



ギャップ (Gap)

N 個の非負整数 a_1, a_2, \dots, a_N が不等式 $0 \leq a_1 < a_2 < \dots < a_N \leq 10^{18}$ を満たしている。ジハク (Jeehak) は、 i が 1 から $N - 1$ まで動くときの、 $a_{i+1} - a_i$ の取り得る値の最大値を知りたい。入力された整数値はジハクのプログラムには直接は渡されず、特別な関数を通してのみアクセスすることができる。詳しくは、あなたの選択したプログラミング言語における実装 (Implementation) の節を見よ。

課題 (Task)

ジハク (Jeehak) が、 i が 1 から $N - 1$ まで動くときの、 $a_{i+1} - a_i$ の取り得る値の最大値を返す関数を実装するのを助けよ。

C/C++ の実装 (Implementation for C and C++)

以下のパラメータを取り、long long 型の整数値を返す関数 findGap(T, N) を実装せよ:

- T — 小課題番号 (1 または 2)
- N — 与えられる整数の個数

関数 findGap の中では、関数 MinMax(s, t, &mn, &mx) を呼び出すことができる。最初の 2 つのパラメータ s と t は long long 型の整数であり、残りの 2 つのパラメータ &mn と &mx は long long 型の整数変数へのポインタである。つまり、mn と mx は long long 型の整数変数である。関数 MinMax(s, t, &mn, &mx) が実行された後、変数 mn には a_i の値のうち s 以上であるものの最小値が格納され、変数 mx には a_j の値のうち t 以下であるものの最大値が格納される。ただし、もし入力された整数値のなかに s 以上 t 以下のものが存在しない場合は、mn と mx の値は両方とも -1 になる。MinMax の呼び出しの際は、s の値は t の値以下でなければならない。この条件が満たされない場合、0 でない終了コードとともにプログラムは終了される。

Pascal の実装 (Implementation for Pascal)

以下のパラメータを取り、Int64 型の整数値を返す関数 findGap(T, N) を実装せよ:

- T — 小課題番号 (1 または 2) (Integer 型)
- N — 与えられる整数の個数 (LongInt 型)

関数 findGap の中では、プロシージャ MinMax(s, t, mn, mx) を呼び出すことができる。最初の 2 つのパラメータ s と t は Int64 型の整数であり、残りの 2 つのパラメータ mn と mx は Int64 型の参照渡し (called by reference) 変数である。つまり、mn と mx は Int64 型の整数変数である。MinMax(s, t, mn, mx) が実行された後、変数 mn には a_i の値のうち s 以上であるものの最小値が格納され、変数 mx には a_j の値のうち t 以下であるものの最大値が格納される。ただし、もし入力された整数値のなかに s 以上かつ t 以下のものが存在しない場合は、mn と mx の値は両方とも -1 になる。MinMax の呼び出しの際は、s の値は t の値以下でなければならない。この条件が満たされない場合、プログラムは終了される。

全てのプログラミング言語における実装時の注意

(Implementation for all)

テストケースを解くためには、実行時間制限やメモリ制限を満たす、ランタイムエラーが無い等の通常の要請に加えて、あなたのプログラムは以下の要請を満たさなければならない:

- 関数 `findGap` は正しい答えを返さなければならない.
- 関数 `MinMax` の呼び出しに付随するコスト M が、設定された上限値を超えてはならない (採点の節を見よ).

C/C++ の例 (Example for C, C++)

$N = 4, a_1 = 2, a_2 = 3, a_3 = 6, a_4 = 8$ の場合を考える.

答えは **3** である. 以下のように `MinMax` を呼び出すことで, `findGap` はこの答えを計算して返すことができる:

- `MinMax(1, 2, &mn, &mx)` を呼び出す. `mn` と `mx` の値は両方とも **2** になる.
- `MinMax(3, 7, &mn, &mx)` を呼び出す. `mn` の値は **3** になり, `mx` の値は **6** になる.
- `MinMax(8, 9, &mn, &mx)` を呼び出す. `mn` と `mx` の値は両方とも **8** になる.

Pascal の例 (Example for Pascal)

$N = 4, a_1 = 2, a_2 = 3, a_3 = 6, a_4 = 8$ の場合を考える.

答えは **3** である. 以下のように `MinMax` を呼び出すことで, `findGap` はこの答えを計算して返すことができる:

- `MinMax(1, 2, mn, mx)` を呼び出す. `mn` と `mx` の値は両方とも **2** になる.
- `MinMax(3, 7, mn, mx)` を呼び出す. `mn` の値は **3** になり, `mx` の値は **6** になる.
- `MinMax(8, 9, mn, mx)` を呼び出す. `mn` と `mx` の値は両方とも **8** になる.

採点方法 (Scoring)

全ての小課題において, $2 \leq N \leq 100,000$ を満たす.

小課題 1 (30 点): `MinMax` が 1 回呼び出されるごとに, M の値が **1** だけ加算される. 全てのテストケースにおいて $M \leq \frac{N+1}{2}$ が満たされた場合に, この小課題で満点が得られる.

小課題 2 (70 点):

`MinMax` の呼び出しにおいて, 入力された整数値のうち s 以上かつ t 以下のものの個数を k とおく. `MinMax` の各呼び出しにおいて, M の値が $k+1$ だけ加算される. 最終的な得点は次の公式により計算される: この小課題の最終的な得点は, 各テストケースの得点の最小値である. 各テストケースの得点は, $M \leq 3N$ のときは 70 点であり, そうでないときは $\frac{60}{\sqrt{\frac{M}{N}+1}-1}$ 点である.

実験 (Experimentation)

採点プログラムのサンプルが、競技システムからダウンロードできる。採点プログラムのサンプルは、標準入力からデータを読み込む。入力の1行目は、小課題番号 T と N の2つの整数を含むこと。次の行は昇順に並べられた N 個の整数を含むこと。採点プログラムのサンプルは、標準出力の1行目に `findGap` によって返された値を書き出す。また、入力テストケースの小課題における M の値を書き出す。

以下の入力は、上に挙げた例を表す:

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