

### Критерии оценивания заданий с развёрнутым ответом

24

Факториалом натурального числа  $n$  (обозначается  $n!$ ) называется произведение всех натуральных чисел от 1 до  $n$ . Например,  $4! = 1 \cdot 2 \cdot 3 \cdot 4 = 24$ .

Даны целые положительные числа  $A$  и  $B$ ,  $0 < A < B < 10^9$  (выполнение этого неравенства гарантируется и не должно проверяться). Необходимо найти и вывести количество точных факториалов, расположенных строго между этими числами, то есть количество таких  $N$ , что  $A < N! < B$ .

Например, при  $A = 1$  и  $B = 10$  нужно вывести число 2 (возможные значения  $K$  – 2 и 3).

Для решения этой задачи ученик написал программу, но, к сожалению, его программа неправильная.

Ниже эта программа для Вашего удобства приведена на пяти языках программирования.

Бейсик	Python
<pre> DIM A, B, C, N, F AS INTEGER INPUT A, B C = 0 N = 1 F = 1 WHILE F &lt;= B     N = N + 1     F = F * N     IF F &gt; A THEN         C = C + 1     END IF WEND PRINT C         </pre>	<pre> a = int(input()) b = int(input()) c = 0 n = 1 f = 1 while f &lt;= b:     n += 1     f *= n     if f &gt; a:         c += 1 print(c)         </pre>
Алгоритмический язык	Паскаль
<pre> алг нач     цел a, b, c, n, f     ввод a, b     c := 0     n := 1     f := 1     нц пока f &lt;= b         n := n + 1         f := f * n         если f &gt; a то             c := c + 1         все     кц     вывод c кон         </pre>	<pre> var a, b, c, n, f: integer; begin     readln(a,b);     c := 0;     n := 1;     f := 1;     while f &lt;= b do begin         n := n + 1;         f := f * n;         if f &gt; a then             c := c + 1         end;     write(c) end.         </pre>

**C++**

```
#include <iostream>
using namespace std;
int main() {
    int a, b, c, n, f;
    cin >> a >> b;
    c = 0;
    n = 1;
    f = 1;
    while (f < b) {
        ++n;
        f *= n;
        if (f > a)
            ++c;
    }
    cout << c;
    return 0;
}
```

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе чисел 3 и 15.
2. В программу ввели значение  $A = 1$ . Назовите минимальное значение  $B$ , при вводе которого программа выведет ответ, который на 2 больше верного.
3. Найдите в программе все ошибки (известно, что их не больше двух) и исправьте их. Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.

Достаточно указать ошибки и способ их исправления для одного языка программирования.

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 чисел 3 и 15 программа выведет число 2.
2. Минимальное значение  $B = 2$ . При  $A = 1$  и  $B = 2$  программа выводит ответ 2 вместо верного ответа 0.
3. Программа содержит две ошибки.

**Первая ошибка.** Неверное сравнение с  $B$  в условии цикла. В задаче требуется подсчитывать факториалы, расположенные строго между заданными числами, поэтому сравнение должно быть строгим.

**Вторая ошибка.** Неверная проверка диапазона. Вычисление факториала в цикле производится после сравнения с  $B$ , полученное значение может выходить за границы заданного диапазона

**Пример исправления для алгоритмического языка****Первая ошибка:**

```
нц пока f <= b
```

Исправленная строка:

```
нц пока f < b
```

**Вторая ошибка.** Следовало бы поменять местами вычисление факториала и его сравнение с  $A$ , но в условии разрешено только исправлять строки, поэтому приходится вставлять дополнительное сравнение.

Исходная строка:

```
если f > a то
```

Исправленная строка:

```
если f > a и f < b то
```

Если язык программирования допускает двойные неравенства (алгоритмический язык, Python), возможна более простая запись исправления:

```
если a < f < b то
```

Для Бейсика, Паскаля и C++ запись в виде двойного неравенства считается ошибочной.

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

Незначительной опiskой, не влияющей на оценку, следует считать отсутствие служебных слов и знаков, не влияющих на содержательную часть исправления (например, слова «то» или точки с запятой)

Указания по оцениванию	Баллы
<p>В задаче требуется выполнить <b>четыре</b> действия.</p> <ol style="list-style-type: none"> <li>1. Указать, что выведет программа при конкретном вводе.</li> <li>2. Определить минимальное значение <math>B</math>, при вводе которого программа выдаёт подходящий ответ для заданного <math>A</math>.</li> <li>3. Исправить первую ошибку в программе.</li> <li>4. Исправить вторую ошибку в программе.</li> </ol> <p>Действие 1 считается выполненным, если указан верный результат работы программы при заданных входных значениях.</p> <p>Действие 2 считается выполненным, если указано верное значение <math>B</math>.</p> <p>Для действий 1 и 2 учащийся не обязан объяснять, как получен результат, достаточно указать верные числа.</p> <p>Каждое из действий 3 и 4 считается выполненным при одновременном выполнении двух условий:</p> <ol style="list-style-type: none"> <li>а) правильно указана строка с ошибкой;</li> <li>б) указан такой новый вариант строки, что при исправлении другой ошибки получается правильная программа.</li> </ol> <p>В исправленной строке допускаются незначительные синтаксические ошибки (лишние или пропущенные знаки препинания, неточные написания служебных слов языка)</p>	

Выполнены все четыре необходимых действия, ни одна верная строка не указана в качестве ошибочной	3
Не выполнены условия, позволяющие поставить 3 балла. Имеет место одна из следующих ситуаций. 1. Выполнены три из четырёх необходимых действий, ни одна верная строка не названа ошибочной. 2. Выполнены все четыре необходимых действия, одна верная строка названа ошибочной	2
Не выполнены условия, позволяющие поставить 2 или 3 балла. Выполнены два из четырёх необходимых действий	1
Не выполнены условия, позволяющие поставить 1, 2 или 3 балла.	0
<i>Максимальный балл</i>	<i>3</i>

25

Дан массив, содержащий 2019 положительных целых чисел, не превышающих 15 000. Необходимо найти количество элементов массива, шестнадцатеричная запись которых заканчивается цифрой A, заменить на это количество все элементы, восьмеричная запись которых содержит ровно 2 цифры, и вывести изменённый массив.

Напишите на одном из языков программирования программу для решения этой задачи. Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из описанных.

<b>Бейсик</b>	<b>Python</b>
<pre> CONST N = 2019 DIM A(N) AS INTEGER DIM I, M, K, D AS INTEGER FOR I = 1 TO N     INPUT A(I) NEXT I ... END </pre>	<pre> # кроме уже указанных # допускается использование # целочисленных переменных # m, k, d a = [] N = 2019 for i in range(0, N):     a.append(int(input())) ... </pre>
<b>Алгоритмический язык</b>	<b>Паскаль</b>
<pre> алг нач     цел N=2019     целтаб a[1:N]     цел i, m, k, d     нц для i от 1 до N         ввод a[i]     кц     ... Кон </pre>	<pre> const     N=2019; var     a: array [1..N] of integer;     i, m, k, d: integer; begin     for i := 1 to N do         readln(a[i]);     ... end. </pre>

**C++**

```
#include <iostream>
using namespace std;
const int N=2019;
int main(){
    int a[N];
    int i, m, k, d;
    for (i=0; i<N; ++i)
        cin >> a[i];

    ...
    return 0;
}
```

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 2.6). 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)

Задача решается в два прохода: на первом проходе определяется количество требуемых элементов, на втором проходе производится корректировка и вывод элементов.

