

隱藏的序列 (sequence)

本題為**互動題**，限用 C++ 上傳。

問題描述

Alice 和 Bob 正在遊玩猜謎遊戲，由 Alice 負責出題目、Bob 負責猜謎。

遊戲過程如下：

- Alice 會先在心中想好一個序列。也就是說，Alice 心中已經有一個隱藏的序列，這個序列不會根據 Bob 的詢問而改變。
- 接著，Bob 可以詢問 t 個問題，每個問題都會是以「請問 p_i 和 p_j 和 p_k 這三數的中位數是誰？」這種形式呈現。Alice 在收到問題後，必須如實回答。
- 在 Bob 問完所有問題後，Alice 會問 Bob q 個問題，每個問題都會是以「請問 p_i 是多少？」這種形式呈現。Bob 在收到問題後，必須給出回答。

這個遊戲的目的就是要讓 Bob 詢問的問題數量 t 儘量小來使得 Bob 能正確回答出 Alice 的所有詢問。請協助 Bob，在 t 儘量小的情況下，正確回答所有 Alice 的 q 個問題。

實作細節

你需要實作兩個函式 `bob_init()` 與 `query_from_alice()`：

```
void bob_init(int n);
```

- 對於每一筆測試資料，正式評分程式會呼叫你實作的 `bob_init()` 函式恰好 1 次。
- n 代表 Alice 心中想著的排列長度

```
int query_from_alice(int a);
```

- a 為 0 到 $n - 1$ 之間的整數
- 對於每一筆測試資料，正式評分程式會呼叫你實作的 `query_from_alice()` 函式恰好 q 次。
- 保證在呼叫完 `bob_init()` 後才會呼叫此函式。
- `query_from_alice()` 需要回傳一個整數 x ，代表 p_a 的實際數值。
- 因為你不可能知道最小值及最大值是多少，所以若 p_a 為序列中的最大或最小值不管你回答什麼 Alice 都會算你答對。

此外，在實作 `bob_init` 時可以呼叫 `compare_numbers()` 這個函式。

```
int compare_numbers(int i, int j, int k);
```

- i, j, k 是於 $0 \sim n - 1$ 的整數且三數需兩兩相異。
- 此函式會回傳 p_i, p_j, p_k 三數的中位數，也就是這三個數中第二大的數。
- 範例評分程式內的 `compare_numbers()` 實作與實際評分程式內的實作完全相同。

測資限制

$$3 \leq n \leq 1000$$

$$-10^9 \leq p_i \leq 10^9$$

$$p_i \neq p_j (\forall i \neq j)$$

$$q \leq 1000$$

judge 不是 adaptive 的

評分說明

對於每一筆測試資料，若你的程式在函式 `bob_init()` 中呼叫 `compare_numbers` 的次數為 t ，則定義 Q 為：

$$Q = \lceil \frac{t}{n} \rceil$$

若你正確回答了所有 Alice 的詢問，根據 Q ，你將得到分數比重 W ：

$$W = \begin{cases} 1 & \text{if } Q \leq 3 \\ \frac{109-3Q}{100} & \text{if } 3 \leq Q \leq 9 \\ \frac{2.46}{\sqrt{Q}} & \text{if } 9 \leq Q \leq 500 \\ 0 & \text{if } Q > 500 \end{cases}$$

本題共有兩組子任務，條件限制如下所示。每一組可有一或多筆測試資料，你在該子任務的得分為所有測試資料中分數比重 W 的最小值，乘以該子任務的總分。

子任務

1. (10 分) $n = 3$
2. (90 分) 無額外限制

範例評分程式

範例評分程式採用以下格式輸入

```
 $n$   $q$   
 $p_1$   $p_2$   $\dots$   $p_n$   
 $ask_1$   
 $ask_2$   
 $\dots$   
 $ask_q$ 
```

代表 Alice 心中所想的序列長度為 n ，有 q 個詢問。

p 為 Alice 心中所想的序列，詢問時 Alice 想問 p_{ask_i} 是多少。