

# 分數計算機

## 問題敘述

Colten 與 Eric 製作了一款好玩的程式設計遊戲，而你現在是這個遊戲的主角，必須想辦法通關！

這個遊戲非常簡單，只要寫程式就好，相信對你來說一定游刃有餘的吧，讓我們先來設計一個簡單的程式看看吧！

這個程式必須能夠讀入四個正整數，輸入共兩行，每行有兩個正整數，每一行的那兩個正整數依序代表該行分數的分子與分母。

你的程式必須要能夠計算出這兩個分數相加後如果不換成帶分數的話會是真分數還是假分數，如果是真分數要輸出 1，反之則輸出 0，開始動手吧！

真分數：分子 < 分母

假分數：分子  $\geq$  分母

## 輸入說明

輸入只有兩行，每行輸入兩個正整數，保證數字範圍不超過  $10^5$ 。

## 輸出說明

輸出 1 或 0 表示判定的結果。

## 範例測資

### 範例輸入 1

1 2  
3 4

### 範例輸出 1

0

### 範例輸入 2

1 3  
1 2

### 範例輸出 2

1

# 簡單明瞭的題目

## 問題敘述

題目任務很簡單，輸入一個整數  $N$ ，請你設計一個程式回報  $N$  是不是奇數。

## 輸入說明

輸入只有一行，輸入一個整數  $N$  ( $|N| \leq 1000$ )。

## 輸出說明

如果  $N$  是奇數，請輸出 **Yes**，反之，則輸出 **No** (任意字母大小寫組合皆可)。

## 範例測資

### 範例輸入 1

2

### 範例輸出 1

No

### 範例輸入 2

3

### 範例輸出 2

Yes

### 範例輸入 3

0

### 範例輸出 3

No

# 倍數判斷與數字相接

## 問題敘述

C 語言不像 Python 一樣，數字儲存的大小可以很大，因此當我們面臨到比較大數字的問題時往往必須透過一些特殊的方法來處理。

為了測試你有沒有具備這個能力，Colten 與 Eric 想要考考你怎麼解決下面這個問題：

給你兩個超大的正整數，請你輸出這兩個整數接在一起的結果與合併後的結果是否為 11 的倍數。

## 輸入說明

第一行輸入兩個正整數，數字不會超過  $10^{100}$ 。

## 輸出說明

第一行輸出兩個數字接在一起後的結果。

第二行如果是 11 的倍數輸出 "Yes"，反之則輸出 "No"，雙引號不用輸出。

## 範例測資

### 範例輸入 1

11 11

### 範例輸入 2

4646464646646546 121545488476

### 範例輸出 1

1111

Yes

### 範例輸出 2

4646464646646546121545488476

No

# 隊伍調查

## 問題敘述

台灣人最愛做的事情就是排隊了，觀察到這個現象的助教們想要來做一個酷酷的調查。

整個排隊隊伍會有  $n$  個人，一開始排隊隊伍由左到右依序是  $1 \sim n$ ，但是有些人不太守規矩因此常常發生插隊的狀況。

這邊定義如果  $a$  插隊  $b$ ，表示  $a$  跑去排到了  $b$  的左邊，舉例來說如果隊伍目前是  $\{1, 2, 3, 4, 5\}$ ，假如現在 2 插隊 4，隊伍就會變成  $\{1, 3, 2, 4, 5\}$ ，雖然你一定很好奇為什麼要插後面的隊，但先不要問這麼多。

助教們要做的調查必須拜託你設計一個有趣的程式，這個程式要能支援三種操作。

操作 1：輸入兩個正整數  $a, b$ ，表示現在發生了一個插隊事件， $a$  插隊  $b$ 。

操作 2：輸入一個正整數  $p$ ，表示現在要查詢隊伍由左邊數來第  $p$  個人是誰。

操作 3：結束程式。

相信聰明的你已經想到要怎麼做了吧！請你設計一個可以滿足題目要求的程式出來吧！

## 輸入說明

第一行輸入一個正整數  $n$ 。

接下來輸入數行，每行的第一個數字表示操作的類型，再依據操作的類型輸入相對應數量的參數。

保證  $n$  的大小不超過  $2 \times 10^5$ ，且操作 1, 2 輸入的參數都會介於  $1 \sim n$  之間，且操作 2 的次數不超過 100 次。

