



第五題：不時輕聲地以 01 遮羞的鄰座吳姪同學 (Tyellow) [此題為互動題 Interactive]

問題敘述

吳姪同學，是板條帝國高中裡面的女裝科競賽選手。吳姪同學非常喜歡坐在他隔壁的尖叫雞同學，所以當尖叫雞同學來找他說話時，他常常會因為太害羞，而只用 0 跟 1 來回答。

今年校內梳理既女裝學科能力競賽結束了，吳姪同學和尖叫雞同學都想當然地成為了校隊選手，於是他們便約定好要一起團練女裝。而吳姪同學認為只是一起團練好像有點太無聊了，於是他想要考考尖叫雞同學，讓尖叫雞同學猜猜他心中所想的「女裝密碼」。

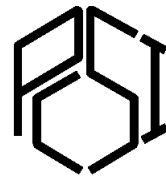
首先吳姪同學先想了一個長度為 N 的 1 到 N 的排列 A ，並且告訴了尖叫雞同學一個數字 K ，以及他的「女裝密碼」可以用 $\sum_{A_i \geq K} A_i \times i$ 來表示。吳姪同學又告訴了尖叫雞同學，他有一個長度為 1000000 的 B 序列可以用，一開始，這個序列的每一項都是 0。

而尖叫雞同學雖然不能直接看到 A 和 B 序列裡面的內容，但是他可以用以下幾種操作來對這兩個序列做一些事：

1. 選擇 x, y ，並且讓 $B_x + = A_y$ 。
2. 選擇 x, y ，並且讓 $B_x - = A_y$ 。
3. 選擇 x, y ，並且讓 $B_x + = B_y$ 。
4. 選擇 x, y ，並且讓 $B_x - = B_y$ 。
5. 選擇 x ，並且讓 $B_x = |B_x|$ 。
6. 選擇 x, c ，並且讓 $B_x = c$ 。

雖然尖叫雞同學看不到 A 和 B 序列裡面的內容，但吳姪同學是看得到的，於是尖叫雞同學還可以選擇一個 x ，並問吳姪同學 B_x 的數字是多少？但由於吳姪同學太害羞了，當 $B_x = 0$ 時，他會回答尖叫雞同學 0，否則他會回答 1。

你能夠順利地找出吳姪同學的女裝密碼（ $\sum_{A_i \geq K} A_i \times i$ ）嗎？



實作細節

C++

你需要在首行加入 `#include "Tyellow.h"`，並完成以下函式：

```
long long game_start(int N,int K);
```

- 評分程式一開始會呼叫 `game_start` 並傳入 N 和 K 的值。
- 傳入的 N 值滿足 $1 \leq N \leq 1000$ 。
- 傳入的 K 值滿足 $1 \leq K \leq N$ 。
- 請在此函數內求出並回傳吳姁同學的女裝密碼。

你的程式可以呼叫以下函式：

```
void B_add_A(int x,int y);
```

- 代表尖叫雞同學選擇讓 $B_x += A_y$ 。
- 傳入之參數 x 必須為 $[1, 1000000]$ 之間的整數。
- 傳入之參數 y 必須為 $[1, N]$ 之間的整數。

```
void B_sub_A(int x,int y);
```

- 代表尖叫雞同學選擇讓 $B_x -= A_y$ 。
- 傳入之參數 x 必須為 $[1, 1000000]$ 之間的整數。
- 傳入之參數 y 必須為 $[1, N]$ 之間的整數。

```
void B_add_B(int x,int y);
```

- 代表尖叫雞同學選擇讓 $B_x += B_y$ 。
- 傳入之參數 x 必須為 $[1, 1000000]$ 之間的整數。
- 傳入之參數 y 必須為 $[1, 1000000]$ 之間的整數。

```
void B_sub_B(int x,int y);
```

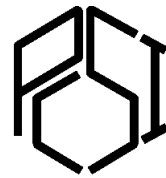
- 代表尖叫雞同學選擇讓 $B_x -= B_y$ 。
- 傳入之參數 x 必須為 $[1, 1000000]$ 之間的整數。
- 傳入之參數 y 必須為 $[1, 1000000]$ 之間的整數。

```
void abs_B(int x);
```