Возможно решение в три прохода, когда на втором проходе выполняется только замена значений, а на третьем – вывод

**Пример правильной программы на алгоритмическом языке**

```
к := 0
нц для i от 1 до N
    если mod(a[i],16)=10
        то к := к + 1
    все
кц
нц для i от 1 до N
    если 8 <= a[i] < 64
        то a[i] := к
    все
    вывод a[i], ' '
кц
```

При использовании языка Python первый проход можно записать в одну строку, используя специальные средства этого языка.

**Пример правильной программы на языке Python**

```
k = sum(1 for i in a if i%16 == 10)
for i in range(0,N):
    if 8 <= a[i] < 64:
        a[i] = k
    print(a[i])
```

Использовать описанную возможность не обязательно, на языке Python допустимо описывать развёрнутый алгоритм решения, аналогичный приведённой выше программе на алгоритмическом языке.

Возможны решения, в которых производится полный подсчёт количества цифр в восьмеричной записи. Такое решение более громоздко и менее эффективно, но при отсутствии ошибок оно оценивается полным баллом. Ниже приводится пример такого решения на языке Паскаль

#### Пример правильной программы на языке Паскаль

```
k := 0;
for i:=1 to N do begin
  if a[i] mod 16 = 10 then k :=k + 1
end;
for i:=1 to N do begin
  m := a[i]; d := 0;
  while m > 0 do begin
    d := d + 1;
    m := m div 8
  end;
  if d = 2 then a[i] := k;
  write(a[i], ' ')
end
```

Указания по оцениванию	Баллы
<p>В программе допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора.</p> <p>Эффективность не имеет значения и не оценивается.</p> <p>Допускается запись программы на языке, не входящем в список языков из условия. В этом случае должны использоваться переменные, аналогичные описанным в условии. Если выбранный 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.</p> <p>Допускается произвольный формат вывода полученного массива, например, вывод всех элементов массива в одну строку или вывод каждого элемента в отдельной строке. Отмечается как ошибка, но не учитывается при выставлении оценки вывод элементов в одну строку без пробелов между ними</p>	
Предложена правильная программа, которая изменяет исходный массив в соответствии с условием и выводит изменённый массив	2
Не выполнены условия, позволяющие поставить 2 балла	1

<p>Предложено в целом верное решение, содержащее не более одной ошибки из числа следующих (если одинаковая ошибка повторяется несколько раз, она считается за одну ошибку).</p> <ol style="list-style-type: none"> <li>1) Отсутствие инициализации или неверная инициализация счётчика.</li> <li>2) Неверное определение последней цифры в шестнадцатеричной записи.</li> <li>3) Неверное определение количества цифр в восьмеричной записи.</li> <li>4) Неверное построение логических условий (неверные логические операции, проверка не всех условий).</li> <li>5) Выход за границы массива.</li> <li>6) Исходный массив не изменяется.</li> <li>7) Изменяются не все элементы, которые должны измениться, или изменяются элементы, которые не должны измениться.</li> <li>8) Полученный массив не выводится или выводится не полностью (например, выводится только первый элемент или только изменённые элементы).</li> <li>9) Используется переменная, не объявленная в разделе описания переменных.</li> <li>10) Индексная переменная в цикле не меняется (например, в цикле <code>while</code>) или меняется неверно</li> </ol>	
Не выполнены условия, позволяющие поставить 1 или 2 балла	0
<i>Максимальный балл</i>	2

26

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может **добавить** в одну из куч **один камень** или **увеличить** количество камней **в куче в четыре раза**. Например, пусть в одной куче 6 камней, а в другой 9 камней; такую позицию мы будем обозначать (6, 9). За один ход из позиции (6, 9) можно получить любую из четырёх позиций: (7, 9), (24, 9), (6, 10), (6, 36). Чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 82. Победителем считается игрок, сделавший последний ход, то есть первым получивший позицию, в которой в кучах будет 82 или больше камней.

В начальный момент в первой куче было 4 камня, во второй куче –  $S$  камней,  $1 \leq S \leq 77$ .

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. В описание выигрышной стратегии не следует включать ходы играющего по ней игрока, которые

не являются для него безусловно выигрышными, то есть не гарантируют выигрыш независимо от игры противника.

Выполните следующие задания.

### Задание 1.

- а) Назовите все значения  $S$ , при которых Петя может выиграть первым ходом.
- б) Петя сделал неудачный первый ход, после которого Ваня выиграл своим первым ходом. Назовите минимальное значение  $S$ , при котором это возможно.

### Задание 2.

Укажите такое значение  $S$ , при котором у Пети есть выигрышная стратегия, причём Петя не может выиграть первым ходом, но может выиграть своим вторым ходом независимо от того, как будет ходить Ваня. Для указанного значения  $S$  опишите выигрышную стратегию Пети.

### Задание 3.

Укажите такое значение  $S$ , при котором у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, и при этом 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). В узлах дерева указывайте игровые позиции. Дерево не должно содержать партий, невозможных при реализации выигрывающим игроком своей выигрышной стратегии. Например, полное дерево игры не будет верным ответом на это задание.

<p style="text-align: center;"><b>Содержание верного ответа</b> (допускаются иные формулировки ответа, не искажающие его смысла)</p>
<p><b>Задание 1.</b></p> <p>а) Петя может выиграть первым ходом, если <math>S = 20, \dots, 77</math>. Для выигрыша достаточно умножить количество камней во второй куче. При меньших значениях <math>S</math> за один ход нельзя получить 82 или более камней в двух кучах.</p> <p>б) Такая ситуация возможна при <math>S = 5</math>. Если Петя умножит вторую кучу, получится позиция (4, 20), из которой Ваня может получить позицию (4, 80) и выиграть. При <math>S &lt; 5</math> никакой первый ход Пети не создаст ситуацию, в которой Ваня может сразу выиграть.</p> <p><b>Задание 2.</b></p> <p>Возможные значения <math>S</math>: 16, 19. В этих случаях Петя, очевидно, не может выиграть первым ходом. Однако при <math>S = 16</math> Петя может получить позицию (16, 16), а при <math>S = 19</math> – позицию (5, 19).</p> <p>В первом случае после хода Вани возникнет одна из позиций (17, 16), (64, 16), (16, 17), (16, 64), во втором случае – одна из позиций (6, 19), (20, 19), (5, 20), (5, 76). В любой из перечисленных позиций Петя может выиграть, умножив количество камней во второй куче</p>



**Задание 3.**

Возможное значение  $S$ : 18. После первого хода Пети возможны позиции (5, 18), (16, 18), (4, 19), (4, 72). В позициях (16, 18) и (4, 72) Ваня может выиграть первым ходом, умножив количество камней в любой куче. Из позиций (5, 18) и (4, 19) Ваня может получить позицию (5, 19), разобранный в задании 2. Игрок, после хода которого возникла эта позиция (в данном случае – Ваня), выигрывает следующим ходом.

В таблице изображены возможные партии при описанной стратегии Вани. Заключительные позиции (в них выигрывает Ваня) выделены жирным шрифтом. На рисунке эти же партии показаны в виде графа (оба способа изображения допустимы).

