

## TFcis13 下學期留社考 程式設計能力上機測驗

命題範圍：基本、字串、資料結構、圖論

預定考試日期：100 年 5 月 4 日

考試注意事項：

1. 本測驗時間共 2 小時 30 分鐘。
2. 本測驗直接在電腦上作答，考試時間結束後須繳交原始碼檔案，題目卷則不收回。(有需要可在題目卷上做計算)
3. 若題目有敘述不清的部分，請於考試中提問。
4. 每題可只答對部分測試資料。
5. 每題均有執行時間限制，超過時限該測資以 0 分計。
6. 測試程式以原始碼為主，即使已編譯成可執行檔，測試時仍會重新編譯。
7. 輸出的最後一個字元，若無特殊提示，皆為換行字元「\n」。
8. 本次考試限用 C 或 C++ 語言。
9. 原始碼檔案名稱按照題號依序為 PRO1、PRO2、PRO3、……，C 原始碼副檔名為 c，C++ 則為 cpp。
10. 考試分數會於評分完成後公布於索尼小站 (sony.tfcis.org) 的 TFcis13 看板，如果不希望成績被公布，請於考試中提出，或在程式碼檔案的第一行以註解標示。
11. C 編譯方式：gcc -O2 -Wall -static -pipe -DTFCIS13 -ansi
12. C++ 編譯方式：g++ -O2 -Wall -static -pipe -DTFCIS13
13. 編譯器：GCC 4

扣分規則表：

該題 0 分	使用作業系統特有的 API 或任何非標準函式 (例：stricmp()、_findfirst())
該題 0 分	程式對檔案進行、寫入的操作 (使用檔案操作函式，但沒有真正操作檔案則不限制)
該題 0 分	使用的記憶體大小超過一般電腦安裝的總量 (MLE)
該題 0 分	透過外部程式、系統指令來協助作答
該題 0 分	編譯錯誤 (Compilation Error, CE)
該題扣 1 分	檔案未依規定命名
該題扣 1 分	因結尾有 system("pause")、while(1) 或其他項目而導致程式執行完未結束
該題扣 1 分	程式正常結束，卻未回傳 0 (可能是 main() 未加上 return 0)
該測資得分減半	因缺少或多出空格、換行，而導致格式錯誤 (即 Presentation Error, PE)

## 第 1 題：基本題

執行時限：2 秒 配分：15 分

由於 TFcis13 上學期的留社考被批評為「鑑別度過低」，甚至有人說「鑑別度趨近於 0」，不過我想大家應該同意 0 分的被踢掉吧。(?)

因此這次 T12 程設教學(lantw44)決定在第一題放上一個超基本的題目，而且不標明「加分題」，以免引來誤會。(不過其實上學期留社考第 8 題不是我出的)

請寫一個程式來計算最小公倍數。

### 輸入

本題有多筆測資，每筆測資以換行字元隔開。

每筆測資有兩個數字  $a$ 、 $b$ ，以  $[a,b]$  的方式表示。

輸入以 EOF 做為結尾。

### 輸出

對於每筆測資，輸出  $a$ 、 $b$  的最小公倍數。

### 範例輸入

```
[1, 5]
[3, 7]
[10, 15]
[8, 6]
```

### 範例輸出

```
5
21
30
24
```

### 測資說明

對於 20% 的測試資料， $a, b \leq 10^4$ 。

對於 40% 的測試資料， $a, b \leq 10^9$ 。

對於 100% 的測試資料， $a, b \leq 10^{19}$ 。

可以保證答案都能用 64-bit unsigned integer 儲存。

## 第 2 題：圖書館管理員

執行時限：2 秒 配分：15 分

某偏遠地區的國中，因為人手不足，所以全校所有實驗室和圖書館都是由同一個人管理。工作了幾個月後，他開始感覺到疲累不堪，尤其整理書架上的圖書，更是使他煩惱。

不過他到最近才發現，原來這間圖書館的書架有一項特殊功能：可以自動將書排列整齊。但這不代表他不需要工作，因為書被借走後再放回時，大家總是喜歡直接把書放在最後面，時間一久，整個書架上書的順序就亂掉了。

由於他大多數時間都在管理實驗室，於是他打算每隔一段時間，再將圖書館的書排好。但他覺得每天都到圖書館去檢查所有書的排列狀況是一件很麻煩的事。由於他可以透過電腦得知每本書的借閱和歸還情形，因此他希望能有個程式為他模擬書架上的圖書排列狀況，並計算目前書本的零亂程度。

這位圖書館管理員恰好是你某位親戚的朋友，他急需你的幫助！現在，請寫一個符合他要求的程式。

### 輸入與輸出

第一行有個整數  $n$  ( $n \leq 100000$ )，代表書架上的書在還沒開放外借時，一共有幾本書。這些書由左到右依序編號為  $1 \sim n$ 。

