



## Gap

$N$  adet negatif olmayan tam sayıdan oluşan  $a_1, a_2, \dots, a_N$  dizisi verilmiştir. Ayrıca bu dizi  $0 \leq a_1 < a_2 < \dots < a_N \leq 10^{18}$  eşitsizliğini sağlamaktadır. Jeihak,  $1, N - 1$  aralığındaki  $i$  değerleri için *en büyük*  $a_{i+1} - a_i$  değerine sahip  $i$  değerini merak etmektedir. Jeihak'ın kodu sayıları direk girdi olarak almayacaktır özel bir fonksiyon ile alacaktır. Kodlama detaylarına aşağından ulaşabilirsiniz.

## Görev

Sizden Jeihak'e bir fonksiyon kodlamanız istenmektedir bu fonksiyon  $1, N - 1$  aralığındaki  $i$  değerleri için *en büyük*  $a_{i+1} - a_i$  değerine sahip  $i$  değerini bulacaktır.

## C ve C++ kodlama detayları

`findGap(T, N)` isimli bir fonksiyon kodlamanız gerekmektedir ve parametre olarak `long long` tipine sahip sırasıyla şu değişkenleri alacaktır:

- `T` — alt görev numarası (1 veya 2)
- `N` — dizideki sayıların sayısı

`findGap` fonksiyonunuz `MinMax(s, t, &mn, &mx)` fonksiyonunu çağıracaktır. Bu fonksiyonun ilk iki parametresi olan `s` ve `t` iki tam sayıdır ve `long long` tipindedir, son iki parametresi olan `&mn` ve `&mx` değişkenleri ise `long long` tipindeki tamsayıları gösteren pointer'lardır. Öte yandan `mn` ve `mx` değişkenleri ise `long long` tipinde tam sayılardır. `MinMax(s, t, &mn, &mx)` fonksiyonu çağrılıp sonlandığında `mn` değeri `s` değerinden büyük en küçük  $a_i$  tamsayısına eşit olacaktır aynı şekilde `mx` değeri ise `t` sayısından küçük en büyük  $a_j$  değerine eşit olacaktır. `s` ve `t` (her ikisi de dahil) değerleri arasında herhangi bir sayı yoksa `mn` ve `mx` değerleri `-1`'e eşit olacaktır. `s` değeri her zaman `t` değerinden büyük olmamalıdır. Bu şart sağlanmadığı takdirde kod sıfır puan olarak sonlandırılacaktır.

## Pascal kodlam detayları

Not: Bu bölümün kullanılmayacağını düşündüğümüz için çevirmedi.

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.

## Genel Kodlama Detayları

Standart kısıtlamaların (zaman ve hafıza sınırı) yanında, gönderiminizin bir girdiden puan alabilmesi için şu kısıtları sağlaması gerekmektedir.

- fonksiyonunuz `findGap` doğru cevabı döndürmelidir,
- maliyeti belirten  $M$  tamsayısı `MinMax` fonksiyonunun çağrılmasıyla artmaktadır, bu değer aşağıda belirtilen kısıtlamaları sağlamalıdır (Puanlandırma bölümünü inceleyiniz).

## C, C++ için örnek

$N = 4$  ve  $a_1 = 2, a_2 = 3, a_3 = 6$ , ve  $a_4 = 8$  durumunu göz önüne alalım.

Cevap, **3** olacaktır, `findGap` fonksiyonunda aşağıdaki `MinMax` sorgularını yaparak doğru bir şekilde hesaplanıp döndürülebilir:

- `MinMax(1, 2, &mn, &mx)` fonksiyonu çağrılır,  $mn$  ve  $mx$  değişkenlerinin her ikisi de **2** değerine sahip olur.
- `MinMax(3, 7, &mn, &mx)` fonksiyonu çağrılır,  $mn$  değişkeni **3** değerine ve  $mx$  değişkeni **6** değerine sahip olur.
- `MinMax(8, 9, &mn, &mx)` fonksiyonu çağrılır,  $mn$  ve  $mx$  değişkenlerinin her ikisi de **8** değerine sahip olur.

## Pascal için örnek

Not: Bu bölümün kullanılmayacağını düşündüğümüz için çevirmedi.

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**.

## Puanlandırma

Tüm altgörevlerde  $2 \leq N \leq 100,000$  kısıtlaması sağlanacaktır.

**Altgörev 1 (30 puan):** Her `MinMax` sorgusu  $M$  değerini **1** artıracaktır. Eğer  $M \leq \frac{N+1}{2}$  şartı bütün girdiler için sağlanırsa kodunuz bu altgörevden tam puan alacaktır.

**Altgörev 2 (70 puan):** Varsayalım  $k$  `MinMax` sorgusundaki  $s$  değerinden büyük eşit ve  $t$  değerinden küçük eşit olan sayıların sayısı olsun. Her `MinMax` sorgusu  $M$  değerini  $k + 1$  artıracaktır. Son puan şu

şekilde hesaplanacaktır: Altgörevin son puanı bütün girdilerdeki en küçük puana eşit olacaktır. Bir test girdisi için, eğer  $M \leq 3N$  şartı sağlanırsa puan 70 olacaktır diğer durumlarda puan  $\frac{60}{\sqrt{\frac{M}{N}+1}-1}$

formülüyle hesaplanacaktır.

## Test

Örnek değerlendirme sistemi sınav sisteminden indirilip denenebilir. Bu sistem girdinin ilk satırında sırasıyla altgörev numarasını ve girdideki sayıların sayısını belirten  $T$  ve  $N$  tamsayılarını alacaktır. Girdinin ikinci satırında ise artan sırada  $N$  adet tam sayı yer alacaktır. Çıktıda ise her bir test girdisi için findGap döndürdüğü sonuç ve yukarıda belirtilen şekilde hesaplanan maliyeti belirten  $M$  tamsayıları yer alacaktır.

Aşağıdaki girdi yukarıdaki örnekleri temsil etmektedir:

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