- 代表尖叫雞同學選擇讓 $B_x = |B_x|$ 。
- 傳入之參數 x 必須為 $[1, 1000000]$ 之間的整數。

```
void set_B(int x,int c);
```

- 代表尖叫雞同學選擇讓 $B_x = c$ 。
- 傳入之參數 x 必須為 $[1, 1000000]$ 之間的整數。
- 傳入之參數 c 必須為 $[-1000000000, 1000000000]$ 之間的整數。



```
bool ask_Tyellow(int x);
```

- 代表尖叫雞同學問吳姪同學 B_x 的值，ask_Tyellow 會回傳一個布林值，如果 B_x 是 0，則會回傳 *False*，否則會回傳 *True*。
- 傳入之參數 x 必須為 $[1, 1000000]$ 之間的整數。

請注意，在上面這些操作中，序列 B 中每一項的最大值只能存到 10^{17} ，最小值只能存到 -10^{17} ，若計算中超過了最大值，或小於了最小值，則會自動被變成 10^{17} 和 -10^{17} 。

你的程式最多可以呼叫 ask_Tyellow 2000000 次以及 set_B 2000000 次。

如果不符合上述條件限制，你的程式會被判為 **Wrong Answer**；否則你的程式會依據你呼叫函數的次數以及是否達成題目要求來評斷給分。



Python3

你需要完成以下函式：

```
def game_start(N: int, K: int) -> int
```

- 評分程式一開始會呼叫 `game_start` 並傳入 N 和 K 的值。
- 傳入的 N 值滿足 $1 \leq N \leq 1000$ 。
- 傳入的 K 值滿足 $1 \leq K \leq N$ 。
- 請在此函數內求出並回傳吳姪同學的女裝密碼。

你的程式可以呼叫以下函式：

```
def B_add_A(x: int, y: int)
```

- 代表尖叫雞同學選擇讓 $B_x + = A_y$ 。
- 傳入之參數 x 必須為 $[1, 1000000]$ 之間的整數。
- 傳入之參數 y 必須為 $[1, N]$ 之間的整數。

```
def B_sub_A(x: int, y: int)
```

- 代表尖叫雞同學選擇讓 $B_x - = A_y$ 。
- 傳入之參數 x 必須為 $[1, 1000000]$ 之間的整數。
- 傳入之參數 y 必須為 $[1, N]$ 之間的整數。

```
def B_add_B(x: int, y: int)
```

- 代表尖叫雞同學選擇讓 $B_x + = B_y$ 。
- 傳入之參數 x 必須為 $[1, 1000000]$ 之間的整數。
- 傳入之參數 y 必須為 $[1, 1000000]$ 之間的整數。

```
def B_sub_B(x: int, y: int)
```

- 代表尖叫雞同學選擇讓 $B_x - = B_y$ 。
- 傳入之參數 x 必須為 $[1, 1000000]$ 之間的整數。
- 傳入之參數 y 必須為 $[1, 1000000]$ 之間的整數。

```
def abs_B(x: int)
```

- 代表尖叫雞同學選擇讓 $B_x = |B_x|$ 。
- 傳入之參數 x 必須為 $[1, 1000000]$ 之間的整數。

```
def set_B(x: int, c: int)
```

- 代表尖叫雞同學選擇讓 $B_x = c$ 。
- 傳入之參數 x 必須為 $[1, 1000000]$ 之間的整數。
- 傳入之參數 c 必須為 $[-1000000000, 1000000000]$ 之間的整數。



```
def ask_Tyellow(x: int) -> bool
```

- 代表尖叫雞同學問吳姪同學 B_x 的值，ask_Tyellow 會回傳一個布林值，如果 B_x 是 0，則會回傳 *False*，否則會回傳 *True*。
- 傳入之參數 x 必須為 $[1, 1000000]$ 之間的整數。

請注意，在上面這些操作中，序列 B 中每一項的最大值只能存到 10^{17} ，最小值只能存到 -10^{17} ，若計算中超過了最大值，或小於了最小值，則會自動被變成 10^{17} 和 -10^{17} 。

