсутствие инициализации или неверная инициализация счётчика. 2) Неверно определены границы цикла проверки, в результате проверяются не все пары или происходит выход за границы массива. 3) При вычислении отношения элементов используется целочисленное деление. 4) Подсчитываются пары, в которых первый элемент больше второго 5) Отсутствует вывод ответа. 6) Используется переменная, не объявленная в разделе описания переменных. 7) Индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно.	1
Не выполнены условия, позволяющие поставить 1 или 2 балла.	0
<i>Максимальный балл</i>	2

С3

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в кучу **один** или **два** камня или увеличить количество камней в куче в **три** раза. Например, имея кучу из 15 камней, за один ход можно получить кучу из 16, 17 или 45 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче становится не менее 64. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 64 или больше камней.

В начальный момент в куче было S камней, $1 \leq S \leq 63$.

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) При каких значениях числа S Петя может выиграть в один ход? Укажите все такие значения.

б) Укажите такое значение S , при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом. Опишите выигрышную стратегию Вани.

2. Укажите три таких значения S , при которых у Пети есть выигрышная стратегия, причём

– Петя не может выиграть за один ход, но

– Петя может выиграть своим вторым ходом, независимо от того, как будет ходить Ваня.

Для каждого из указанных значений S опишите выигрышную стратегию Пети.

3. Укажите значение S , при котором у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, однако у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом.

Для указанного значения S опишите выигрышную стратегию Вани. Постройте дерево всех партий, возможных при этой выигрышной стратегии Вани (в виде рисунка или таблицы). На рёбрах дерева указывайте, кто делает ход, в узлах – количество камней в позиции.

Содержание верного ответа

(допускаются иные формулировки ответа, не искажающие его смысла)

- а) Петя может выиграть, если $S = 22, \dots, 63$. При меньших значениях S за один ход нельзя получить кучу, в которой больше 63 камней.
 б) Ваня может выиграть первым ходом (как бы ни играл Петя), если исходно в куче будет $S = 21$ камень. Тогда после первого хода Пети в куче будет 22 камня, или 23 камня, или 63 камня. Во всех случаях Ваня утраивает количество камней и выигрывает первым ходом.
- Возможные значения S : 19, 20, 7. В этих случаях Петя, очевидно, не может выиграть первым ходом. Однако он может получить кучу из 21 камня (при $S = 19$ нужно добавить 2 камня, при $S = 20$ нужно добавить 1 камень, при $S = 7$ нужно утроить количество камней). Ситуация, когда в куче 21 камень, разобрана в п. 1 б. В ней игрок, который будет ходить (теперь это Ваня), выиграть не может, а его противник (то есть Петя) следующим ходом выигрывает.

Исходное положение	Положения после очередных ходов			
	1-й ход Пети (разобраны все ходы)	1-й ход Вани (только ход по стратегии)	2-й ход Пети (разобраны все ходы)	2-й ход Вани (только ход по стратегии)
18	18+1=19	19+2=21	21 + 1 = 22	22 * 3 = 66
			21 + 2 = 23	23 * 3 = 69
			21 * 3 = 63	63 * 3 = 189
	18+2=20	20+1=21	21 + 1 = 22	22 * 3 = 66
			21 + 2 = 23	23 * 3 = 69
			21 * 3 = 63	63 * 3 = 189
	18*3=54	54*3=162		

- Возможное значение S : 18. После первого хода Пети в куче будет 19, 20 или 54 камня. Если в куче станет 54 камня, Ваня утроит количество камней и выигрывает первым ходом. Ситуации, когда в куче 19 или 20 камней, уже разобраны в п. 2. В этих ситуациях игрок, который будет ходить (теперь это Ваня), выигрывает своим вторым ходом.

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

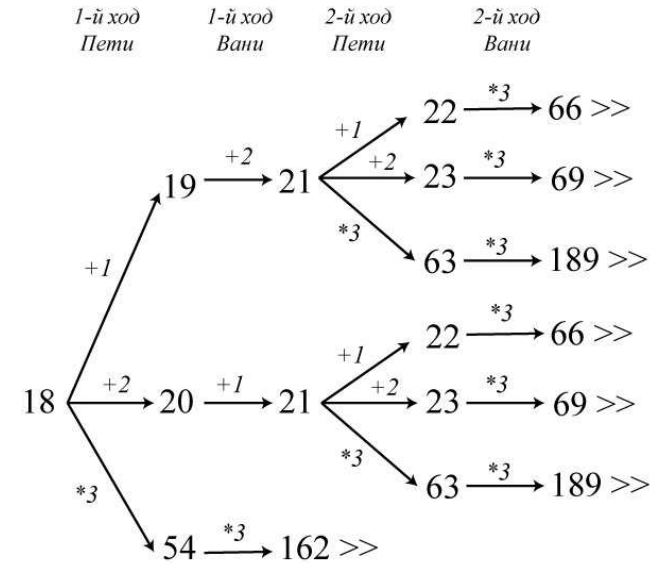


Рис.1. Дерево всех партий, возможных при Ваниной стратегии. Знаком >> обозначены позиции, в которых партия заканчивается.

