



## Разность

Дан массив из  $N$  неотрицательных целых чисел  $a_1, a_2, \dots, a_N$ , удовлетворяющих неравенству  $0 \leq a_1 < a_2 < \dots < a_N \leq 10^{18}$ . Джихак хочет узнать наибольшее возможное значение  $a_{i+1} - a_i$ , где  $i$  меняется 1 до  $N - 1$ . Массив не будет дан программе Джихака напрямую, но к нему можно получить доступ с помощью специальной функции. Смотри раздел Требования для пишущих на C, C++ или Pascal соответственно.

## Gap

There are  $N$  non-negative integers  $a_1, a_2, \dots, a_N$  satisfying the following inequality  $0 \leq a_1 < a_2 < \dots < a_N \leq 10^{18}$ . Jeehak wants to know the *largest possible* value of  $a_{i+1} - a_i$  where  $i$  ranges from 1 to  $N - 1$ . The input integers will not be given directly to Jeehak's program but will be accessible through a special function. See sections Implementation of your selected programming language for details.

## Условие

Помогите Джихаку реализовать функцию, которая возвращает максимальное возможное значение  $a_{i+1} - a_i$ , где  $i$  меняется от 1 до  $N - 1$ .

## Task

Help Jeehak to implement a function to return the largest possible value of  $a_{i+1} - a_i$  where  $i$  ranges from 1 to  $N - 1$ .

## Требования для пишущих на C и C++

Вам нужно реализовать функцию `findGap(T, N)`, которая возвращает целое число типа `long long` и принимает следующие параметры:

- $T$  — номер подзадачи (1 или 2)
- $N$  — количество чисел в массиве

Ваша функция `findGap` может вызывать `MinMax(s, t, &mn, &mx)`, где первые два параметра  $s$  и  $t$  - целые числа типа `long long`, а последние два параметра `&mn` и `&mx` - ссылки на целочисленные переменные типа `long long`, т.е. `mn` и `mx` - целочисленные переменные типа `long long`. После вызова `MinMax(s, t, &mn, &mx)` в переменной `mn` лежит значение наименьшего  $a_i$ , большего или равного  $s$ , а в `mx` хранится значение наибольшего  $a_j$ , меньшего или равного  $t$ . В случае, если среди чисел массива нет чисел в отрезке между  $s$  и  $t$  включительно, тогда в обеих переменных `mn` и `mx` лежит `-1`.  $s$  должно быть не меньше  $t$  при вызове `MinMax`. Если это условие не будет выполнено, тогда программа завершится ненулевым кодом ошибки.

## Implementation for C and C++

You need to implement one function `findGap(T, N)` that takes the following parameter and returns an integer of type `long long`:

- `T` — the subtask number (1 or 2)
- `N` — the number of given integers

Your function `findGap` can call function `MinMax(s, t, &mn, &mx)` where the first two parameters `s` and `t` are integers of type `long long` and the last two parameters `&mn` and `&mx` are pointers to integer variables of type `long long`, i.e., `mn` and `mx` are integer variables of type `long long`. When `MinMax(s, t, &mn, &mx)` returns, the variable `mn` will have the value of smallest  $a_i$  larger than or equal to the value of `s` and the variable `mx` will have the value of largest  $a_j$  smaller than or equal to the value of `t`. In case there are no input integers between `s` and `t` (inclusive), then both `mn` and `mx` will have the value `-1`. The value of `s` should be no larger than the value of `t` when `MinMax` is called. If this condition is not met, program will be terminated with a non-zero exit code.

