

## A 研究所での、実話。

### 問題文

長さ  $N$  の非負整数列が与えられ、その  $i$  番目の要素  $V_i$  は、ID が  $i$  である神経細胞の電位(mV)を示しています。電位が しきい値  $T$  以上である神経細胞の ID を出力してください。

### 制約

- ・  $1 \leq N \leq 1000$
- ・  $0 \leq V_i \leq 100$
- ・  $0 \leq T \leq 100$
- ・ 入力はすべて整数

### 入力

$N$   $T$   
 $V_0$   $V_1$   $V_2$   $\cdots$   $V_{N-2}$   $V_{N-1}$

### 出力

$T \leq V_i$  となるような  $i$  を半角スペース区切りで出力せよ。

#### 入力例 1

```
5 8
7 9 3 0 8
```

#### 出力例 1

```
1 4
```

電位が、しきい値(8mV)以上である神経細胞の ID は、1 番と 4 番です。

#### 入力例 2

```
9 30
56 72 19 67 32 33 0 77 19
```

#### 出力例 2

```
0 1 3 4 5 7
```

### 【半角スペース区切りの複数の入力値を受け取る方法】

例) 8 3 という 二つの入力値を受け取る場合

```
a, b = map(int, input().split())    # a に 8, b に 3 が代入される
```

例) 長さ  $N$  ( $N = 8$ ) の入力値 (7 9 5 4 8 7 6 5) をリストにして受け取る場合

```
A = list(map(int, input().split())) # A に [7, 9, 5, 4, 8, 7, 6, 5] が代入される
```

## B アルゴリズムの違いによる、計算量の違い

### 問題文

長さ  $N$  の非負整数列が与えられ、その  $i$  番目の要素  $V_i$  は、ID が  $i$  である神経細胞の電位(mV)を示しています。  
「ID が  $L$  の細胞から、ID が  $R$  の細胞までの電位の総和 ( $V_L + V_{L+1} + V_{L+2} + \dots + V_{R-1} + V_R$ ) を出力する」  
というクエリが  $Q$  個与えられるので、各クエリに答えてください。

### 制約

- $1 \leq N \leq 1000$
- $0 \leq V_i \leq 100$
- $0 \leq L \leq R \leq N - 1$
- $1 \leq Q \leq 10^5$
- 入力はすべて整数

### 入力

$N$   $Q$   
 $V_0$   $V_1$   $V_2$   $\dots$   $V_{N-2}$   $V_{N-1}$   
 $L_0$   $R_0$   
 $L_1$   $R_1$   
 $L_2$   $R_2$   
:  
 $L_Q$   $R_Q$

### 出力

各クエリに対する回答 (ID が  $L$  の細胞から、ID が  $R$  の細胞までの電位の総和) を  $Q$  行に渡って出力せよ。

### 入力例

8 5  
7 9 3 0 8 5 1 9  
2 6  
1 4  
0 7  
6 6  
4 5

### 出力例

17  
20  
42  
1  
13

二重ループを復習していたそのまさんは、この問題も難なく解くことが出来ました。  
しかし、そのまさんは自分の解法では  
**クエリの数 (Q) が増えるほど計算時間が膨大に増えていく** という欠点を見つけました。  
実際、 $Q$  が  $10^5$  のときでは **1400ms** 程度の計算時間を要しました。

しかし、隣のマウナくんを見してみると、  
 $Q$  が  $10^5$ なのに、計算時間がたったの **12ms** でした。

マウナくんのコードを見せてもらうと、別の解法をとっていることが分かりました。  
さて、マウナくんはどのように解いたのでしょうか？  
効率の良いアルゴリズムを設計し、多量のクエリにも耐えうるコードを作成してください。