Указания по оцениванию	Баллы
<p>В задаче от ученика требуется выполнить 3 задания. Их трудность возрастает. Количество баллов в целом соответствует количеству выполненных заданий (подробнее см. ниже).</p> <p>Ошибка в решении, не искажающая основного замысла и не приведшая к неверному ответу, например, арифметическая ошибка при вычислении количества камней в заключительной позиции, при оценке решения не учитывается.</p> <p>Первое задание считается выполненным полностью, если выполнены полностью оба пункта а) и б). Пункт а) считается выполненным полностью, если правильно указаны все позиции, в которых Петя выигрывает первым ходом, и указано, каким должен быть первый ход. Пункт б) считается выполненным полностью, если правильно указана позиция, в которой Ваня выигрывает первым ходом и описана стратегия Вани, т. е. показано, как Ваня может получить кучу, в которой содержится нужное количество камней, при любом ходе Пети.</p> <p>Первое задание считается выполненным частично, если (1) правильно указаны все позиции, в которых Петя выигрывает первым ходом, (2) правильно указана позиция, в которой Ваня выигрывает первым ходом, и (3) явно сказано, что при любом ходе Пети Ваня может получить кучу, которая содержит нужное для выигрыша количество камней. При этом может быть не указано, каким именно ходом выигрывает Ваня.</p> <p>Второе задание выполнено, если правильно указаны три позиции, выигрышные для Пети, и описана соответствующая стратегия Пети – так, как это написано в примере решения, или другим способом, например, с помощью дерева всех партий, возможных при выбранной стратегии Пети. Допускается также ситуация, когда указаны все три выигрышные позиции, и явно сказано, что из любой из них Петя может получить позицию, в которой второй игрок выигрывает своим первым ходом (позиция разобрана в п.1 б).</p> <p>Третье задание выполнено, если правильно указана позиция, выигрышная для Вани, и построено дерево всех партий, возможных при Ваниной стратегии. При изображении дерева для каждой позиции, где должен ходить Петя, должны быть показаны все возможные ходы, а для позиций, где должен ходить Ваня, – только ход, соответствующий выбранной Ваней стратегии.</p>	

ВНИМАНИЕ! При игре по Ваниной стратегии в некоторые состояния кучи можно прийти несколькими путями. Это позволяет «склеить» такие вершины в дереве возможных партий и представлять множество всех возможных партий ориентированным ациклическим графом, который не является деревом. Вместо дерева на рис. 1 можно использовать, например, граф, изображённый на рис. 2. **Это не является ошибкой.**

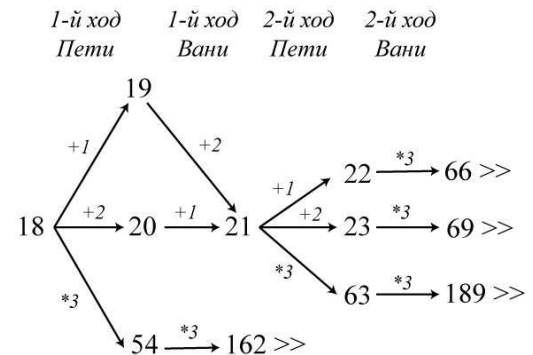


Рис. 2. Граф, представляющий множество всех партий, возможных при Ваниной стратегии. Знаком >> обозначены позиции, в которых партия заканчивается.

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

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

Дед Мороз и Снегурочка приходят на детские утренники с мешком конфет. Дед Мороз делит конфеты поровну между всеми присутствующими детьми (детей на утреннике никогда не бывает больше 100), а оставшиеся конфеты отдает Снегурочке. Снегурочка каждый раз записывает в блокнот количество полученных конфет. Если конфеты разделились между всеми детьми без остатка, Снегурочка ничего не получает и ничего не записывает. Когда утренники закончились, Деду Морозу стало интересно, какое число чаще всего записывала Снегурочка. Дед Мороз и Снегурочка – волшебные, поэтому число утренников N , на которых они побывали, может быть очень большим.

Напишите программу, которая будет решать эту задачу. Перед текстом программы кратко опишите алгоритм решения задачи и укажите используемый язык программирования и его версию.

Желательно, чтобы программа была эффективной как по времени работы, так и по используемой памяти. Программу будем считать эффективной по памяти, если используемая память не зависит от размера входных данных (то есть числа утренников). Программу будем считать эффективной по времени, если при увеличении размера входных данных N в t раз (t – любое число) время её работы увеличивается не более чем в t раз.

Описание входных данных

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

Каждая из следующих N строк содержит два целых числа: сначала D – количество пришедших на очередной утренник детей, а затем K – количество конфет в мешке Деда Мороза на этом утреннике. Гарантируется выполнение следующих соотношений:

$$1 \leq N \leq 10000$$

$$1 \leq D \leq 100 \text{ (для каждого } D)$$

$$D \leq K \leq 1000 \text{ (для каждой пары } D, K)$$

Описание выходных данных

Программа должна вывести одно число – то, которое Снегурочка записывала чаще всего. Если несколько чисел записывались одинаково часто, надо вывести большее из них. Если Снегурочка ни разу ничего не записывала, надо вывести ноль.