	Положения после очередных ходов			
Исходное положение	1-й ход Пети (разобраны все ходы, указана полученная позиция)	1-й ход Вани (только ход по стратегии, указана полученная позиция)	2-й ход Пети (разобраны все ходы, указана полученная позиция)	2-й ход Вани (только ход по стратегии, указана полученная позиция)
(4, 18) Всего 22	(16, 18) Всего 34	(16, 72) <b>Всего 88</b>		
	(4, 72) Всего 76			
	(5, 18) Всего 23	(5, 19) Всего 24	(6, 19) Всего 25	(6, 76) <b>Всего 82</b>
			(20, 19) Всего 39	(20, 76) <b>Всего 96</b>
	(4, 19) Всего 23		(5, 76) Всего 81	
			(5, 20) Всего 74	(5, 80) <b>Всего 85</b>

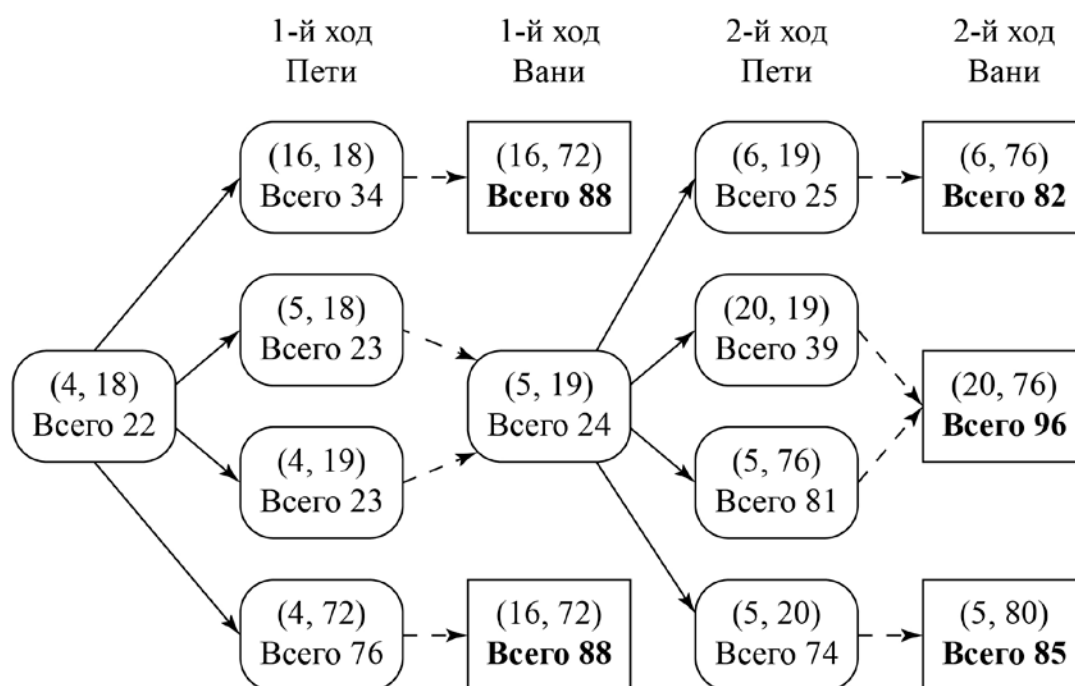


Рис. 1. Граф всех партий, возможных при описанной стратегии Вани. Ходы Пети показаны сплошными стрелками, ходы Вани – пунктирными стрелками. Заключительные позиции обозначены прямоугольниками.

*Примечание для эксперта.* Дерево всех партий может быть изображено в виде таблицы или ориентированного графа – так, как показано на рисунке, или другим способом. Например, вместо приведённого здесь «экономного» варианта, в котором позиции не дублируются, возможно построение полного дерева, в котором одинаковые позиции, возникающие при различном ходе игры, показаны отдельно. Важно, чтобы множество полных путей в графе находилось во взаимно однозначном соответствии с множеством партий, возможных при описанной в решении стратегии. В некоторых позициях заключительный выигрывающий ход можно сделать несколькими способами. В таблице и на рисунке указан один из них, в работе допускается выбор любого допустимого заключительного выигрывающего хода.

Указания по оцениванию	Баллы
<p>В задаче от ученика требуется выполнить <b>три</b> задания. 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>Задание 1 выполнено, если выполнены оба пункта: для пункта (а) перечислены все удовлетворяющие условию значения <math>S</math>, и только они, для пункта (б) указано верное значение <math>S</math>, и только оно. Обоснование найденных значений не обязательно</p>	

Задание 2 выполнено, если верно указана выигрышная для Пети позиция (любая из двух возможных) и описана соответствующая стратегия.  Задание 3 выполнено, если правильно указана выигрышная для Вани позиция и построено дерево всех возможных при выигрышной стратегии партий (и только их).  Во всех случаях стратегии могут быть описаны так, как это сделано в примере решения, или другим способом	
Выполнены все три задания	3
Не выполнены условия, позволяющие поставить 3 балла, и выполнено хотя бы одно из следующих условий. – Выполнено задание 3. – Выполнены задания 1 и 2	2
Не выполнены условия, позволяющие поставить 2 или 3 балла, и выполнено хотя бы одно из заданий 1 и 2	1
Не выполнено ни одно из условий, позволяющих поставить 1, 2 или 3 балла	0
<i>Максимальный балл</i>	<i>3</i>

27

Дана последовательность  $N$  целых неповторяющихся положительных чисел. Рассматриваются все пары элементов последовательности, разность которых делится на  $m = 80$ . Среди всех таких пар нужно найти и вывести пару с максимальной разностью элементов. Если одинаковую максимальную разность имеют несколько пар, можно вывести любую из них. Если подходящих пар в последовательности нет, нужно вывести два нуля.

#### Описание входных и выходных данных

В первой строке входных данных задаётся количество чисел  $N$  ( $2 \leq N \leq 10\,000$ ). В каждой из последующих  $N$  строк записано одно натуральное число, не превышающее 10 000. Гарантируется, что никакое число не встречается в последовательности более одного раза.

*Пример входных данных:*

8  
95  
163  
5  
40  
15  
3  
85  
80

*Пример выходных данных для приведённого выше примера входных данных:*

3 163

*Пояснение.* Из данных восьми чисел можно составить три пары, удовлетворяющие условию: (15, 95), (3, 163), (5, 85). Наибольшая разность получается в паре (3, 163).

Напишите эффективную по времени и по памяти программу для решения этой задачи.

Программа считается эффективной по времени, если при одновременном увеличении количества исходных чисел  $N$  и параметра  $m$  в  $k$  раз время работы программы увеличивается не более чем в  $k$  раз.

Программа считается эффективной по памяти, если память, необходимая для хранения всех переменных программы, не превышает 4 Кбайт и не увеличивается с ростом  $N$ .

Максимальная оценка за правильную (не содержащую синтаксических ошибок и дающую правильный ответ при любых допустимых входных данных) программу, эффективную по времени и по памяти, – 4 балла.

Максимальная оценка за правильную программу, эффективную только по времени или только по памяти, – 3 балла.

Максимальная оценка за правильную программу, не удовлетворяющую требованиям эффективности, – 2 балла.

Вы можете сдать **одну** или **две** программы решения задачи. Если Вы сдадите две программы, каждая из них будет оцениваться независимо от другой, итоговой станет **бóльшая** из двух оценок.

Перед текстом программы кратко опишите алгоритм решения. Укажите использованный язык программирования и его версию.