## Требования для пишущих на Pascal

Вам нужно реализовать функцию `findGap(T, N)`, которая возвращает целое число типа `Int64` и принимает следующие параметры:

- `T` — номер подзадачи (1 или 2) (типа `Integer`)
- `N` — количество чисел в массиве (типа `LongInt`)

Ваша функция `findGap` может вызывать процедуру `MinMax(s, t, mn, mx)`, где первые два параметра `s` и `t` - целые числа типа `Int64`, а последние два параметра `mn` и `mx` - переменные, **обращение к которым происходит по ссылке**, типа `Int64`, т.е. `mn` и `mx` - целочисленные переменные типа `Int64`. После вызова `MinMax(s, t, mn, mx)` в переменной `mn` лежит значение наименьшего  $a_i$ , большего или равного `s`, а в `mx` хранится значение наибольшего  $a_j$ , меньшего или равного `t`. В случае, если среди чисел массива нет чисел в отрезке между `s` и `t` включительно, тогда в обеих переменных `mn` и `mx` лежит `-1`. `s` должно быть не меньше `t` при вызове `MinMax`. Если это условие не будет выполнено, тогда программа завершится.

## Implementation for Pascal

You need to implement one function `findGap(T, N)` that takes the following parameter and returns an integer of type `Int64`:

- `T` — the subtask number (1 or 2) (`Integer` type)
- `N` — the number of given integers (`LongInt` type)

Your function `findGap` can call procedure `MinMax(s, t, mn, mx)` where the first two parameters `s` and `t` are integers of type `Int64` and the last two parameters `mn` and `mx` are variables **called by reference** of type `Int64`, i.e., `mn` and `mx` are integer variables of type `Int64`. When `MinMax(s, t, mn, mx)` exits, the variable `mn` will have the value of smallest  $a_i$  larger than or equal to the value of `s` and the variable `mx` will have the value of largest  $a_j$  smaller than or equal to the value of `t`. In case there are no input integers between `s` and `t` (inclusive), then both `mn` and `mx` will have the value `-1`. The value of `s` should be no larger than the value of `t` when `MinMax` is called. If this condition is not met, the program will be terminated.

## Требования для всех

Помимо стандартных требований (ограничение по времени и по памяти, отсутствие ошибок при выполнении и т.д.), чтобы пройти набор тестов, ваша программа должна удовлетворять следующим условиям:

- ваша функция `findGap` должна возвращать правильный ответ,
- величина  $M$ , равная количеству вызовов `MinMax`, не должна превышать разрешенного значения (смотри раздел Система оценивания).

## Implementation for all

In addition to the standard requirements (time and memory limits, no runtime errors, etc), your submission has to achieve the following in order to solve a testcase:

- your function `findGap` must return the correct answer,
- the cost  $M$  associated with calls to function `MinMax` must not exceed the allowed limit (see section Scoring).

## Пример для C, C++

Рассмотрим случай, когда  $N = 4$  и  $a_1 = 2, a_2 = 3, a_3 = 6$ , и  $a_4 = 8$ .

Ответ, равный **3**, можно получить, если делать вызовы `MinMax` в следующем порядке:

- После вызова `MinMax(1, 2, &mn, &mx)` в `mn` и `mx` хранится **2**.
- После вызова `MinMax(3, 7, &mn, &mx)` в `mn` лежит **3**, а в `mx` хранится число **6**.
- После вызова `MinMax(8, 9, &mn, &mx)` в `mn` и `mx` лежит **8**.

## Example for C, C++

Consider the case where  $N = 4$  and  $a_1 = 2, a_2 = 3, a_3 = 6$ , and  $a_4 = 8$ .

The answer, which is **3**, can be calculated and thus returned by `findGap` if the following calls to `MinMax` are made:

- `MinMax(1, 2, &mn, &mx)` is called and `mn` and `mx` both have the value **2**.
- `MinMax(3, 7, &mn, &mx)` is called and `mn` have the value **3** and `mx` has the value **6**.
- `MinMax(8, 9, &mn, &mx)` is called and `mn` and `mx` both have the value **8**.

## Пример для Pascal

Рассмотрим случай, когда  $N = 4$  и  $a_1 = 2, a_2 = 3, a_3 = 6$ , и  $a_4 = 8$ .