Пример входных данных:

```
7
10 58
15 315
20 408
100 1000
32 63
32 63
11 121
```

Пример выходных данных для приведённого выше примера входных данных:

31

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

Поскольку количество детей не превышает 100, остаться может не больше 99 конфет. Можно создать массив из 99 элементов и использовать их в качестве счётчиков, хранящих информацию о количестве записей каждого числа.

Программа читает исходные данные, не запоминая их в массиве. Для каждой пары (D, K) количество оставшихся конфет определяется как остаток от деления K на D . Если этот остаток положителен, надо увеличить соответствующий счётчик. По окончании ввода и обработки данных надо найти индекс максимального элемента в массиве счётчиков. При этом нужно правильно (в соответствии с условием) обработать ситуации равенства максимумов и отсутствия ненулевых счётчиков.

Пример правильной и эффективной программы на языке Паскаль

```
program c4;
const DMAX=100;      {максимально возможное количество детей}
var
  N: integer;        {количество утренников}
  D: integer;        {количество детей на утреннике}
  K: integer;        {количество конфет}
  r: integer;        {остаток}
  c: array[1..DMAX-1] of integer; {счетчики остатков}
  i: integer;
  imax: integer;
begin
  {предварительная очистка счетчиков}
  for i:=1 to DMAX-1 do c[i]:=0;
  readln(N);
  {ввод данных, подсчет количества каждого остатка}
  for i:=1 to N do begin
    readln(D, K);
    r := K mod D;
    if r>0 then c[r]:=c[r]+1;
  end;
  {выбор самого частого остатка}
  imax:=1;
  for i:=2 to DMAX-1 do begin
    if c[i]>=c[imax] then imax:=i;
  end;
  if c[imax]=0 then imax:=0;
  writeln(imax);
end.
```

Указания по оцениванию	Баллы
<p>Программа правильно работает для любых входных данных и является эффективной как по времени, так и по памяти. Допускается наличие в тексте программы одной синтаксической ошибки: пропущен или неверно указан знак пунктуации, неверно написано или пропущено зарезервированное слово языка программирования, не описана или неверно описана переменная, применяется операция, недопустимая для соответствующего типа данных (если одна и та же ошибка встречается несколько раз, то это считается за одну ошибку).</p>	4
<p>Не выполнены условия, позволяющие поставить 4 балла. Программа работает верно и является эффективной по времени, но размер используемой памяти зависит от количества исходных данных. Например, входные данные (все значения D и K) запоминаются в массиве или другой структуре данных, размер которой зависит от N.</p> <p>Допускается одна из следующих ошибок:</p> <ol style="list-style-type: none"> 1) Вместо остатка от деления K на D вычисляется остаток от деления D на K. 2) Отсутствует предварительная инициализация счётчиков (для языков, в которых нет гарантированного начального обнуления). 3) Неверно разрешается конфликт равенства максимумов (выбирается меньшее значение индекса вместо большего). 4) Неверно выводится ответ (или не выводится никакого ответа), если все остатки оказались равны нулю. <p>Допускается наличие от одной до трёх синтаксических ошибок, описанных в критериях на 4 балла.</p>	3
<p>Не выполнены условия, позволяющие поставить 3 или 4 балла. Программа работает в целом верно, эффективно или нет. Возможны переборные решения, при которых все исходные данные хранятся в массиве (или в двух массивах), этот массив многократно просматривается, при каждом просмотре подсчитывается количество пар с определённым остатком. В реализации алгоритма допущено более 1 ошибки из числа перечисленных в критериях на 3 балла или допущены другие ошибки, приводящие к неверной работе программы в отдельных случаях.</p> <p>Допускается наличие от одной до пяти синтаксических ошибок, описанных в критериях на 4 балла.</p>	2

<p>Не выполнены условия, позволяющие поставить 2, 3 или 4 балла. 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.</p>	1
<p>Не выполнены условия, позволяющие поставить 1, 2, 3 или 4 балла.</p>	0
<i>Максимальный балл</i>	4

С1 Требовалось написать программу, при выполнении которой с клавиатуры вводится натуральное число N (гарантируется, что $10 \leq N \leq 10^8$) и выводится двузначное число, образованное двумя его первыми (старшими) цифрами. Например, при $N = 2014$ надо вывести 20. Ученик написал такую программу:

Бейсик	Паскаль
<pre>DIM N AS LONG INPUT N WHILE N>100 N = N MOD 100 WEND PRINT N END</pre>	<pre>var n: longint; begin read(n); while n>100 do begin n := n mod 100 end; write(n); end.</pre>
Си	Алгоритмический язык
<pre>#include <stdio.h> void main(){ long int n; scanf("%ld",&n); while (n>100) { n = n % 100; } printf("%ld", n); }</pre>	<pre><u>алг</u> нач <u>цел</u> n <u>ввод</u> n нц пока n>100 n := mod(n,100) <u>кц</u> <u>вывод</u> n кон</pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 2014.
2. Приведите пример числа, при вводе которого программа выдаст верный ответ.
3. Найдите в программе все ошибки (их может быть одна или несколько). Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.