Содержание верного ответа (допускаются иные формулировки ответа, не искажающие его смысла)
<p>Разность двух чисел кратна <math>m</math>, остатки от деления этих чисел на <math>m</math> равны. При этом для получения максимальной разности нужно, чтобы одно из чисел было как можно больше, а второе как можно меньше.</p> <p>Будем хранить в одном массиве из <math>m</math> элементов максимальные числа, имеющие соответствующий остаток от деления на <math>m</math>, а в другом – минимальные, и из всех пар максимумов и минимумов выберем пару с наибольшей разностью.</p> <p>При этом нужно убедиться, что это именно пара. Поскольку в условии гарантируется, что числа в последовательности не повторяются, если минимум оказался равен максимуму, значит, пары на самом деле нет.</p> <p>Ниже приведена программа на алгоритмическом языке, реализующая этот алгоритм</p>
<b>Решение 1. Правильная и эффективная программа на алгоритмическом языке</b>

```

алг задача27
нач
    цел m = 80
    цел таб mn[0:m-1], mx[0:m-1]
    цел N
    цел x | очередное число из последовательности
    цел p | остаток
    цел pm | остаток, дающий лучшую разность
    цел i

    нц для i от 0 до m-1
        mn[i] := 0; mx[i] := 0
    кц
    pm := 0
    ввод N
    нц N раз
        ввод x
        p:=mod(x,m)
        если mn[p] = 0 или x < mn[p]
            то mn[p] := x
        все
        mx[p] := imax(mx[p],x)
        если mx[p]-mn[p] > mx[pm]-mn[pm]
            то pm := p
        все
    кц
    если mx[pm] = mn[pm]
        то вывод '0 0'
        иначе вывод mn[pm], ' ', mx[pm]
    все
кон

```

В приведённом решении лучшая пара находится «на лету»: после обработки каждого числа из последовательности проверяется, не увеличилась ли возможная разность.

Эту проверку можно провести после завершения ввода, вычислив разности для всех остатков и сравнив их. Такая программа тоже эффективна по времени и по памяти, а при больших  $N$  она работает даже быстрее предыдущей, так как выполняет меньше вычислений и сравнений.

Ниже приведена реализующая этот алгоритм программа на языке Python

### Решение 2. Правильная и эффективная программа на языке Python

```

m = 80
mn = [0] * m
mx = [0] * m
N = int(input())
for i in range(N):
    x = int(input())
    p = x % m
    if mn[p] == 0 or x < mn[p]:

```

```

        mn[p] = x
        if x > mx[p]:
            mx[p] = x
p = 0
for i in range(m):
    if mx[i]-mn[i] > mx[p]-mn[p]:
        p = i
if mx[p] == mn[p]:
    print(0,0)
else:
    print(mn[p], mx[p])

```

Возможно также «лобовое» решение: запишем все исходные числа в массив, переберём все возможные пары и выберем подходящую. Такое решение не является эффективным ни по памяти (требуемая память зависит от размера исходных данных), ни по времени (количество возможных пар, а значит, количество действий и время счёта с ростом количества исходных элементов растёт квадратично). Подобная программа оценивается не выше 2 баллов.

Ниже приведена реализующая описанный выше алгоритм программа на языке Паскаль (использована версия PascalABC)

### Решение 3. Правильная, но неэффективная программа на языке Паскаль

```

const m=80;
var
    N: integer;      {количество чисел}
    a: array [1..10000] of integer; {исходные данные}
    x1, x2: integer; {ответ - пара чисел}
    i, j: integer;

begin
    readln(N);
    for i:=1 to N do readln(a[i]);
    x1 := 0; x2 := 0;
    for i := 1 to N do begin
        for j := i+1 to N do begin
            if ((a[i] - a[j]) mod m = 0) and
                (abs(a[i]-a[j]) > abs(x1-x2))
            then begin
                x1 := a[i]; x2 := a[j]
            end
        end
    end;
    writeln(x1, ' ', x2)
end.

```

Указания по оцениванию	Баллы
Если в работе представлены две программы решения задачи, то каждая из них независимо оценивается по указанным ниже критериям, итоговой считается бóльшая из двух оценок. Описание алгоритма решения без программы оценивается 0 баллов	

<p>Программа правильно работает для любых входных данных произвольного размера. Используемая память не зависит от количества прочитанных чисел <math>N</math>, время работы пропорционально <math>N</math> и <math>m</math>.</p> <p>Допускается наличие в тексте программы до трёх синтаксических ошибок одного из следующих видов:</p> <ol style="list-style-type: none"> <li>1) пропущен или неверно указан знак пунктуации;</li> <li>2) неверно написано, пропущено или написано лишнее зарезервированное слово языка программирования;</li> <li>3) не описана или неверно описана переменная;</li> <li>4) применяется операция, недопустимая для соответствующего типа данных.</li> </ol> <p>Если одна и та же ошибка встречается несколько раз, это считается за одну ошибку</p>	4
<p>Не выполнены условия, позволяющие поставить 4 балла.</p> <p>Программа в целом работает правильно для любых входных данных произвольного размера. Время работы пропорционально количеству введённых чисел <math>N</math> и параметру <math>m</math>.</p> <p>Используемая память, возможно, зависит от количества прочитанных чисел (например, входные данные запоминаются в массиве или другой аналогичной структуре данных).</p> <p>Количество синтаксических ошибок («описок»), указанных в критериях на 4 балла, – не более пяти.</p> <p>Допускается наличие не более одной ошибки следующих видов:</p> <ol style="list-style-type: none"> <li>1) ошибка при вводе данных (не считывается значение <math>N</math> или неверно организован ввод последовательности);</li> <li>2) ошибка при инициализации или отсутствие инициализации там, где она необходима;</li> <li>3) используется неверный тип данных;</li> <li>4) использована одна переменная (константа) вместо другой;</li> <li>5) используется один знак операции вместо другого;</li> <li>6) отсутствует вывод ответа или выводится не то значение (например, выводится разность вместо пары чисел);</li> <li>7) неверная работа с массивом, в том числе выход за границы массива;</li> <li>8) пропущены или неверно расставлены операторные скобки (при использовании языков с операторными скобками);</li> <li>9) неверно обрабатывается ситуация, когда в массиве нет подходящих пар</li> </ol>	3
<p>Не выполнены условия, позволяющие поставить 3 или 4 балла, при этом программа работает в целом верно и эффективно по времени. Допускается наличие до трёх содержательных ошибок, описанных в критериях на 3 балла, и до девяти синтаксических ошибок, описанных в критериях на 4 балла</p>	2

<p>ИЛИ</p> <p>Программа работает в целом верно и эффективно по времени, но в ней есть не подходящие под перечень из критериев на 3 балла ошибки, которые в некоторых особых случаях приводят к неверным результатам.</p> <p>ИЛИ</p> <p>Представлено корректное переборное решение, в котором все исходные данные сохраняются в массиве (или другой аналогичной структуре) и рассматриваются все возможные пары. При этом не допускаются содержательные логические ошибки, например выход индексов за границы массива, рассмотрение пар вида <math>a[i]-a[i]</math> и т. д.</p>	
<p>Не выполнены условия, позволяющие поставить 2, 3 или 4 балла.</p> <p>При этом программа представлена и содержит как минимум два обязательных элемента, возможно, реализованных с ошибками:</p> <ol style="list-style-type: none"> <li>1) рассматриваются пары с подходящей разностью;</li> <li>2) обнаруживается и выводится пара с максимальной разностью</li> </ol>	1
<p>Не выполнены условия, позволяющие поставить 1, 2, 3 или 4 балла</p>	0
<p><i>Максимальный балл</i></p>	4



### Критерии оценивания заданий с развёрнутым ответом

24

Факториалом натурального числа  $n$  (обозначается  $n!$ ) называется произведение всех натуральных чисел от 1 до  $n$ . Например,  $4! = 1 \cdot 2 \cdot 3 \cdot 4 = 24$ .

