Критерии оценивания заданий с развёрнутым ответом

С1 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

```
DIM N AS LONG
INPUT N
WHILE N>10
    N = N MOD 10
WEND
PRINT N
END
```

Паскаль

```
var n: longint;
begin
    read(n);
    while n>10 do begin
        n := n mod 10
    end;
    write(n);
end.
```

Си

```
#include <stdio.h>
void main(){
    long int n;
    scanf("%ld",&n);
    while (n>10) {
        n = n%10;
    }
    printf("%ld", n);
}
```

Алгоритмический язык

```
алг
нач
    цел n
    ввод n
    нц пока n>10
        n := mod(n,10)
    кц
    вывод n
кон
```

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 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
В задаче требуется выполнить **три** действия. 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. 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. 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).	
Правильно выполнены все действия: 1) указан верный результат для приведённого примера входных данных; 2) дан пример числа, для которого программа с ошибками выдаёт тот же результат, что и правильная программа; 3) указаны и исправлены две ошибки в программе; 4) не указаны в качестве ошибочных никакие другие строки программы.	3

Не выполнены условия, позволяющие поставить 3 балла, и имеет место один из следующих случаев: 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
Не выполнены условия, позволяющие поставить 2 или 3 балла, и имеет место один из следующих случаев: 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
Не выполнены условия, позволяющие поставить 1, 2 или 3 балла.	0
Максимальный балл	*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

```
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.
```

Бейсик

```
N=2014
DIM A(N) AS INTEGER
DIM I, J, K AS INTEGER
FOR I = 1 TO N
    INPUT A(I)
NEXT I
…
END
```

Си

```
#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]);
    …
}
```

Алгоритмический язык

```
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
```

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
Максимальный балл	*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, $1 \le S \le 63$.

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)

1. а) Петя может выиграть, если S = 22, …, 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.

2. 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 выиграет.

	Положения после очередных ходов			
Исходное положение	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**		

3. 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 выиграет 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).

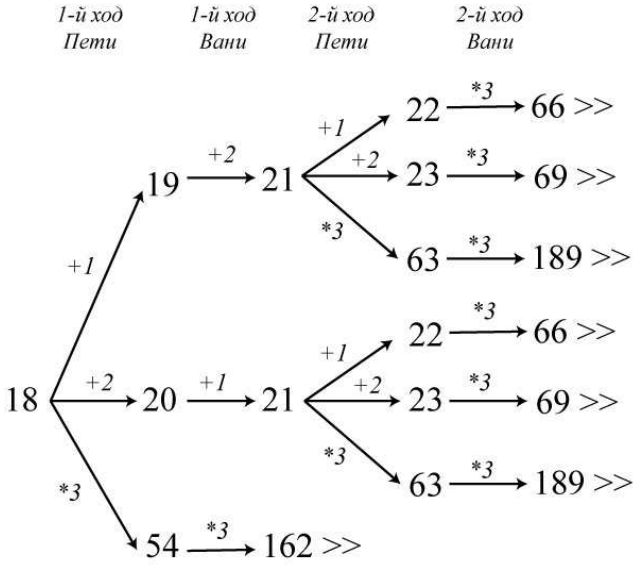
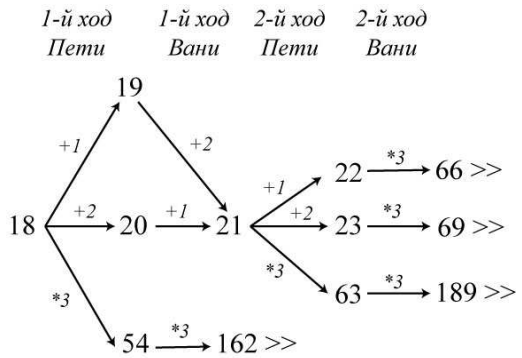


Рис.1. Дерево всех партий, возможных при Ваниной стратегии. Знаком >> обозначены позиции, в которых партия заканчивается.

Указания по оцениванию	Баллы
В задаче от ученика требуется выполнить 3 задания. Их трудность возрастает. Количество баллов в целом соответствует количеству выполненных заданий (подробнее см. ниже). Ошибка в решении, не искажающая основного замысла и не приведшая к неверному ответу, например, арифметическая ошибка при вычислении количества камней в заключительной позиции, при оценке решения не учитывается. 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. 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. 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). 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.	

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

$1 \le N \le 10000$

$1 \le D \le 100$ (для каждого D)

$D \le K \le 1000$ (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
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
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. Допускается одна из следующих ошибок: 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. Допускается наличие от одной до трёх синтаксических ошибок, описанных в критериях на 4 балла.	3
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. Допускается наличие от одной до пяти синтаксических ошибок, описанных в критериях на 4 балла.	2

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.	1
Не выполнены условия, позволяющие поставить 1, 2, 3 или 4 балла.	0
Максимальный балл	*4*

Критерии оценивания заданий с развёрнутым ответом

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

Бейсик

```
DIM N AS LONG
INPUT N
WHILE N>100
    N = N MOD 100
WEND
PRINT N
END
```

Паскаль

```
var n: longint;
begin
    read(n);
    while n>100 do begin
        n := n mod 100
    end;
    write(n);
end.
```

Си

```
#include <stdio.h>
void main(){
    long int n;
    scanf("%ld",&n);
    while (n>100) {
        n = n % 100;
    }
    printf("%ld", n);
}
```