Обратите внимание: вам нужно исправить приведённую программу, а не написать свою. Вы можете только заменять ошибочные строки, но не можете удалять строки или добавлять новые. Заменять следует только ошибочные строки: за исправления, внесённые в строки, не содержащие ошибок, баллы будут снижаться.

Содержание верного ответа

(допускаются иные формулировки ответа, не искажающие его смысла)

1. При вводе числа 2014 программа выведет число 14.

Комментарий для экспертов. Приведённая программа выводит ответ 100 для $N = 100$ и число, образованное двумя последними цифрами, для любого другого значения n .

2. Пример числа, для которого программа даёт верный ответ: 2020.

Комментарий для экспертов. Программа даст верный ответ для любого числа, у которого общее количество цифр чётно, при этом две первые цифры совпадают с двумя последними, в частности, для любого двузначного числа.

3. Ошибки содержатся в двух строках программы:

1) Неверное условие цикла: неравенство должно быть нестрогим, иначе можно в качестве ответа получить 100.

2) Неверно реализовано отбрасывание цифр. Во-первых, вместо нахождения остатка нужно использовать целочисленное деление. Во-вторых, делить надо не на 100, а на 10, то есть отбрасывать цифры не по две, а по одной, иначе при нечётном количестве цифр в числе получится неверный ответ.

Пример исправления для языка Паскаль:

Первая строка с ошибкой:

```
while n>100 do begin
```

Исправленная строка:

```
while n>=100 do begin
```

Другой способ исправления:

```
while n>99 do begin
```

Вторая строка с ошибкой:

```
n := n mod 100
```

Исправленная строка:

```
n := n div 10
```

В программах на других языках ошибочные строки и их исправления аналогичны.

Допустимы избыточные скобки, не изменяющие правильный порядок действий. Незначительной опиской, не влияющей на оценку, следует считать отсутствие при исправлении первой ошибки слов `do begin` в программе на Паскале и фигурной скобки в программе на Си.

Указания по оцениванию	Баллы
<p>В задаче требуется выполнить три действия.</p> <p>1. Указать результат программы при данном вводе. Это действие считается выполненным, если указан верный результат работы программы при заданных входных данных. Экзаменуемый не обязан объяснять, как получен этот результат, достаточно указать верное число.</p> <p>2. Указать пример ввода, при котором программа выводит верный ответ. Это действие считается выполненным, если указан пример числа, при котором программа выдаёт верный ответ. Экзаменуемый не обязан описывать все ситуации, в которых программа выдаёт верный ответ, ему достаточно указать пример ввода, при котором это происходит. Если экзаменуемый приводит несколько примеров, действие считается выполненным только в том случае, если программа выдаёт верный ответ для всех приведённых примеров.</p> <p>3. Найти и исправить ошибки в программе. Это действие считается выполненным, если верно указаны обе строки с ошибкой и предложены верные варианты исправления; при этом никакие строки, не содержащие ошибок, не указаны в качестве строк, требующих внесения исправлений. В исправленной строке допускаются незначительные синтаксические ошибки (лишние или пропущенные знаки препинания, неточные написания служебных слов языка).</p>	
<p>Правильно выполнены все действия:</p> <p>1) указан верный результат для приведённого примера входных данных;</p> <p>2) дан пример числа, для которого программа с ошибками выдаёт тот же результат, что и правильная программа;</p> <p>3) указаны и исправлены две ошибочные строки в программе;</p> <p>4) не указаны в качестве ошибочных никакие другие строки программы.</p>	3

<p>Не выполнены условия, позволяющие поставить 3 балла, и имеет место один из следующих случаев:</p> <p>1) Выполнены два первых действия (верный результат при указанных данных, верный пример числа, для которого программа выдаёт правильный результат), найдена и исправлена одна ошибка в программе, ни одна верная строка не названа ошибочной.</p> <p>2) Выполнены два первых действия (верный результат при указанных данных, верный пример числа, для которого программа выдаёт правильный результат), найдены и исправлены две ошибки в программе, одна верная строка названа ошибочной.</p> <p>3) Выполнено одно из первых двух действий (верный результат при указанных данных или верный пример числа, для которого программа выдаёт правильный результат), найдены и исправлены две ошибки в программе, ни одна верная строка не названа ошибочной.</p>	2
<p>Не выполнены условия, позволяющие поставить 2 или 3 балла, и имеет место один из следующих случаев:</p> <p>1. Выполнены два первых действия (верный результат при указанных данных, верный пример числа, для которого программа выдаёт правильный результат). При этом не существенно, насколько правильно выполнено третье действие.</p> <p>2. Найдены и исправлены две ошибки в программе, не более чем одна верная строка названа ошибочной. При этом не существенно, насколько правильно выполнены первое и второе действия.</p> <p>3. Найдена и исправлена одна ошибка в программе, ни одна верная строка не названа ошибочной. При этом не существенно, насколько правильно выполнены первое и второе действия.</p>	1
<p>Не выполнены условия, позволяющие поставить 1, 2 или 3 балла.</p>	0
<i>Максимальный балл</i>	3