Даны целые положительные числа  $A$  и  $B$ ,  $0 < A < B < 10^9$  (выполнение этого неравенства гарантируется и не должно проверяться). Необходимо найти и вывести количество точных факториалов, расположенных строго между этими числами, то есть количество таких  $N$ , что  $A < N! < B$ .

Например, при  $A = 1$  и  $B = 10$  нужно вывести число 2 (возможные значения  $K$  – 2 и 3).

Для решения этой задачи ученик написал программу, но, к сожалению, его программа неправильная.

Ниже эта программа для Вашего удобства приведена на пяти языках программирования.

Бейсик	Python
<pre> DIM A, B, C, N, F AS INTEGER INPUT A, B C = 0 N = 2 F = 2 WHILE F &lt; B     IF F &gt;= A THEN         C = C + 1     END IF     F = F * N     N = N + 1 WEND PRINT C         </pre>	<pre> a = int(input()) b = int(input()) c = 0 n = 2 f = 2 while f &lt; b:     if f &gt;= a:         c += 1     f *= n     n += 1 print(c)         </pre>
Алгоритмический язык	Паскаль
<pre> алг нач     цел a, b, c, n, f     ввод a, b     c := 0     n := 2     f := 2     нц пока f &lt; b         если f &gt;= a то             c := c + 1         все         f := f * n         n := n + 1     кц     вывод c кон         </pre>	<pre> var a, b, c, n, f: integer; begin     readln(a,b);     c := 0;     n := 2;     f := 2;     while f &lt; b do begin         if f &gt;= a then             c := c + 1;         f := f * n;         n := n + 1     end;     write(c) end.         </pre>

**C++**

```
#include <iostream>
using namespace std;
int main() {
    int a, b, c, n, f;
    cin >> a >> b;
    c = 0;
    n = 2;
    f = 2;
    while (f < b) {
        if (f >= a)
            ++c;
        f *= n;
        ++n;
    }
    cout << c;
    return 0;
}
```

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе чисел 1 и 15.
2. В программу ввели значение  $A = 4$ . Назовите минимальное значение  $B$ , при вводе которого программа выведет правильный ответ.
3. Найдите в программе все ошибки (известно, что их не больше двух) и исправьте их. Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.

Достаточно указать ошибки и способ их исправления для одного языка программирования.

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)
<ol style="list-style-type: none"> <li>1. При вводе чисел 1 и 15 программа выведет число 3.</li> <li>2. При <math>A = 4</math> правильный ответ выводится при минимальном <math>B = 7</math>.</li> <li>3. Программа содержит две ошибки.</li> </ol> <p><b>Первая ошибка.</b> Неверное сравнение с <math>A</math>. В задаче требуется подсчитывать факториалы, расположенные строго между заданными числами, поэтому сравнение должно быть строгим.</p> <p><b>Вторая ошибка.</b> Неверное вычисление факториала. Строки с изменением <math>F</math> и <math>N</math> переставлены, из-за этого при вычислении факториалов происходит лишнее умножение на 2 и вместо правильной последовательности 2, 6, 24... строится неверная последовательность 2, 4, 12...</p>

**Пример исправления для алгоритмического языка****Первая ошибка:**

```
если f >= a то
```

Исправленная строка:

```
если f > a то
```

**Вторая ошибка** может быть исправлена разными способами. Самое простое – поменять местами строки с вычислением  $F$  и  $N$ , но в условии разрешено только исправлять строки, поэтому приходится действовать по-другому.

**Способ 1.** Коррекция умножения.

Исходная строка:

```
f := f * n
```

Исправленная строка:

```
f := f * (n+1)
```

**Способ 2.** Коррекция начального значения  $F$ .

Исходная строка:

```
f := 2
```

Исправленная строка:

```
f := 1
```

**Способ 3.** Коррекция начального значения  $N$ .

Исходная строка:

```
n := 2
```

Исправленная строка:

```
n := 3
```

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

Незначительной опiskой, не влияющей на оценку, следует считать отсутствие служебных слов и знаков, не влияющих на содержательную часть исправления (например, слова «то» или точки с запятой).

Указания по оцениванию	Баллы
<p>В задаче требуется выполнить <b>четыре</b> действия.</p> <ol style="list-style-type: none"> <li>1. Указать, что выведет программа при конкретном вводе.</li> <li>2. Определить минимальное значение <math>B</math>, при вводе которого программа выдаёт верный ответ для заданного <math>A</math>.</li> <li>3. Исправить первую ошибку в программе.</li> <li>4. Исправить вторую ошибку в программе.</li> </ol> <p>Действие 1 считается выполненным, если указан верный результат работы программы при заданных входных значениях.</p> <p>Действие 2 считается выполненным, если указано верное значение <math>B</math>.</p> <p>Для действий 1 и 2 учащийся не обязан объяснять, как получен результат, достаточно указать верные числа.</p> <p>Каждое из действий 3 и 4 считается выполненным при одновременном выполнении двух условий:</p> <ol style="list-style-type: none"> <li>а) правильно указана строка с ошибкой;</li> <li>б) указан такой новый вариант строки, что при исправлении другой ошибки получается правильная программа</li> </ol>	

В исправленной строке допускаются незначительные синтаксические ошибки (лишние или пропущенные знаки препинания, неточные написания служебных слов языка)	
Выполнены все четыре необходимых действия, ни одна верная строка не указана в качестве ошибочной	3
Не выполнены условия, позволяющие поставить 3 балла. Имеет место одна из следующих ситуаций. 1. Выполнены три из четырёх необходимых действий, ни одна верная строка не названа ошибочной. 2. Выполнены все четыре необходимых действия, одна верная строка названа ошибочной	2
Не выполнены условия, позволяющие поставить 2 или 3 балла. Выполнены два из четырёх необходимых действий	1
Не выполнены условия, позволяющие поставить 1, 2 или 3 балла	0
<i>Максимальный балл</i>	<i>3</i>

25

Дан массив, содержащий 2019 положительных целых чисел, не превышающих 15 000. Необходимо найти количество элементов массива, шестнадцатеричная запись которых заканчивается цифрой D, заменить на это количество все элементы, восьмеричная запись которых содержит ровно 3 цифры, и вывести изменённый массив.

Напишите на одном из языков программирования программу для решения этой задачи.

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

<b>Бейсик</b>	<b>Python</b>
<pre>CONST N = 2019 DIM A(N) AS INTEGER DIM I, M, K, D AS INTEGER FOR I = 1 TO N     INPUT A(I) NEXT I ... END</pre>	<pre># кроме уже указанных # допускается использование # целочисленных переменных # m, k, d a = [] N = 2019 for i in range(0, N):     a.append(int(input())) ...</pre>
<b>Алгоритмический язык</b>	<b>Паскаль</b>
<pre>алг нач     цел N=2019     целтаб a[1:N]     цел i, m, k, d     нц для i от 1 до N         ввод a[i]</pre>	<pre>const     N=2019; var     a: array [1..N] of integer;     i, m, k, d: integer; begin     for i := 1 to N do</pre>

кц ... Кон	readln(a[i]); ... end.
<b>C++</b>	
<pre>#include &lt;iostream&gt; using namespace std; const int N=2019; int main(){     int a[N];     int i, m, k, d;     for (i=0; i&lt;N; ++i)         cin &gt;&gt; a[i];      ...     return 0; }</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 2.6). В этом случае Вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии.