Ответ, равный **3**, можно получить, если делать вызовы `MinMax` в следующем порядке:

- После вызова `MinMax(1, 2, mn, mx)` в `mn` и `mx` хранится **2**.

- После вызова  $\text{MinMax}(3, 7, \text{mn}, \text{mx})$  в  $\text{mn}$  лежит **3**, а в  $\text{mx}$  хранится число **6**.
- После вызова  $\text{MinMax}(8, 9, \text{mn}, \text{mx})$  в  $\text{mn}$  и  $\text{mx}$  лежит **8**.

## Example for Pascal

Consider the case where  $N = 4$  and  $a_1 = 2, a_2 = 3, a_3 = 6$ , and  $a_4 = 8$ .

The answer, which is **3**, can be calculated and thus returned by `findGap` if the following calls to `MinMax` are made:

- $\text{MinMax}(1, 2, \text{mn}, \text{mx})$  is called and  $\text{mn}$  and  $\text{mx}$  both have the value **2**.
- $\text{MinMax}(3, 7, \text{mn}, \text{mx})$  is called and  $\text{mn}$  have the value **3** and  $\text{mx}$  has the value **6**.
- $\text{MinMax}(8, 9, \text{mn}, \text{mx})$  is called and  $\text{mn}$  and  $\text{mx}$  both have the value **8**.

## Система оценивания

Во всех подзадачах  $2 \leq N \leq 100,000$ .

**Подзадача 1 (30 баллов):** Каждый вызов `MinMax` добавляет **1** к  $M$ . Вы получите все баллы за подзадачу, если  $M \leq \frac{N+1}{2}$  на всех тестах подзадачи.

**Подзадача 2 (70 баллов):** Пусть  $k$  равно количеству чисел больших или равных  $s$  и меньших или равных  $t$  при вызове `MinMax`. Каждый вызов `MinMax` добавит  $k + 1$  к  $M$ . Количество баллов, полученное за подзадачу, равно минимуму из баллов, набранных за каждый тест. Количество баллов, полученное за тест, равно **70**, если  $M \leq 3N$ , и равно  $\frac{60}{\sqrt{\frac{M}{N}+1}-1}$ , иначе.

## Scoring

In all subtasks the constraint  $2 \leq N \leq 100,000$  holds.

**Subtask 1 (30 points):** Each call to `MinMax` will add **1** to  $M$ . You will receive the full score for the subtask if  $M \leq \frac{N+1}{2}$  for all test cases.

**Subtask 2 (70 points):** Let  $k$  be the number of input integers larger than or equal to  $s$  and smaller than or equal to  $t$  in a call to `MinMax`. Each call to `MinMax` will add  $k + 1$  to  $M$ . The final score will be calculated by the following rule: Final score for the subtask is the minimum score you received among all test cases. For a test case, the score is **70** if  $M \leq 3N$  and the score is  $\frac{60}{\sqrt{\frac{M}{N}+1}-1}$ ,

otherwise.

## Тестирование на семплах

Систему для проверки на семплах можете скачать из проверяющей системы. Система для проверки на семплах считывает данные со стандартного потока. Первая строка ввода должна содержать два целых числа номер подзадачи  $T$  и  $N$ . В следующей строке должны содержаться  $N$  целых чисел в возрастающем порядке. Система проверки на семплах записывает в стандартный поток вывода в одной строке значение, возвращаемое функцией `findGap`, и значение  $M$ , соответствующее тесту.

В следующих строках описан ранее упомянутый пример:

```
2 4  
2 3 6 8
```

## Experimentation

The sample grader which can be downloaded from the scoring system will read data from standard input. The first line of input should contain two integers, subtask number  $T$ , and  $N$ . The next line should contain  $N$  integers in ascending order. The sample grader will write to standard output the value returned by `findGap` in the first line and the value of  $M$  appropriate for the subtask the input test case belongs to.

The following input describes the above example:

```
2 4  
2 3 6 8
```