Дан массив, содержащий 2014 положительных целых чисел. Напишите на одном из языков программирования программу, которая находит в этом массиве количество элементов, значение которых более чем в два раза превосходит значение следующего элемента. Например, для массива из 6 элементов, содержащего числа 100, 32, 15, 10, 4, 2, программа должна выдать ответ 3 (условию соответствуют элементы со значениями 100, 32 и 10). Программа должна вывести общее количество подходящих элементов, значения элементов выводить не нужно. Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из описанных переменных.

Паскаль	Бейсик
<pre>const N=2014; var a: array [1..N] of integer; i, j, k: integer; begin for i:=1 to N do readln(a[i]); ... end.</pre>	<pre>N=2014 DIM A(N) AS INTEGER DIM I, J, K AS INTEGER FOR I = 1 TO N INPUT A(I) NEXT I ... END</pre>
Си	Алгоритмический язык
<pre>#include <stdio.h> #define N 2014 void main(){ int a[N]; int i, j, k; for (i=0; i<N; i++) scanf("%d", &a[i]); ... }</pre>	<pre><u>алг</u> <u>нач</u> <u>цел</u> N=2014 <u>целтаб</u> a[1:N] <u>цел</u> i, j, k <u>нц</u> <u>для</u> i <u>от</u> 1 <u>до</u> N <u>ввод</u> a[i] <u>кц</u> ... <u>кон</u></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). В этом случае вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии.

Содержание верного ответа

(допускаются иные формулировки ответа, не искажающие его смысла)

Программа просматривает все пары соседних чисел в массиве и подсчитывает количество таких пар, в которых первый элемент более чем вдвое превышает второй.

Пример фрагмента программы на Паскале

```
k:=0;
for i:=1 to N-1 do begin
  if a[i] > 2*a[i+1] then k:=k+1;
end;
writeln(k);
```

При сравнении элементов можно вместо умножения использовать деление, но при этом следует позаботиться о том, чтобы при делении получался вещественный результат, а не округлённое целое значение. Можно, например, записать условие проверки так:

```
if a[i] / a[i+1] > 2
```

Следующая запись недопустима, так как при таком делении происходит округление до меньшего целого, в результате, например, пара (32, 15) не будет определена как допустимая:

```
if a[i] div a[i+1] > 2
```

Если проверка выполняется с помощью деления, следует быть особенно внимательным при использовании языка Си: в этом языке нет специальной операции целочисленного деления, деление целых чисел с помощью операции/по умолчанию выполняется как целочисленное.

Замена в предыдущем неверном решении строгого сравнения на нестрогое не делает программу правильной, так как в этом случае будет, например, ошибочно учтена не соответствующая условиям пара (4, 2).

Ещё один пример неверного решения (приводится только тело цикла):

```
j := a[i] / a[i+1]
if j > 2 then k:=k+1;
```

Здесь целому числу присваивается вещественное значение. Такое присваивание либо (в зависимости от языка программирования) приведёт к ошибке трансляции, либо даст неверный результат при выполнении, как при целочисленном делении.

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 языках, использованных в задании. 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
Предложено в целом верное решение, содержащее не более одной ошибки из числа следующих (если одинаковая ошибка повторяется несколько раз, она считается за одну ошибку): 1) Отсутствие инициализации или неверная инициализация счётчика. 2) Неверно определены границы цикла проверки, в результате проверяются не все пары или происходит выход за границы массива. 3) При вычислении отношения элементов используется целочисленное деление. 4) Подсчитываются пары, в которых второй элемент больше первого. 5) Отсутствует вывод ответа. 6) Используется переменная, не объявленная в разделе описания переменных. 7) Индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно.	1
Не выполнены условия, позволяющие поставить 1 или 2 балла.	0
<i>Максимальный балл</i>	2

С3

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в кучу **один** или **два** камня или увеличить количество камней в куче в **три** раза. Например, имея кучу из 15 камней, за один ход можно получить кучу из 16, 17 или 45 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче становится не менее 75. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 75 или больше камней.

В начальный момент в куче было S камней, $1 \leq S \leq 74$.

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) При каких значениях числа S Петя может выиграть в один ход?

Укажите все такие значения.

б) Укажите такое значение S , при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом. Опишите выигрышную стратегию Вани.

2. Укажите три таких значения S , при которых у Пети есть выигрышная стратегия, причём

– Петя не может выиграть за один ход, но

– Петя может выиграть своим вторым ходом, независимо от того, как будет ходить Ваня.