<b>Содержание верного ответа</b> (допускаются иные формулировки ответа, не искажающие его смысла)	
<p>Задача решается в два прохода: на первом проходе определяется количество требуемых элементов, на втором проходе производится корректировка и вывод элементов.</p> <p>Возможно решение в три прохода, когда на втором проходе выполняется только замена значений, а на третьем – вывод</p>	
<b>Пример правильной программы на алгоритмическом языке</b>	
<pre>к := 0 нц для i от 1 до N     если mod(a[i],16)=13         то k := k + 1     все кц нц для i от 1 до N     если 64 &lt;= a[i] &lt; 512         то a[i] := k     все     вывод a[i], ' ' кц</pre>	
<p>При использовании языка Python первый проход можно записать в одну строку, используя специальные средства этого языка</p>	

**Пример правильной программы на языке Python**

```
k = sum(1 for i in a if i%16 == 13)
for i in range(0,N):
    if 64 <= a[i] < 512:
        a[i] = k
    print(a[i])
```

Использовать описанную возможность не обязательно, на языке Python допустимо описывать развёрнутый алгоритм решения, аналогичный приведённой выше программе на алгоритмическом языке.

Возможны решения, в которых производится полный подсчёт количества цифр в восьмеричной записи. Такое решение более громоздко и менее эффективно, но при отсутствии ошибок оно оценивается полным баллом. Ниже приводится пример такого решения на языке Паскаль

**Пример правильной программы на языке Паскаль**

```
k := 0;
for i:=1 to N do begin
    if a[i] mod 16 = 13 then k :=k + 1
end;
for i:=1 to N do begin
    m := a[i]; d := 0;
    while m > 0 do begin
        d := d + 1;
        m := m div 8
    end;
    if d = 3 then a[i] := k;
    write(a[i], ' ')
end
```

Указания по оцениванию	Баллы
<p>В программе допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора.</p> <p>Эффективность не имеет значения и не оценивается.</p> <p>Допускается запись программы на языке, не входящем в список языков из условия. В этом случае должны использоваться переменные, аналогичные описанным в условии. Если выбранный 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</p> <p>Допускается произвольный формат вывода полученного массива, например, вывод всех элементов массива в одну строку или вывод каждого элемента в отдельной строке. Отмечается как ошибка, но не учитывается при выставлении оценки вывод элементов в одну строку без пробелов между ними</p>	

Предложена правильная программа, которая изменяет исходный массив в соответствии с условием и выводит изменённый массив	2
<p>Не выполнены условия, позволяющие поставить 2 балла.</p> <p>Предложено в целом верное решение, содержащее не более одной ошибки из числа следующих (если одинаковая ошибка повторяется несколько раз, она считается за одну ошибку).</p> <ol style="list-style-type: none"> <li>1) Отсутствие инициализации или неверная инициализация счётчика.</li> <li>2) Неверное определение последней цифры в шестнадцатеричной записи.</li> <li>3) Неверное определение количества цифр в восьмеричной записи.</li> <li>4) Неверное построение логических условий (неверные логические операции, проверка не всех условий).</li> <li>5) Выход за границы массива.</li> <li>6) Исходный массив не изменяется.</li> <li>7) Изменяются не все элементы, которые должны измениться, или изменяются элементы, которые не должны измениться.</li> <li>8) Полученный массив не выводится или выводится не полностью (например, выводится только первый элемент или только изменённые элементы).</li> <li>9) Используется переменная, не объявленная в разделе описания переменных.</li> <li>10) Индексная переменная в цикле не меняется (например, в цикле <code>while</code>) или меняется неверно</li> </ol>	1
Не выполнены условия, позволяющие поставить 1 или 2 балла	0
Максимальный балл	2

26

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может **добавить** в одну из куч **один камень** или **увеличить** количество камней **в куче в четыре раза**. Например, пусть в одной куче 6 камней, а в другой 9 камней; такую позицию мы будем обозначать (6, 9). За один ход из позиции (6, 9) можно получить любую из четырёх позиций: (7, 9), (24, 9), (6, 10), (6, 36). Чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 61. Победителем считается игрок, сделавший последний ход, то есть первым получивший позицию, в которой в кучах будет 61 или больше камней.

В начальный момент в первой куче было 3 камня, во второй куче –  $S$  камней,  $1 \leq S \leq 57$ .

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. В описание выигрышной стратегии не следует включать ходы играющего по ней игрока, которые не являются для него безусловно выигрышными, то есть не гарантируют выигрыш независимо от игры противника.

Выполните следующие задания.

### Задание 1.

- Назовите все значения  $S$ , при которых Петя может выиграть первым ходом.
- Петя сделал неудачный первый ход, после которого Ваня выиграл своим первым ходом. Назовите минимальное значение  $S$ , при котором это возможно.

### Задание 2.

Укажите такое значение  $S$ , при котором у Пети есть выигрышная стратегия, причём Петя не может выиграть первым ходом, но может выиграть своим вторым ходом независимо от того, как будет ходить Ваня. Для указанного значения  $S$  опишите выигрышную стратегию Пети.

### Задание 3.

Укажите такое значение  $S$ , при котором у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, и при этом 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). В узлах дерева указывайте игровые позиции. Дерево не должно содержать партий, невозможных при реализации выигрывающим игроком своей выигрышной стратегии. Например, полное дерево игры не будет верным ответом на это задание.

Содержание верного ответа (допускаются иные формулировки ответа, не искажающие его смысла)
<p><b>Задание 1.</b></p> <p>а) Петя может выиграть первым ходом, если <math>S = 15, \dots, 57</math>. Для выигрыша достаточно умножить количество камней во второй куче. При меньших значениях <math>S</math> за один ход нельзя получить 61 или более камней в двух кучах.</p> <p>б) Такая ситуация возможна при <math>S = 4</math>. Если Петя умножит вторую кучу, получится позиция (3,16), из которой Ваня может получить позицию (3, 64) и выиграть. При <math>S &lt; 4</math> никакой первый ход Пети не создаст ситуацию, в которой Ваня может сразу выиграть.</p> <p><b>Задание 2.</b></p> <p>Возможные значения <math>S</math>: 12, 14. В этих случаях Петя, очевидно, не может выиграть первым ходом. Однако при <math>S = 12</math> Петя может получить позицию (12, 12), а при <math>S = 14</math> – позицию (4, 14).</p> <p>В первом случае после хода Вани возникнет одна из позиций (13, 12), (48, 12), (12, 13), (12, 48), во втором случае – одна из позиций (5, 14), (16, 14), (4, 15), (4, 56). В любой из перечисленных позиций Петя может выиграть, умножив количество камней во второй куче</p>



**Задание 3.**

Возможное значение  $S$ : 13. После первого хода Пети возможны позиции (4, 13), (12, 13), (3, 14), (3, 52). В позициях (12, 13) и (3, 52) Ваня может выиграть первым ходом, умножив количество камней в любой куче. Из позиций (4, 13) и (3, 14) Ваня может получить позицию (4, 14), разобранный в задании 2. Игрок, после хода которого возникла эта позиция (в данном случае – Ваня), выигрывает следующим ходом.

В таблице изображены возможные партии при описанной стратегии Вани. Заключительные позиции (в них выигрывает Ваня) выделены жирным шрифтом. На рисунке эти же партии показаны в виде графа (оба способа изображения допустимы).