接下來有多行指令(一行不超過 200 字)，以下解釋各個指令的意義：(指令數量不超過 10000，遇到 EOF 結束程式)

a	在書架上加入一本書(其實是有人還書或是購買新書)。這本書會被放在最右邊，而且它的編號規定為「原先最右側書本編號+1」。
c	顯示目前的書本數量。例如有 5 本書，則輸出「5 books」，書本數量是 0 或 1，記得 book 用單數！
d	從書架上刪除一本書(其實是書被借走)，後面接一個整數(可以保證一定會接整數，不會漏掉)代表要刪掉的圖書編號，右側的書會全部左移來填補空位。如果並沒有這本書，請輸出一行「Not found」。(其他書本的編號都不會改變)
e	計算目前圖書的凌亂程度。凌亂程度的定義是將每本書的編號減去左側書本的編號的總和，減去書本總數，再加 1。如果書本數量是 0 或 1，則凌亂程度定義為 0。
q	列出目前在書架上所有書本編號，由小排到大，以換行字元隔開。(每個測資中只會出現一次)

如果指令的首字母錯誤(不區分大小寫)，則顯示「Invalid command」；若首字母正確，之後有多餘的字元，仍視為指令正確。

### 範例輸入

```
5
Delete 3
c
anonymous
d 3
e
query
```

### 範例輸出

```
4 books
Not found
1
1
2
4
5
6
```

### 範例測資說明

範例輸入第 2 行：刪除第 3 本書，此時書架上的是 1、2、4、5。

範例輸入第 3 行：有 4 本書，就輸出「4 books」。

範例輸入第 4 行：加入一本書，此時書架上的是 1、2、4、5、6。

範例輸入第 5 行：沒有第 3 本書，輸出「Not found」。

範例輸入第 6 行：凌亂程度是 $(2-1)+(4-2)+(5-4)+(6-5)-5+1=1$ 。

範例輸入第 7 行：輸出 1、2、4、5、6 即可。

### 第 3 題：莫名其妙的要求

執行時限：2 秒 配分：20 分

某一天突然班上有個同學向你說：「借我一臺可以算 80 位數的計算機！」。而且他很急著要。借他工程用計算機後，他卻說必須每一位數都精準，但是這臺計算機只能以科學記號顯示前幾位數，並不符合他的要求。

由於他一直都沒說為什麼要做這種 80 位數的計算，你原本已經決定不理他了，不過他卻在這時說：「如果在放學前沒弄出來，一項重要的報告就交不出來了。」

你原本想找 T12 的學長來幫忙，卻發現完全找不到人，於是決定靠自己寫出一個這樣的程式。

這位同學非常開心，於是就說：「既然決定要寫程式，那改成 500 位數好了，而且只需要計算整數的加、減、乘法！」

所以，請寫一個程式來做 500 位數的大型整數運算。

#### 輸入

本題有多筆測資，每行有一筆測資，以 EOF 做為結尾。

每行有三個數字，以空白隔開。第一個數字代表要進行的運算，1 表示加，2 表示減，3 表示乘；第二和第三個數字都是要計算的數字，可以保證都不超過 500 位數。

所有數字都是非負整數，減法時第二個數字一定比第三個數字大。

#### 輸出

對於每筆測資輸出計算結果。

注意不論是輸入還是輸出，數字的第一位數都不可為 0。(恰好等於 0 除外)

#### 範例輸入

```
1 885 9877
2 4492 555
3 668 1248
```

#### 範例輸出

```
10762
3937
833664
```

#### 測資說明

在 20% 的輸入檔中，數字不超過  $10^{17}$ 。

其餘輸入檔都有可能超過  $10^{17}$ 。

## 第 4 題：模糊的地圖

執行時限：2 秒 配分：25 分

T12 社長(sb283)在路上檢到了一張古地圖，然後很不小心地就拿到了社部放著，想說沒有什麼用。

這張地圖恰好被一位要做歷史報告的同學在上電腦課時看到了，他正好要利用這張圖的資訊來完成他的報告。但這張圖因為保存不當，有些線條已經脫落了。這是一張關於某地區的海上島嶼的地圖，可以看得出島上的各個地點以及相連的道路，但島嶼的輪廓已經不清楚了。

可惜這張圖是黑白的，對海洋也沒有特別標示，因此這位同學決定藉由相連的道路來判斷島嶼輪廓。可以確定的是當地沒有「跨海大橋」或「海底隧道」，只有兩地相連，就代表在同一座島上。一座島上至少有一個地點。

不過由於地圖實在太複雜，他想請 T13 資訊社的社員來幫他寫個程式對這張地圖進行分析，這樣他可以在還不知道輪廓的情況下，知道兩地是否相連，同時也幫助他重新畫出各個島嶼的輪廓。