Для каждого из указанных значений S опишите выигрышную стратегию Пети.

3. Укажите значение S , при котором у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, однако у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом.

Для указанного значения S опишите выигрышную стратегию Вани. Постройте дерево всех партий, возможных при этой выигрышной стратегии Вани (в виде рисунка или таблицы). На рёбрах дерева указывайте, кто делает ход, в узлах – количество камней в позиции.

Содержание верного ответа

(допускаются иные формулировки ответа, не искажающие его смысла)

1. а) Петя может выиграть, если $S = 25, \dots, 74$. При меньших значениях S за один ход нельзя получить кучу, в которой больше 74 камней.

б) Ваня может выиграть первым ходом (как бы ни играл Петя), если исходно в куче будет $S = 24$ камня. Тогда после первого хода Пети в куче будет 25 камней, или 26 камней, или 72 камня. Во всех случаях Ваня утраивает количество камней и выигрывает первым ходом.

2. Возможные значения S : 22, 23, 8. В этих случаях Петя, очевидно, не может выиграть первым ходом. Однако он может получить кучу из 24 камней (при $S = 22$ нужно добавить 2 камня, при $S = 23$ нужно добавить 1 камень, при $S = 8$ нужно утроить количество камней). Ситуация, когда в куче 24 камня, разобрана в п. 1 б. В ней игрок, который будет ходить (теперь это Ваня), выиграть не может, а его противник (то есть Петя) следующим ходом выигрывает.

Исходное положение	Положения после очередных ходов			
	1-й ход Пети (разобраны все ходы)	1-й ход Вани (только ход по стратегии)	2-й ход Пети (разобраны все ходы)	2-й ход Вани (только ход по стратегии)
21	$21 + 1 = 22$	$22 + 2 = 24$	$24 + 1 = 25$	$25 * 3 = 75$
			$24 + 2 = 26$	$26 * 3 = 78$
			$24 * 3 = 72$	$72 * 3 = 216$
	$21 + 2 = 23$	$23 + 1 = 24$	$24 + 1 = 25$	$25 * 3 = 75$
			$24 + 2 = 26$	$26 * 3 = 78$
			$24 * 3 = 72$	$72 * 3 = 216$
$21 * 3 = 63$	$63 * 3 = 189$			

3. Возможное значение S : 21. После первого хода Пети в куче будет 22, 23 или 63 камня. Если в куче станет 63 камня, Ваня утроит количество камней и выигрывает первым ходом. Ситуации, когда в куче 22 или 23 камня уже разобраны в п. 2. В этих ситуациях игрок, который будет ходить (теперь это Ваня), выигрывает своим вторым ходом.

В таблице изображено дерево возможных партий при описанной

стратегии Вани. Заключительные позиции (в них выигрывает Ваня) подчёркнуты. На рисунке это же дерево изображено в графическом виде (оба способа изображения дерева допустимы).

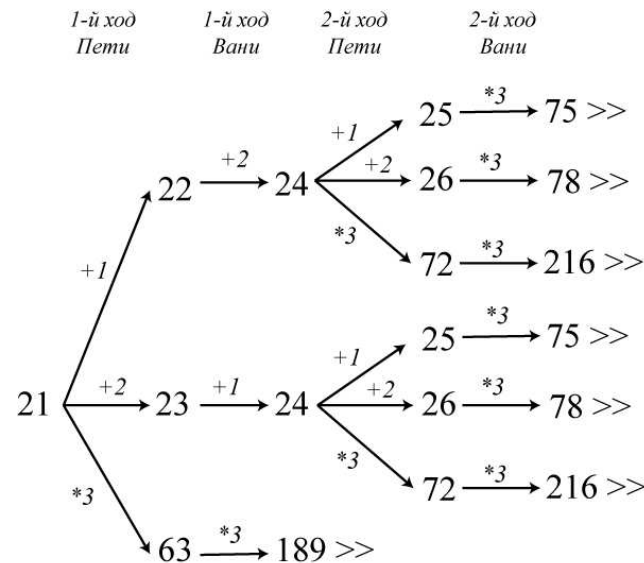


Рис.1. Дерево всех партий, возможных при Ваниной стратегии. Знаком >> обозначены позиции, в которых партия заканчивается.