	Положения после очередных ходов			
Исходное положение	1-й ход Пети (разобраны все ходы, указана полученная позиция)	1-й ход Вани (только ход по стратегии, указана полученная позиция)	2-й ход Пети (разобраны все ходы, указана полученная позиция)	2-й ход Вани (только ход по стратегии, указана полученная позиция)
(3, 13) Всего 16	(12, 13) Всего 25	(12, 52) <b>Всего 64</b>		
	(3, 52) Всего 55			
	(4, 13) Всего 17	(4, 14) Всего 18	(5, 14) Всего 19	(5, 56) <b>Всего 61</b>
	(3, 14) Всего 17		(16, 14) Всего 34	(16, 56) <b>Всего 72</b>
			(4, 56) Всего 31	
			(4, 15) Всего 74	(4, 60) <b>Всего 64</b>

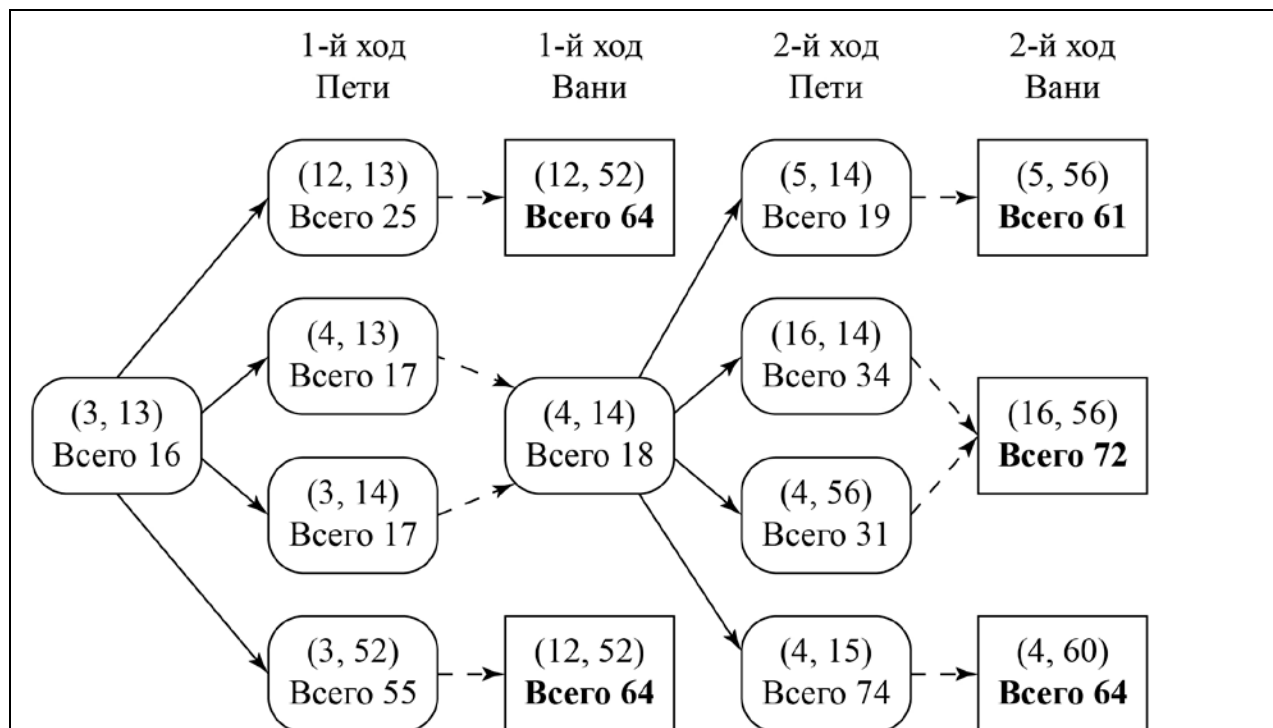


Рис. 1. Граф всех партий, возможных при описанной стратегии Вани. Ходы Пети показаны сплошными стрелками, ходы Вани – пунктирными стрелками. Заключительные позиции обозначены прямоугольниками.

*Примечание для эксперта.* Дерево всех партий может быть изображено в виде таблицы или ориентированного графа – так, как показано на рисунке, или другим способом. Например, вместо приведённого здесь «экономного» варианта, в котором позиции не дублируются, возможно построение полного дерева, в котором одинаковые позиции, возникающие при различном ходе игры, показаны отдельно. Важно, чтобы множество полных путей в графе находилось во взаимно однозначном соответствии с множеством партий, возможных при описанной в решении стратегии. В некоторых позициях заключительный выигрывающий ход можно сделать несколькими способами. В таблице и на рисунке указан один из них, в работе допускается выбор любого допустимого заключительного выигрывающего хода.

Указания по оцениванию	Баллы
<p>В задаче от ученика требуется выполнить <b>три</b> задания. 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>Задание 1 выполнено, если выполнены оба пункта: для пункта (а) перечислены все удовлетворяющие условию значения <math>S</math>, и только они, для пункта (б) указано верное значение <math>S</math>, и только оно. Обоснование найденных значений не обязательно</p>	

Задание 2 выполнено, если верно указана выигрышная для Пети позиция (любая из двух возможных) и описана соответствующая стратегия. Задание 3 выполнено, если правильно указана выигрышная для Вани позиция и построено дерево всех возможных при выигрышной стратегии партий (и только их). Во всех случаях стратегии могут быть описаны так, как это сделано в примере решения, или другим способом	
Выполнены все три задания	3
Не выполнены условия, позволяющие поставить 3 балла, и выполнено хотя бы одно из следующих условий. – Выполнено задание 3. – Выполнены задания 1 и 2	2
Не выполнены условия, позволяющие поставить 2 или 3 балла, и выполнено хотя бы одно из заданий 1 и 2	1
Не выполнено ни одно из условий, позволяющих поставить 1, 2 или 3 балла	0
<i>Максимальный балл</i>	<i>3</i>

27

Дана последовательность  $N$  целых неповторяющихся положительных чисел. Рассматриваются все пары элементов последовательности, разность которых делится на  $m = 60$ . Среди всех таких пар нужно найти и вывести пару с максимальной разностью элементов. Если одинаковую максимальную разность имеют несколько пар, можно вывести любую из них. Если подходящих пар в последовательности нет, нужно вывести два нуля.

#### Описание входных и выходных данных

В первой строке входных данных задаётся количество чисел  $N$  ( $2 \leq N \leq 10\,000$ ). В каждой из последующих  $N$  строк записано одно натуральное число, не превышающее 10 000. Гарантируется, что никакое число не встречается в последовательности более одного раза.

*Пример входных данных:*

8  
75  
123  
5  
40  
15  
3  
65  
80

*Пример выходных данных для приведённого выше примера входных данных:*

3 123

*Пояснение.* Из данных восьми чисел можно составить три пары, удовлетворяющие условию: (15, 75), (3, 123), (5, 65). Наибольшая разность получается в паре (3, 123).

Напишите эффективную по времени и по памяти программу для решения этой задачи.

Программа считается эффективной по времени, если при одновременном увеличении количества исходных чисел  $N$  и параметра  $m$  в  $k$  раз время работы программы увеличивается не более чем в  $k$  раз.

Программа считается эффективной по памяти, если память, необходимая для хранения всех переменных программы, не превышает 4 Кбайт и не увеличивается с ростом  $N$ .

Максимальная оценка за правильную (не содержащую синтаксических ошибок и дающую правильный ответ при любых допустимых входных данных) программу, эффективную по времени и по памяти, – 4 балла.

Максимальная оценка за правильную программу, эффективную только по времени или только по памяти, – 3 балла.

Максимальная оценка за правильную программу, не удовлетворяющую требованиям эффективности, – 2 балла.

Вы можете сдать **одну** или **две** программы решения задачи. Если Вы сдадите две программы, каждая из них будет оцениваться независимо от другой, итоговой станет **бóльшая** из двух оценок.

Перед текстом программы кратко опишите алгоритм решения. Укажите использованный язык программирования и его версию.