請在 game_start 函式內的第一行加入 `from __main__ import B_add_A, B_sub_A, B_add_B, B_sub_B, abs_B, ask_Tyellow, set_B` 來導入函式，才能做使用。如：

```
def game_start(N: int, K: int) -> int:  
    from __main__ import B_add_A, B_sub_A, B_add_B, B_sub_B, abs_B, ask_Tyellow, set_B
```

你的程式最多可以呼叫 ask_Tyellow 2000000 次以及 set_B 2000000 次。

如果不符合上述條件限制，你的程式會被判為 **Wrong Answer**；否則你的程式會依據你呼叫函數的次數以及是否達成題目要求來評斷給分。



互動範例

考慮以下的測試資料：

$$N = 5$$

$$K = 4$$

$$A = [1, 2, 3, 4, 5]$$

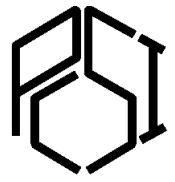
一個不會被評分程式判斷為 **Wrong Answer** 的互動例子顯示如下：

評分程式端	參賽者端
呼叫 <code>game_start(5,4)</code>	呼叫 <code>B_add_A(1,3)</code> 呼叫 <code>B_add_A(2,5)</code> 呼叫 <code>B_add_B(3,1)</code>
回傳 1	呼叫 <code>ask_Tyellow(3)</code> 呼叫 <code>set_B(3,0)</code>
回傳 0	呼叫 <code>ask_Tyellow(3)</code> 呼叫 <code>set_B(1,9)</code> 呼叫 <code>B_sub_A(1,4)</code> 呼叫 <code>B_sub_A(1,5)</code>
回傳 0	呼叫 <code>ask_Tyellow(1)</code> 回傳 41

評分說明

本題共有 10 組測試題組，條件限制如下所示。每一組可有一或多筆測試資料，該組所有測試資料皆需答對才會獲得該組分數。

子任務	分數	額外輸入限制
1	0	互動範例 ($N = 5$ ， A 序列如同上面的範例)。
2	1	$N = 1$ 。
3	2	$N = 2$ 。
4	3	$N = 3$ 。
5	15	$K = 1$ 。
6	5	$N \leq 10$ 。
7	7	$N \leq 50$ 。
8	8	$N \leq 100$ 。
9	11	$N \leq 500$ 。
10	48	無額外限制。



- 對於子任務 2、3、4、6：

若你達成了題目要求的條件，不論函式呼叫的次數，你會獲得該子任務的分數。

- 對於子任務 5：

若你在該子任務中的第 i 筆測資中呼叫了 `set_B` 函式 X_i 次；呼叫了 `ask_Tyellow` 函式 Y_i 次，令 $P_i = 2^{-X_i}$ ，你在該子任務的得分會用以下方式計算：

– $15 \times$ 該組所有測試資料中你獲得最小的 P_i 。

- 對於子任務 7：

若你在該子任務中的第 i 筆測資中呼叫了 `set_B` 函式 X_i 次；呼叫了 `ask_Tyellow` 函式 Y_i 次，令 $P_i = \frac{1}{10} + (\frac{9}{10})^{\max(1, \frac{Y_i}{440})} \times 1.3^{-X_i}$ ，你在該子任務的得分會用以下方式計算：

– $7 \times$ 該組所有測試資料中你獲得最小的 P_i 。

- 對於子任務 8：

若你在該子任務中的第 i 筆測資中呼叫了 `set_B` 函式 X_i 次；呼叫了 `ask_Tyellow` 函式 Y_i 次，令 $P_i = \frac{1}{10} + (\frac{9}{10})^{\max(1, \frac{Y_i}{570})} \times 1.5^{-X_i}$ ，你在該子任務的得分會用以下方式計算：

– $8 \times$ 該組所有測試資料中你獲得最小的 P_i 。

- 對於子任務 9：

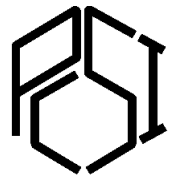
若你在該子任務中的第 i 筆測資中呼叫了 `set_B` 函式 X_i 次；呼叫了 `ask_Tyellow` 函式 Y_i 次，令 $P_i = \frac{2}{10} + (\frac{8}{10})^{\max(1, \frac{Y_i}{5050})} \times 1.7^{-X_i}$ ，你在該子任務的得分會用以下方式計算：