Указания по оцениванию	Баллы
<p>В задаче от ученика требуется выполнить 3 задания. Их трудность возрастает. Количество баллов в целом соответствует количеству выполненных заданий (подробнее см. ниже).</p> <p>Ошибка в решении, не искажающая основного замысла и не приведшая к неверному ответу, например, арифметическая ошибка при вычислении количества камней в заключительной позиции, при оценке решения не учитывается.</p> <p>Первое задание считается выполненным полностью, если выполнены полностью оба пункта а) и б). Пункт а) считается выполненным полностью, если правильно указаны все позиции, в которых Петя выигрывает первым ходом, и указано, каким должен быть первый ход. Пункт б) считается выполненным полностью, если правильно указана позиция, в которой Ваня выигрывает первым ходом, и описана стратегия Вани, т. е. показано, как Ваня может получить кучу, в которой содержится нужное количество камней, при любом ходе Пети.</p> <p>Первое задание считается выполненным частично, если (1) правильно указаны все позиции, в которых Петя выигрывает первым ходом, (2) правильно указана позиция, в которой Ваня выигрывает первым ходом, и (3) явно сказано, что при любом ходе Пети Ваня может получить кучу, которая содержит нужное для выигрыша количество камней. При этом может быть не указано, каким именно ходом выигрывает Ваня.</p> <p>Второе задание выполнено, если правильно указаны три позиции, выигрышные для Пети, и описана соответствующая стратегия Пети – так, как это написано в примере решения, или другим способом, например, с помощью дерева всех партий, возможных при выбранной стратегии Пети. Допускается также ситуация, когда указаны все три выигрышные позиции, и явно сказано, что из любой из них Петя может получить позицию, в которой второй игрок выигрывает своим первым ходом (позиция разобрана в п.1 б).</p> <p>Третье задание выполнено, если правильно указана позиция, выигрышная для Вани и построено дерево всех партий, возможных при Ваниной стратегии. При изображении дерева для каждой позиции, где должен ходить Петя, должны быть показаны все возможные ходы, а для позиций, где должен ходить Ваня, – только ход, соответствующий выбранной Ваней стратегии.</p>	

ВНИМАНИЕ! При игре по Ваниной стратегии в некоторые состояния кучи можно прийти несколькими путями. Это позволяет «склеить» такие вершины в дереве возможных партий и представлять множество всех возможных партий ориентированным ациклическим графом, который не является деревом. Вместо дерева на рис. 1 можно использовать, например, граф, изображённый на рис. 2. **Это не является ошибкой.**

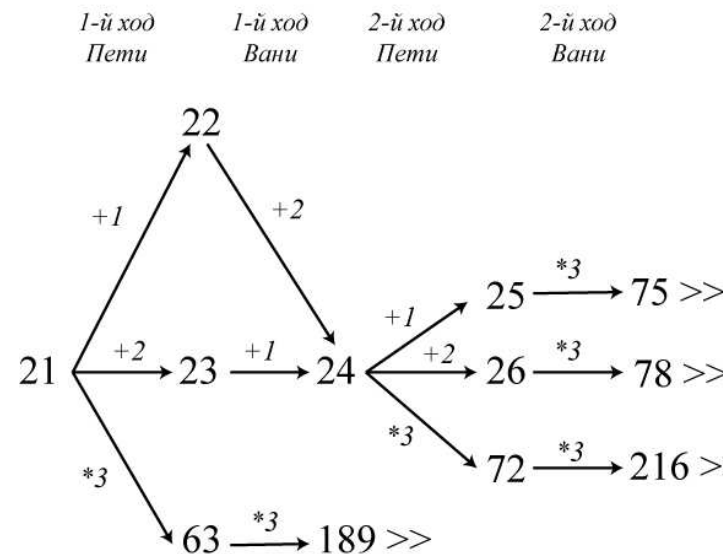


Рис.2. Граф, представляющий множество всех партий, возможных при Ваниной стратегии. Знаком >> обозначены позиции, в которых партия заканчивается.

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

<p>Выполнены второе и третье задания. Первое задание выполнено полностью или частично.</p> <p>Здесь и далее в решениях допускаются арифметические ошибки, которые не искажают сути решения и не приводят к неправильному ответу.</p>	3
<p>Не выполнены условия, позволяющие поставить 3 балла, и выполнено одно из следующих условий.</p> <p>Задание 3 выполнено полностью.</p> <p>Первое и второе задания выполнены полностью.</p> <p>Первое задание выполнено полностью или частично; для заданий 2 и 3 указаны правильные значения S.</p>	2

Не выполнены условия, позволяющие поставить 3 или 2 балла, и выполнено одно из следующих условий. Первое задание выполнено полностью. Во втором задании правильно указано одно из возможных значений S , и для этого значения указана и обоснована выигрышная стратегия Пети. Первое задание выполнено полностью или частично, и для одного из остальных заданий правильно указано значение S . 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 Дед Мороз и Снегурочка приходят на детские утренники с мешком конфет. Дед Мороз делит конфеты поровну между всеми присутствующими детьми (детей на утреннике никогда не бывает больше 100), а оставшиеся конфеты отдает Снегурочке. Снегурочка каждый раз записывает в блокнот количество полученных конфет. Если конфеты разделились между всеми детьми без остатка, Снегурочка ничего не получает и ничего не записывает. Когда утренники закончились, Деду Морозу стало интересно, сколько различных чисел встречается в записях Снегурочки. Дед Мороз и Снегурочка – волшебные, поэтому число утренников N , на которых они побывали, может быть очень большим.

Напишите программу, которая будет решать эту задачу. Перед текстом программы кратко опишите алгоритм решения задачи и укажите используемый язык программирования и его версию.

