



Gap

現有 N 個非負整數 a_1, a_2, \dots, a_N ，且 $0 \leq a_1 < a_2 < \dots < a_N \leq 10^{18}$ 。Jeehak 想知道 $a_{i+1} - a_i$ 的最大值是甚麼，其中 i 的範圍是從 1 至 $N - 1$ 。但這些輸入整數是不會直接給予 Jeehak 的程。這些整數是要通過系統提供的一個特別函數來測知的。具體函數的運用請參閱下面對不同編程語言的相關實現說明。

Task 任務

請幫 Jeehak 編寫一個函數以找出並返回在 i 由 1 至 $N - 1$ 的範圍內 $a_{i+1} - a_i$ 的最大值。

Implementation for C and C++ (C/C++ 的程式實現方法)

你需要編寫一個函數 `findGap(T, N)`，它將接受以下的一些參數並返回一個類型為 `long long` 的返回值：

- T — 子任務號碼 (1 或 2)
- N — 給定的輸入整數的數目

你的函數 `findGap` 可以調用系統提供的一個函數 `MinMax(s, t, &mn, &mx)` 其中第一及第二個參數 s 和 t 是 `long long` 類型的整數，而最後兩個參數 `&mn` 和 `&mx` 則是指向 `long long` 類型整數的指針，換言之， mn 及 mx 是 `long long` 類型的整數。當 `MinMax(s, t, &mn, &mx)` 返回時， mn 將含有所有 a_i 中最小的但又大於或等於 s 的一個數，同時 mx 將含有所有 a_i 中最大的但又小於或等於 t 的一個數。若沒有任何整數是在 s 到 t 的範圍內時，則 mn 和 mx 的返回值均會是 -1 。當調用函數 `MinMax` 時， s 的值應不大於 t 的值，若不能滿足這個條件的話，程式將終止運行，而程式的返回值將會為一個非零的數值。

Implementation for Pascal (Pascal 的程式實現方法)

你需要編寫一個函數 `findGap(T, N)`，它將接受以下的一些參數並返回一個類型為 `Int64` 的返回值：

- T — 子任務號碼 (1 或 2) (類型為 `Integer`)
- N — 給定的輸入整數的數目 (類型為 `LongInt`)

你的函數 `findGap` 可以調用系統提供的一個函數 `MinMax(s, t, mn, mx)` 其中第一及第二個參數 s 和 t 是 `Int64` 類型的整數，而最後兩個參數 mn 和 mx 則是通過引用調用的 `Int64` 類型的變量，即 mn 及 mx 是 `Int64` 類型的整數。當 `MinMax(s, t, mn, mx)` 返回時， mn 將含有所有 a_i 中最小的但又大於或等於 s 的一個數，同時 mx 將含有所有 a_i 中最大的但又小於或等於 t 的一個數。若沒有任何整數是在 s 到 t 的範圍內時，則 mn 和 mx 的返回值均會是 -1 。當調用函數 `MinMax` 時， s 的值應不大於 t 的值，若不能滿足這個條件的話，程式將終止運行，而程式的返回值將會為一個非零的數值。

Implementation for all (所有語言的實現方法)

除了一般標準的限制 (如記憶體，運行時間及運行時沒有錯誤) 外，你提交的程式必須要做到以下的要求才算是解決了一個測試點：

- 你的函數 `findGap` 必定要返回正確的答案，
- 因調用函數 `MinMax` 而產生的成本 M 必須不超過容許的上限 (見計分方法中的詳細說明)

Example for C, C++ (C/C++ 的例子)

假設 $N = 4$ ， $a_1 = 2, a_2 = 3, a_3 = 6, a_4 = 8$ 。

其正確答案是 3，這個由 `findGap` 計算並返回的答案是可以經由下面調用 `MinMax` 的步驟而求得的：

- 調用 `MinMax(1, 2, &mn, &mx)` 得到 `mn` 及 `mx` 均為 2
- 調用 `MinMax(3, 7, &mn, &mx)` 得到 `mn` 為 3 而 `mx` 為 6
- 調用 `MinMax(8, 9, &mn, &mx)` 得到 `mn` 及 `mx` 均為 8.

Example for Pascal (Pascal 的例子)

假設 $N = 4$ ， $a_1 = 2, a_2 = 3, a_3 = 6, a_4 = 8$ 。

其正確答案是 3，這個由 `findGap` 計算並返回的答案是可以經由下面調用 `MinMax` 的步驟而求得的：

- 調用 `MinMax(1, 2, mn, mx)` 得到 `mn` 及 `mx` 均為 2
- 調用 `MinMax(3, 7, mn, mx)` 得到 `mn` 為 3 而 `mx` 為 6
- 調用 `MinMax(8, 9, mn, mx)` 得到 `mn` 及 `mx` 均為 8.

Scoring (計分方法)

對於所有測試數據而言，它們都會合符 $2 \leq N \leq 100,000$ 這個條件。

子任務 1 (30 分): 每次調用 `MinMax` 時系統都會加 1 到 M 。若你能在這個子任務內做到使所有測試數據都滿足 $M \leq \frac{N+1}{2}$ 的話，則你可得到滿分。

子任務 2 (70 分): 設在調用函數 `MinMax` 時，共有 k 個輸入整數是大於或等於 s 同時亦小於或等於 t 的。每次這樣的函數調用評分系統都會加 $k + 1$ 到 M 中。而程式最終的得分是根據以下的計算法則來決定：這子任務的最終得分是取你的程式在所有測試數據中最低分的一個；而對每個測試數據而言，若 $M \leq 3N$ 則你可得到 70 分，否則你的得分將會是 $\frac{60}{\sqrt{\frac{M}{N} + 1} - 1}$ 。

Experimentation (評測實驗)

你可以從評分系統中下載樣例評分程式 (grader)。這個樣例評分程式會從標準輸入中讀取數據。這些數據的第一行含有兩個整數，它們分別是子任務的編號 T 及整數 N 。隨後一行應有 N 個以升序排列着的整數。樣例評分程式將會在標準輸出中輸出以下資料：第一行是你的 `findGap` 函數的返回值，第二行則是對應於子任務編號的 M 的值。

以上的輸入資料是對應於上面的例子：

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