Алгоритмический язык

```
алг
нач
    цел n
    ввод n
    нц пока n>100
        n := mod(n,100)
    кц
    вывод n
кон
```

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
В задаче требуется выполнить **три** действия. 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. 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. 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).	
Правильно выполнены все действия: 1) указан верный результат для приведённого примера входных данных; 2) дан пример числа, для которого программа с ошибками выдаёт тот же результат, что и правильная программа; 3) указаны и исправлены две ошибочные строки в программе; 4) не указаны в качестве ошибочных никакие другие строки программы.	3

Не выполнены условия, позволяющие поставить 3 балла, и имеет место один из следующих случаев: 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
Не выполнены условия, позволяющие поставить 2 или 3 балла, и имеет место один из следующих случаев: 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
Не выполнены условия, позволяющие поставить 1, 2 или 3 балла.	0
Максимальный балл	*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 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

```
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.
```

Бейсик

```
N=2014
DIM A(N) AS INTEGER
DIM I, J, K AS INTEGER
FOR I = 1 TO N
    INPUT A(I)
NEXT I
…
END
```

Си

```
#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]);
    …
}
```

Алгоритмический язык

```
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
```

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
Максимальный балл	*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, $1 \le S \le 74$.

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, …, 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 выиграет.

	Положения после очередных ходов			
Исходное положение	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 выиграет 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).

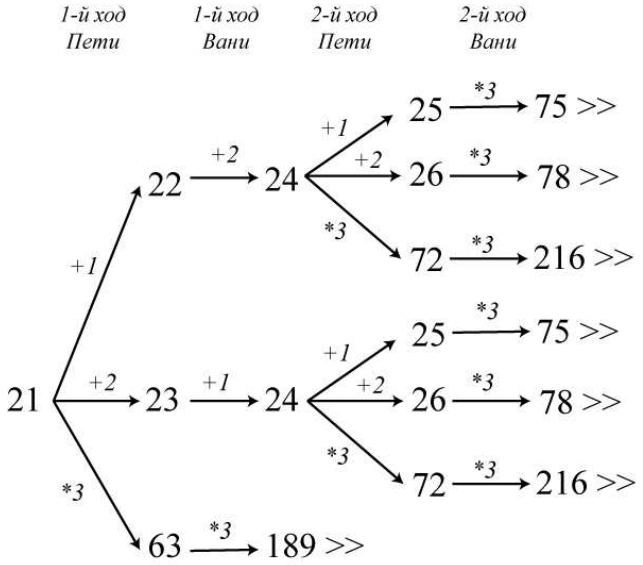
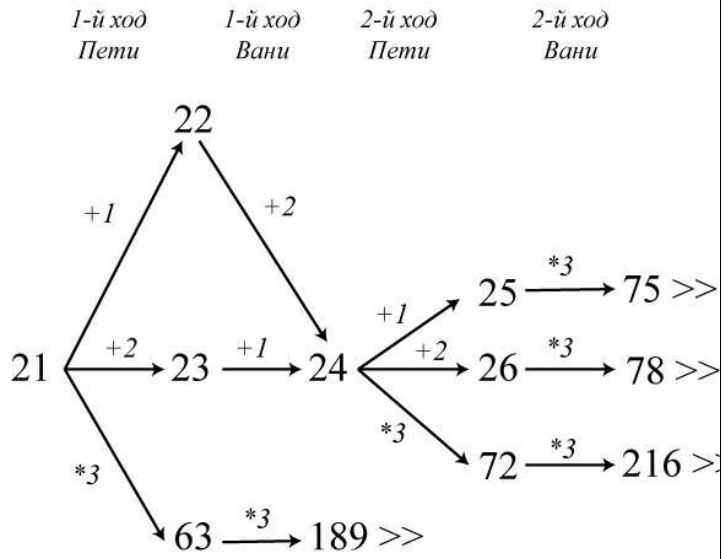


Рис.1. Дерево всех партий, возможных при Ваниной стратегии. Знаком >> обозначены позиции, в которых партия заканчивается.

Указания по оцениванию	Баллы
В задаче от ученика требуется выполнить 3 задания. Их трудность возрастает. Количество баллов в целом соответствует количеству выполненных заданий (подробнее см. ниже). Ошибка в решении, не искажающая основного замысла и не приведшая к неверному ответу, например, арифметическая ошибка при вычислении количества камней в заключительной позиции, при оценке решения не учитывается. 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. 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. 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). 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.	

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

Не выполнены условия, позволяющие поставить 3 или 2 балла, и выполнено одно из следующих условий. 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
Максимальный балл	*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:

$1 \le N \le 10000$

$1 \le D \le 100$ (для каждого D)

$D \le K \le 1000$ (для каждой пары D, K)

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)

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 подсчитать количество ненулевых элементов в массиве 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;
```

```
   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

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. Допускается одна из следующих ошибок: 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. Допускается наличие от одной до трёх синтаксических ошибок, описанных в критериях на 4 балла.	3
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. Допускается наличие от одной до пяти синтаксических ошибок, описанных в критериях на 4 балла.	2
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.	1
Не выполнены условия, позволяющие поставить 1, 2, 3 или 4 балла.	0
Максимальный балл	*4*

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	Б13
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	Б4
B11	DECH
B12	178
B13	55
B14	40
B15	31