Желательно, чтобы программа была эффективной как по времени работы, так и по используемой памяти. Программу будем считать эффективной по памяти, если используемая память не зависит от размера входных данных (то есть числа утренников). Программу будем считать эффективной по времени, если при увеличении размера входных данных N в t раз (t – любое число) время её работы увеличивается не более чем в t раз.

Описание входных данных

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

Каждая из следующих N строк содержит два целых числа: сначала D – количество пришедших на очередной утренник детей, а затем K – количество конфет в мешке Деда Мороза на этом утреннике. Гарантируется выполнение следующих соотношений:

$$1 \leq N \leq 10000$$

$$1 \leq D \leq 100 \text{ (для каждого } D\text{)}$$

$$D \leq K \leq 1000 \text{ (для каждой пары } D, K\text{)}$$

Описание выходных данных

Программа должна вывести одно число – количество различных чисел в записях Снегурочки. Если Снегурочка ни разу ничего не записывала, надо вывести ноль.

Пример входных данных:

```
7
10 58
15 315
20 408
100 1000
32 63
32 63
11 121
```

Пример выходных данных для приведённого выше примера входных данных:

```
2
```

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)	
<p>Поскольку количество детей не превышает 100, остаться может не больше 99 конфет. Можно создать массив из 99 элементов и использовать их в качестве счётчиков, хранящих информацию о количестве записей каждого числа.</p> <p>Программа читает исходные данные, не запоминая их в массиве. Для каждой пары (D, K) количество оставшихся конфет определяется как остаток от деления K на D. Если этот остаток положителен, надо увеличить соответствующий счётчик. По окончании ввода и обработки данных надо подсчитать количество ненулевых элементов в массиве счётчиков.</p>	
<p>Пример правильной и эффективной программы на языке Паскаль</p> <pre>program c4; const DMAX=100; {максимально возможное количество детей} var N: integer; {количество утренников} D: integer; {количество детей на утреннике} K: integer; {количество конфет} r: integer; {остаток} c: array[1..DMAX-1] of integer; {счетчики остатков} i: integer;</pre>	

```

m: integer;      {счетчик разных чисел в записях}
begin
  {предварительная очистка счетчиков}
  for i:=1 to DMAX-1 do c[i]:=0;
  readln(N);
  {ввод данных, подсчет количества каждого остатка}
  for i:=1 to N do begin
    readln(D, K);
    r := K mod D;
    if r>0 then c[r]:=c[r]+1;
  end;
  {подсчет количества разных записей}
  m:=0;
  for i:=1 to DMAX-1 do begin
    if c[i]>0 then m:=m+1;
  end;
  writeln(m);
end.

```

Возможны другие варианты решения. Например, можно вместо массива целочисленных счетчиков использовать массив данных логического типа, отмечающих только факт использования того или иного значения. Можно подсчитывать итоговое значение непосредственно в процессе ввода данных, увеличивая величину m на 1 каждый раз, когда встречается новое (не встречавшееся ранее) значение остатка.

Указания по оцениванию	Баллы
Программа правильно работает для любых входных данных и является эффективной как по времени, так и по памяти. Допускается наличие в тексте программы одной синтаксической ошибки: пропущен или неверно указан знак пунктуации, неверно написано или пропущено зарезервированное слово языка программирования, не описана или неверно описана переменная, применяется операция, недопустимая для соответствующего типа данных (если одна и та же ошибка встречается несколько раз, то это считается за одну ошибку).	4

<p>Не выполнены условия, позволяющие поставить 4 балла. Программа работает верно и является эффективной по времени, но размер используемой памяти зависит от количества исходных данных. Например, входные данные (все значения D и K) запоминаются в массиве или другой структуре данных, размер которой зависит от N.</p> <p>Допускается одна из следующих ошибок:</p> <ol style="list-style-type: none"> 1) Вместо остатка от деления K на D вычисляется остаток от деления D на K. 2) Отсутствует предварительная инициализация счётчиков (для языков, в которых нет гарантированного начального обнуления). 3) Неверно обрабатывается ситуация, когда остаток оказался равен 0. <p>Допускается наличие от одной до трёх синтаксических ошибок, описанных в критериях на 4 балла.</p>	3
<p>Не выполнены условия, позволяющие поставить 3 или 4 балла. Программа работает в целом верно, эффективно или нет. Возможны переборные решения, при которых все исходные данные хранятся в массиве (или в двух массивах), этот массив многократно просматривается, при каждом просмотре выбираются пары с определённым остатком. В реализации алгоритма допущено более 1 ошибки из числа перечисленных в критериях на 3 балла или допущены другие ошибки, приводящие к неверной работе программы в отдельных случаях.</p> <p>Допускается наличие от одной до пяти синтаксических ошибок, описанных в критериях на 4 балла.</p>	2
<p>Не выполнены условия, позволяющие поставить 2, 3 или 4 балла. 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.</p>	1
<p>Не выполнены условия, позволяющие поставить 1, 2, 3 или 4 балла.</p>	0
<i>Максимальный балл</i>	4