因為他急於知道答案，所以他可能會隨時詢問「以目前的狀況來說，一共有幾座島嶼」，即使這會高估真實數值，但他仍希望有這項功能。因此，他可能還沒輸入完地圖的資訊(其實是地圖上那層厚厚的灰塵還沒全部擦掉)，就詢問島嶼數量。

### 輸入

本題只有一個測試資料。

第一行有兩個數字  $n$ 、 $m$ ，表示這張地圖上有  $n$  個地點(點的編號是  $1 \sim n$ )， $m$  條道路。

接下來每行有三個數字  $a$ 、 $b$ 、 $c$ 。若  $a$  為 1，表示  $b$ 、 $c$  兩點間有一條道路；若  $a$  為 2，表示詢問  $b$ 、 $c$  兩地以目前狀況來說是否在同一座島上；若  $a$  為 3，則忽略  $b$ 、 $c$ ，直接輸出「以目前的狀況來說，一共有幾座島嶼」。

地圖讀入完畢後，若  $a$ 、 $b$ 、 $c$  均為 0，則表示程式結束。程式結束前要輸出地圖上共有幾座島嶼。

保證  $a$ 、 $b$ 、 $c$  都可以用 32-bit signed integer 儲存。

### 輸出

詢問島嶼數量時，請輸出一個數字代表地圖上有幾座島。

詢問是否在同一座島時，如果在同一座島上請輸出「Yes」，不在同一座島上請輸出「No」。

### 範例輸入

```
3 2
1 1 2
2 1 2
2 1 3
3 0 875
1 1 3
2 2 3
3 8745 5648987
0 0 0
```

### 範例輸出

```
Yes
No
2
Yes
1
1
```

### 測資說明

對於 40% 的測試資料， $n \leq 110$ ， $m \leq 110$ 。

對於 100% 的測試資料， $n \leq 100000$ ， $m \leq 100000$ 。

## 第 5 題：做麵包

執行時限：5 秒 配分：25 分

不用懷疑，本題標題「絕對沒有」暗指任何其他人、事、物，絕對只有字面上的意思。

某間麵包店每天會固定用掉 1 包麵粉(你可以假設袋子很大)，但麵包店內倉庫的每包麵粉保存期限不一定相同，而每天所選的一定是沒有過期的麵粉(因為這不是黑心商店)。但是這間店的管理不佳，有時麵粉會用不完，導致浪費的情形發生。

為了避免這樣的情形發生，選擇要使用的麵粉非常重要，已知每次的選擇都會盡量避免以後發生浪費(也就是最佳情形，我相信大家應該都知道要怎麼做)。

請你寫一個程式幫忙評估麵粉的買進速率會不會造成浪費的情形發生。

以下日期都以「第幾天」的方式來計算。工作開始時是第一天。注意每天開始時，會先買進麵粉，才開始丟棄過期麵粉和製作麵包。

### 輸入

本題只有一個測資。

每一行代表那天購買麵粉的情形。每行的第一個數  $P$ ，代表那天清晨買進  $P$  包麵粉。接下來在同一行會有  $P$  個數，以空白分隔，代表這些麵粉的保存期限(指的是到期日，超過日期就不能再使用)。請注意不是每天都會買進麵粉。

$P$  為 -1 表示輸入結束。

### 輸出

如果當天有過期的情形發生，請輸出發生的日期，以及那天要丟棄的麵粉數量。(以一個冒號和空格隔開)

如果麵粉都能如期用完，則在輸入結束時輸出「Good」。不論是否能如期用完，都要在程式結束前輸出目前剩下的麵粉數量。

詳細規則請參考範例輸出。

### 範例輸入 1

```
3 3 4 5
2 3 8
0
-1
```

### 範例輸出 1

```
Good
2
```



## 範例輸入 2

```
5 1 1 1 1 1
6 2 3 4 5 6 6
6 3 3 3 4 5 5
0
0
-1
```

## 範例輸出 2

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

## 測資說明

日期一定可以用 32-bit signed integer 儲存。

倉庫最多可以儲存 1000000 包麵粉，可以保證過程中絕對不會超過這個數量。

對於 40% 的測試資料，天數  $\leq 110$ ，保存期限  $\leq 220$ 。

對於 60% 的測試資料，天數  $\leq 1100$ ，保存期限  $\leq 2147483647$ 。

對於 100% 的測試資料，天數  $\leq 110000$ ，保存期限  $\leq 2147483647$ 。

## 提示

範例輸入 1 第 1 天結束時：剩下 2 包。

範例輸入 1 第 2 天結束時：剩下 3 包。

範例輸入 1 結束在 3 天，沒有浪費的情形，輸出「Good」。第 3 天剩下兩包，輸出 2。

範例輸入 2 第 1 天：買進 5 包，用掉 1 包，所以當天結束剩下的是 1、1、1、1。

範例輸入 2 第 2 天：買進 6 包，丟棄 4 包。

以此類推，最後剩下 4 包。