<p align="center"><b>Содержание верного ответа</b> (допускаются иные формулировки ответа, не искажающие его смысла)</p>
<p>Разность двух чисел кратна <math>m</math>, остатки от деления этих чисел на <math>m</math> равны. При этом для получения максимальной разности нужно, чтобы одно из чисел было как можно больше, а второе как можно меньше.</p> <p>Будем хранить в одном массиве из <math>m</math> элементов максимальные числа, имеющие соответствующий остаток от деления на <math>m</math>, а в другом – минимальные, и из всех пар максимумов и минимумов выберем пару с наибольшей разностью.</p> <p>При этом нужно убедиться, что это именно пара. Поскольку в условии гарантируется, что числа в последовательности не повторяются, если минимум оказался равен максимуму, значит, пары на самом деле нет.</p> <p>Ниже приведена программа на алгоритмическом языке, реализующая этот алгоритм</p>
<p><b>Решение 1. Правильная и эффективная программа на алгоритмическом языке</b></p>

```

алг задача27
нач
    цел m = 60
    цел таб mn[0:m-1], mx[0:m-1]
    цел N
    цел x | очередное число из последовательности
    цел p | остаток
    цел pm | остаток, дающий лучшую разность
    цел i

    нц для i от 0 до m-1
        mn[i] := 0; mx[i] := 0
    кц
    pm := 0
    ввод N
    нц N раз
        ввод x
        p:=mod(x,m)
        если mn[p] = 0 или x < mn[p]
            то mn[p] := x
        все
        mx[p] := imax(mx[p],x)
        если mx[p]-mn[p] > mx[pm]-mn[pm]
            то pm := p
        все
    кц
    если mx[pm] = mn[pm]
        то вывод '0 0'
    иначе вывод mn[pm], ' ', mx[pm]
все
кон

```

В приведённом решении лучшая пара находится «на лету»: после обработки каждого числа из последовательности проверяется, не увеличилась ли возможная разность.

Эту проверку можно провести после завершения ввода, вычислив разности для всех остатков и сравнив их. Такая программа тоже эффективна по времени и по памяти, а при больших  $N$  она работает даже быстрее предыдущей, так как выполняет меньше вычислений и сравнений.

Ниже приведена реализующая этот алгоритм программа на языке Python

## Решение 2. Правильная и эффективная программа на языке Python

```

m = 60
mn = [0] * m
mx = [0] * m
N = int(input())
for i in range(N):
    x = int(input())
    p = x % m
    if mn[p] == 0 or x < mn[p]:
        mn[p] = x
    if x > mx[p]:

```

```

        mx[p] = x
p = 0
for i in range(m):
    if mx[i]-mn[i] > mx[p]-mn[p]:
        p = i
if mx[p] == mn[p]:
    print(0,0)
else:
    print(mn[p], mx[p])

```

Возможно также «любое» решение: запишем все исходные числа в массив, переберём все возможные пары и выберем подходящую. Такое решение не является эффективным ни по памяти (требуемая память зависит от размера исходных данных), ни по времени (количество возможных пар, а значит, количество действий и время счёта с ростом количества исходных элементов растёт квадратично). Подобная программа оценивается не выше 2 баллов.

Ниже приведена реализующая описанный выше алгоритм программа на языке Паскаль (использована версия PascalABC)

### Решение 3. Правильная, но неэффективная программа на языке Паскаль

```

const m=60;
var
    N: integer;      {количество чисел}
    a: array [1..10000] of integer; {исходные данные}
    x1, x2: integer; {ответ - пара чисел}
    i, j: integer;

begin
    readln(N);
    for i:=1 to N do readln(a[i]);
    x1 := 0; x2 := 0;
    for i := 1 to N do begin
        for j := i+1 to N do begin
            if ((a[i] - a[j]) mod m = 0) and
                (abs(a[i]-a[j]) > abs(x1-x2))
            then begin
                x1 := a[i]; x2 := a[j]
            end
        end
    end
    writeln(x1, ' ', x2)
end.

```

Указания по оцениванию	Баллы
Если в работе представлены две программы решения задачи, то каждая из них независимо оценивается по указанным ниже критериям, итоговой считается бóльшая из двух оценок. Описание алгоритма решения без программы оценивается 0 баллов	
Программа правильно работает для любых входных данных произвольного размера. Используемая память не зависит от количества прочитанных чисел $N$ , время работы пропорционально $N$ и $m$ .	4

<p>Допускается наличие в тексте программы до трёх синтаксических ошибок одного из следующих видов:</p> <ol style="list-style-type: none"> <li>1) пропущен или неверно указан знак пунктуации;</li> <li>2) неверно написано, пропущено или написано лишнее зарезервированное слово языка программирования;</li> <li>3) не описана или неверно описана переменная;</li> <li>4) применяется операция, недопустимая для соответствующего типа данных.</li> </ol> <p>Если одна и та же ошибка встречается несколько раз, это считается за одну ошибку</p>	
<p>Не выполнены условия, позволяющие поставить 4 балла.</p> <p>Программа в целом работает правильно для любых входных данных произвольного размера. Время работы пропорционально количеству введённых чисел <math>N</math> и параметру <math>m</math>.</p> <p>Используемая память, возможно, зависит от количества прочитанных чисел (например, входные данные запоминаются в массиве или другой аналогичной структуре данных)</p> <p>Количество синтаксических ошибок («описок»), указанных в критериях на 4 балла, – не более пяти.</p> <p>Допускается наличие не более одной ошибки следующих видов:</p> <ol style="list-style-type: none"> <li>1) ошибка при вводе данных (не считается значение <math>N</math> или неверно организован ввод последовательности);</li> <li>2) ошибка при инициализации или отсутствие инициализации там, где она необходима;</li> <li>3) используется неверный тип данных;</li> <li>4) использована одна переменная (константа) вместо другой;</li> <li>5) используется один знак операции вместо другого;</li> <li>6) отсутствует вывод ответа или выводится не то значение (например, выводится разность вместо пары чисел);</li> <li>7) неверная работа с массивом, в том числе выход за границы массива;</li> <li>8) пропущены или неверно расставлены операторные скобки (при использовании языков с операторными скобками);</li> <li>9) неверно обрабатывается ситуация, когда в массиве нет подходящих пар</li> </ol>	3
<p>Не выполнены условия, позволяющие поставить 3 или 4 балла, при этом программа работает в целом верно и эффективно по времени. Допускается наличие до трёх содержательных ошибок, описанных в критериях на 3 балла, и до девяти синтаксических ошибок, описанных в критериях на 4 балла.</p> <p>ИЛИ</p> <p>Программа работает в целом верно и эффективно по времени, но в ней есть не подходящие под перечень из критериев на 3 балла ошибки, которые в некоторых особых случаях приводят к неверным результатам</p>	2

ИЛИ Представлено корректное переборное решение, в котором все исходные данные сохраняются в массиве (или другой аналогичной структуре) и рассматриваются все возможные пары. При этом не допускаются содержательные логические ошибки, например выход индексов за границы массива, рассмотрение произведений вида $a[i]*a[i]$ и т. д.	
Не выполнены условия, позволяющие поставить 2, 3 или 4 балла. При этом программа представлена и содержит как минимум два обязательных элемента, возможно, реализованных с ошибками: 1) рассматриваются пары, с подходящей суммой; 2) обнаруживается и выводится пара с максимальным произведением	1
Не выполнены условия, позволяющие поставить 1, 2, 3 или 4 балла	0
<i>Максимальный балл</i>	4