全部的操作次數加起來不會超過 5000 次，且保證操作 3 一定只會出現在最後一個操作。

另外保證不會有自己插隊自己的情況發生。

## 輸出說明

當執行操作 2 後，根據參數輸出相對應的答案，輸出完要換行。

## 範例測資

### 範例輸入 1

```
5
1 2 4
2 1
2 2
2 3
2 4
2 5
1 4 2
2 1
2 2
2 3
2 4
2 5
3
```

### 範例輸出 1

```
1
3
2
4
5
1
3
4
2
5
```

# 來用程式計算程式的時間複雜度吧！

## 問題敘述

最簡單的時間複雜度估計方式就是計算迴圈總共跑了幾次。

那現在給你一段擁有  $n$  行的部分程式碼，每行只會包含 for 迴圈，格式為 `for(int 變數名稱 A=0; 變數名稱 A< 變數名稱 B; 變數名稱 A++)`，請你設計一個程式計算這一個程式碼的時間複雜度是多少，為了統一格式，請將完整的时间複雜度輸出，運算式左到右依序是迴圈上到下。

運算式乘號請省略掉，但加號不可省略。

兩層以上的迴圈會在每一行的前面空 4 個空白區分，詳細的例子請參考範例輸入。

## 輸入說明

第一行輸入一個正整數  $n$ 。

接下來輸入  $n$  行，每行包含一個字串，保證格式滿足題目敘述。

### 測資範圍限制

- $1 \leq n \leq 10$ 。
- 保證變數名稱都只會有一個字元，且這一個字元一定是小寫英文字母。
- 每一個迴圈內最多只會再有一個迴圈 (意旨不會出現同一層有兩個以上的迴圈的情況)。

## 輸出說明

輸出一個字串，表示其時間複雜度，格式為  $O(\text{運算式})$ 。

## 範例測資

### 範例輸入 1

```
3
for(int i=0;i<n;i++)
    for(int k=0;k<n;k++)
for(int i=0;i<n;i++)
```

### 範例輸入 2

```
1
for(int i=0;i<m;i++)
```

### 範例輸入 3

```
5
for(int i=0;i<n;i++)
for(int i=0;i<m;i++)
for(int i=0;i<n;i++)
for(int i=0;i<b;i++)
for(int i=0;i<n;i++)
```

### 範例輸入 4

```
10
for(int i=0;i<n;i++)
    for(int k=0;k<n;k++)
        for(int j=0;j<m;j++)
            for(int t=0;t<q;t++)
for(int i=0;i<n;i++)
    for(int k=0;k<n;k++)
        for(int j=0;j<m;j++)
            for(int t=0;t<q;t++)
for(int q=0;q<a;q++)
for(int h=0;h<r;h++)
```

### 範例輸出 1

$O(nn+n)$

### 範例輸出 2

$O(m)$

### 範例輸出 3

$O(n+m+n+b+n)$

### 範例輸出 4

$O(nnmq+nnmq+a+r)$

# 餘弦相似度

## 問題敘述

Colten 的科展為了判斷兩個序列的相似度而使用了餘弦相似度來作為判斷的依據。

餘弦相似度運用到三角函數與向量內積的概念，假設現在給定兩個  $n$  維的向量  $A, B$ ，那麼這兩個向量的餘弦相似度 *Similarity* 的計算方式如下：

$$Similarity = \frac{A \cdot B}{|A||B|} = \frac{\sum_{i=1}^n (A_i \times B_i)}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}}$$

Colten 現在有兩個  $N$  維的向量  $X, Y$ ，他想要設計一個程式來幫忙計算  $X$  向量跟  $Y$  向量的餘弦相似度，但是他為了製作科展的報告書而沒有時間，你能幫忙他嗎？

## 輸入說明

第一行輸入一個正整數  $N$ 。

第二行依序輸入  $N$  個正整數  $X_i$ 。

第三行依序輸入  $N$  個正整數  $Y_i$ 。

### 測資範圍限制

- $1 \leq N, X_i, Y_i \leq 100$

## 輸出說明

輸出只有一行，包含一個浮點數，表示  $X$  向量與  $Y$  向量的餘弦相似度 (請無條件捨去到小數點後第二位)。

## 範例測資

### 範例輸入 1

```
3
10 20 30
40 50 60
```

### 範例輸出 1

```
0.97
```