– $11 \times$ 該組所有測試資料中你獲得最小的 P_i 。

- 對於子任務 10：

若你在該子任務中的第 i 筆測資中呼叫了 `set_B` 函式 X_i 次；呼叫了 `ask_Tyellow` 函式 Y_i 次，則你在該子任務的得分會依照以下方式計算，並取該組所有測試資料中計算出來最小者作為最後的得分：

- 若 $Y_i > 2000000$ ，你將不會獲得任何分數。
- 若 $32222 < Y_i \leq 2000000$ ，你將會獲得 $(4 + \frac{2000000 - Y_i}{218642}) \times 2^{-X_i}$ 分。
- 若 $2040 < Y_i \leq 32222$ ，你將會獲得 $(14 + \frac{32222 - Y_i}{3020}) \times 2^{-X_i}$ 分。
- 若 $1030 < Y_i \leq 2040$ ，你將會獲得 $(24 + \frac{2040 - Y_i}{170}) \times 2^{-X_i}$ 分。
- 若 $30 < Y_i \leq 1030$ ，你將會獲得 $(34 + \frac{1030 - Y_i}{100}) \times 2^{-X_i}$ 分。
- 若 $Y_i \leq 30$ ，你將會獲得 48×2^{-X_i} 分。



範例評分程式

範例評分程式以下列格式讀取輸入：

- 第一列： N
- 第二列： A_1, A_2, \dots, A_N

其中 N 如題目所述， A_i 為選手可自行設計之吳姪同學所想的排列 A ，注意範例評分程式並不會幫助你檢查你所輸入的東西是否符合題目限制。

請注意：使用自己上傳的測試資料進行測試時，沒有下面 MSG 描述的情形時你會得到 **Accepted**。如果你的程式被評為 **Accepted**，範例評分程式會輸出 **Accepted: N q1 q2**，其中 $q1$ 表示呼叫 `ask_Tyellow` 的次數； $q2$ 表示呼叫 `set_B` 的次數。如果你的程式被評為 **Wrong Answer**，範例評分程式會輸出 **Wrong Answer: MSG**，其中 **MSG** 格式與意義如下：

- **index out of range**：操作或詢問時所帶入變數之值超出規定範圍。
- **constant out of range**：呼叫 `set_B` 函式時帶入的 c 值超出規定範圍。

在 CMS 內的附件檔案中，有一個名為「Tyellow.zip」的壓縮檔，下載後解壓縮可以找到三個資料夾「cpp」、「py」和「examples」，資料夾的意義分別為：

- **cpp**：內部包含一個檔案「Tyellow.cpp」，你可以直接上傳該檔案來獲得 1 分，並且往後你要寫的程式都可以參考這份檔案，並在這份檔案內進行修改、編譯及執行。

請注意，檔案內有兩行註解分別為「do not modify above」和「do not modify below」，這兩行意味著希望你盡量只更動被這兩行夾住的區域，若你對互動題並沒有很熟悉，請不要更動外面的區域，更動外面的區域不會也不可能讓你能直接拿到更高的分數。

基本上你不需要更動除此之外的檔案，但若有需求可以在理解運作原理後進行更動。

- **py**：內部包含一個檔案「Tyellow.py」，你可以直接上傳該檔案來獲得 1 分，並且往後你要寫的程式都可以參考這份檔案，並在這份檔案內進行修改。

若要執行程式，請切換到另一個檔案「grader.py」，直接執行後「grader.py」會嘗試與你修改過的「Tyellow.py」做連結後開始運行。**請盡量只更動「Tyellow.py」**，若你對互動題並沒有很熟悉，請不要更動「grader.py」，更動「grader.py」不會也不可能讓你能直接拿到更高的分數。

基本上你不需要更動除此之外的檔案，但若有需求可以在理解運作原理後進行更動。

- **examples**：內部包含互動範例的輸入和合法輸出。

請不要嘗試撰寫題目指定需要函式以外的任何東西，例如自行輸入、輸出等，無論是「grader.cpp」還是「grader.py」都僅供參考用，並與 Judge 上的有所落差。若有疑似偷取資料的行為，將視為作弊，嚴重者可能會取消資格並以校規處分。