



大獎 (The Big Prize)

Big Prize是一個有名的電視遊戲節目。你是晉級到最後回合的幸運選手，你站在一排 n 個箱子前方，這些箱子從左到右由0編號到 $n - 1$ 。每個箱子內有一個獎品，除非箱子被打開，否則無法看見箱子內的獎品。總共有 $v \geq 2$ 種不同的獎品種類(type)，種類的編號是由1到 v 依價值遞減順序編號。

Type 1是價值最貴的獎品：鑽石，箱子裡總共恰好有一個鑽石。Type v 是最便宜的獎品：一根棒棒糖。為了讓遊戲更為刺激，便宜的獎品數量遠遠超過較貴獎品的數量。更精確地講，對於所有 $2 \leq t \leq v$ ，我們知道以下事實：如果type $t - 1$ 的獎品有 k 個，那麼type t 的獎品數量絕對超過 k^2 。

你的目標是要贏得鑽石，在遊戲的最後，你必須開啟一個箱子而得到那個箱子裡的獎品。在必須選擇要開啟的箱子之前，你可以問遊戲節目主持人Rambod一些問題。每一次問問題時，你指定某個箱子 i ，Rambod會回答你包含兩個整數的陣列 a ，其意義如下：

- 在箱子 i 左邊恰好有 $a[0]$ 個獎品比箱子 i 中的獎品更貴。
- 在箱子 i 右邊恰好有 $a[1]$ 個獎品比箱子 i 中的獎品更貴。

例如，假設 $n = 8$ 。你選擇 $i = 2$ 作為你的問題，Rambod將告訴你 $a = [1, 2]$ ，其意義為：

- 在箱子0和1中恰好一個箱子的獎品比箱子2中的更貴。
- 在箱子3, 4, ..., 7中恰好有2個箱子的獎品比箱子2中的更貴。你的任務是使用少量的問題來找出裝著鑽石的箱子

實作細節 (Implementation details)

你需要實作以下函式：

```
int find_best(int n)
```

- 這個函式會被評分程式恰好呼叫一次。
- n : 箱子數量。
- 此函式必須回傳裝著鑽石的箱子編號，該編號就是唯一的整數 d ($0 \leq d \leq n - 1$)，而箱子 d 裝的獎品是type 1。

以上函式可以呼叫下列函式：

```
int[] ask(int i)
```

- i : 你的問題所指定的箱子編號， i 的值必須介於0與 $n - 1$ 之間(包含0與 $n - 1$)。
- 這個函式回傳兩個元素的陣列 a 。此處， $a[0]$ 是在箱子 i 左邊而價值比箱子 i 中獎品更貴的獎品數

量，而 $a[1]$ 是在箱子 i 右邊而價值比箱子 i 中獎品更貴的獎品數量。

範例

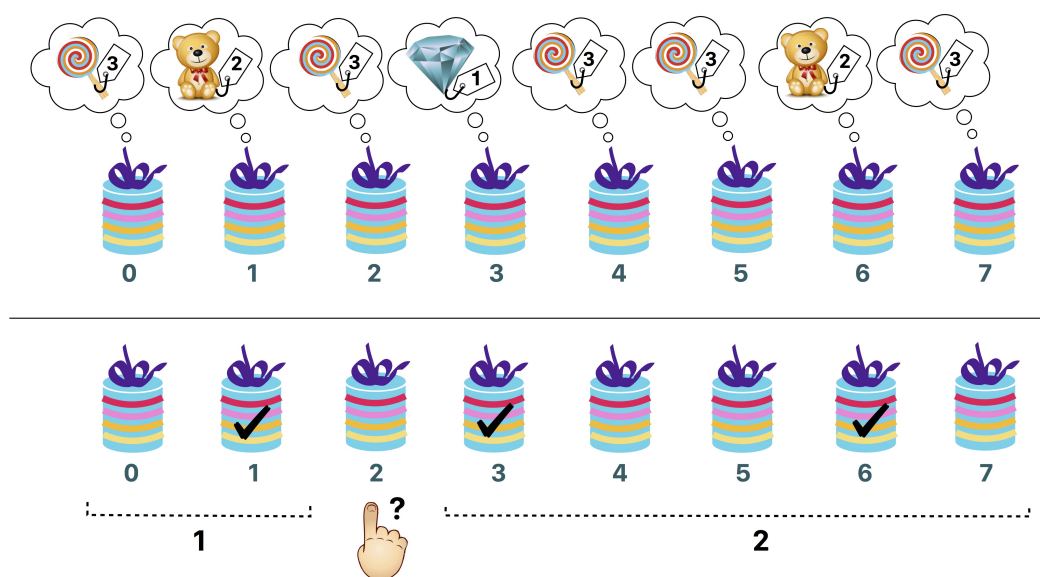
評分程式執行下列呼叫：

```
find_best(8)
```

一共有 $n = 8$ 箱子，假設獎品種類是 $[3, 2, 3, 1, 3, 3, 2, 3]$ 。呼叫函式 `ask` 的所有可能方式與其回傳值列示如下。

- `ask(0)` returns $[0, 3]$
- `ask(1)` returns $[0, 1]$
- `ask(2)` returns $[1, 2]$
- `ask(3)` returns $[0, 0]$
- `ask(4)` returns $[2, 1]$
- `ask(5)` returns $[2, 1]$
- `ask(6)` returns $[1, 0]$
- `ask(7)` returns $[3, 0]$

在此範例中，鑽石在箱子3中，所以函式`find_best` 應該回傳3。



以上的圖顯示了這個範例。圖的上半部表示各個箱子內獎品的type。圖的下半部表示當詢問`ask(2)` 的情形，打勾箱子中的獎品是比箱子2中的獎品更貴。

限制 (Constraints)

- $3 \leq n \leq 200\,000$.
- 每個箱子內獎品的種類(type)是介於1與 v 之間，包含1與 v 。

- 恰好一個獎品是type 1。
- 對於所有的 $2 \leq t \leq v$ ，如果有 k 個獎品是type $t - 1$ ，那麼絕對超過(*strictly more than*) k^2 個獎品是type t 。

子任務與評分(Subtasks and scoring)

在某些測試資料中，評分程式的行為是會調整的，這個意思是，對於這些測資，評分程式的獎品序列並非是固定的，相反的，評分程式的答案可能根據你的解答程式所問的詢問而改變。評分程式的回覆保證滿足以下條件：在每一次回覆詢問後，必定存在至少一個獎品序列符合到目前為止所做的所有回覆。

1. (20 points) 恰好有1個鑽石以及 $n - 1$ 個棒棒糖（也就是 $v = 2$ ）。你可以呼叫 `ask` 函式最多 10 000 次。
2. (80 points) 無額外限制。

在子任務2中你可能得到部分分數。令 q 是在所有測資中你呼叫`ask`函式的最大次數。那麼，你在子任務2的分數依照下列表格計算：

Questions	Score
$10\,000 < q$	0 (CMS 回報 'Wrong Answer')
$6000 < q \leq 10\,000$	70
$5000 < q \leq 6000$	$80 - (q - 5000)/100$
$q \leq 5000$	80

樣板評分程式(Sample grader)

樣板評分程式不是可調整的，相反的，它只是讀取一個固定的陣列 p 當作獎品類別。對於所有 $0 \leq b \leq n - 1$ ，在箱子 b 中的獎品種類就是 $p[b]$ 。樣板評分程式讀取以下格式的輸入：

- line 1: n
- line 2: $p[0] \ p[1] \ \dots \ p[n - 1]$

樣板評分程式以單獨一行列印`find_best`的回傳值以及呼叫`ask`函式的次數。