

# 2013 網際網路程式設計全國大賽

## 國中組決賽

- 題目：本次比賽共 七 題 (含本封面共 18 頁)。
- 題目輸入：全部題目的輸入都來自**標準輸入**。
- 題目輸出：全部題目的輸出皆輸出到螢幕 (**標準輸出**)。  
輸出和裁判的答案必須完全一致，英文大小寫不同或多餘空白換行字元皆視為錯誤答案。
- 時間限制：每一題的執行時間限制如下表所示。  
其間執行的電腦上不會有別的动作、也不會使用鍵盤或滑鼠。
- 比賽中上傳之程式碼請依照以下規則命名：
  1. 若使用 C 做為比賽語言則命名為 `pa.c`, `pb.c`, 以此類推。
  2. 若使用 C++ 做為比賽語言則命名為 `pa.cpp`, `pb.cpp`, 以此類推。
 未按照此規則命名之程式碼將可能因此得到 `Compilation Error`。
- `long long` 型別的整數使用方式請參考下一頁。
- `cin` 輸入經測試發現速度遠慢於 `scanf` 輸入，答題者若使用需自行承擔因輸入速度過慢導致 `Time Limit Exceeded` 的風險。

表 1: 題目資訊

	題目名稱	執行時間限制
題目 A	新·烤餅乾	2 秒
題目 B	蚯蚓搬家問題	2 秒
題目 C	紀念品排隊問題	2 秒
題目 D	可樂，可樂，更多的可樂	2 秒
題目 E	可魚果餵魚問題	2 秒
題目 F	捷運路線	2 秒
題目 G	小可魚買點心	2 秒

## 2013 網際網路程式設計全國大賽 解題程式輸入輸出範例

C 程式範例：

```
1 #include <stdio.h>
2 int main(void)
3 {
4     int cases, i;
5     long long a, b;
6     scanf("%d", &cases);
7     for(i = 0; i < cases; ++i)
8     {
9         scanf("%I64d %I64d", &a, &b);
10        printf("%I64d\n", a + b);
11    }
12    return 0;
13 }
```

C++ 程式範例：

```
1 #include <iostream>
2 int main(void)
3 {
4     int cases;
5     std::cin >> cases;
6     for(int i = 0; i < cases; ++i)
7     {
8         long long a, b;
9         std::cin >> a >> b;
10        std::cout << a + b << std::endl;
11    }
12    return 0;
13 }
```

# 題目 A

## 新 · 烤餅乾

執行時間限制: 2 秒

不久前舉辦的一年一度烤餅乾世界大賽，很不幸地因為大會出的題目有點複雜，導致選手都在計算自己有多少模具可以使用，而沒有足夠時間將選手們各自的烤餅乾技巧完全發揮出來，以致於烤出來的餅乾都很普通，沒有到驚為天人的美味。大會對此感到非常可惜，認為這樣就喪失了原本舉辦烤餅乾大賽的真諦，因此大會決定今年要破例，在今天舉辦今年第二場的烤餅乾世界大賽，讓各位選手能夠做出最得意的作品！

小櫻雖然在前一陣子的比賽中，因為沒有完全發揮所長而沒有得名，但是這次不一樣，對面即將到來的烤餅乾世界大賽，小櫻非常地有把握，相信自己一定可以在自己的堅強實力以及你的幫助之下，重新奪回冠軍！

這次的大會決定要做一些大更動，首先更換的是模具，這次的模具五花八門，各種形狀都有；再來賽制的部分也改為回合制，一個回合是兩個人，比賽題目當場公布，公布題目之後再挑選模具進行烤餅乾。然而由於每種模具都只有一個，因此先挑的人會有優勢，為了避免不公平以及增加比賽的刺激感，大會制訂了一個規則：

1. 兩個人先抽一個公正的六面骰，六面骰的每一面都有一個數字，兩人的六面骰的數字都不會重複 (總共會有 12 個不同的數字)
2. 擲骰子，朝上的那一面數字較大者可以獲得優先選擇模具的權利

在擲骰子前就可以看到自己以及對手的所有數字為多少，小櫻想知道她總共可以有多少種情況可以獲得優先權呢？

## ■ 輸入說明

輸入的第一行有一個正整數  $T(T \leq 100)$ ，代表測試資料的組數。

每一組測試資料有兩行，每行有六個正整數  $N_i(0 < N_i \leq 10^6)$ ，各以一個空白隔開，第一行代表小櫻獲得的六面骰上的六個數字；第二行代表對手獲得的六面骰上的六個數字。每一組測試資料中的數字皆不重複。

## ■ 輸出說明

對於每一筆測試資料請輸出一行，包含一個整數表示小櫻有幾種情況會贏。

## ■ 範例輸入

```
2
5 10 15 20 25 30
2 4 8 16 32 64
4 2 1 6 5 3
7 11 9 10 8 12
```

## ■ 範例輸出

```
20
0
```

## ■ 範例說明

- 第一筆測試資料，小櫻有以下幾種情況會贏：
  - 擲出 5，可贏過對手 2, 4  $\Rightarrow$  2 種情況會贏
  - 擲出 10，可贏過對手 2, 4, 8  $\Rightarrow$  3 種情況會贏
  - 擲出 15，可贏過對手 2, 4, 8  $\Rightarrow$  3 種情況會贏
  - 擲出 20，可贏過對手 2, 4, 8, 16  $\Rightarrow$  4 種情況會贏
  - 擲出 25，可贏過對手 2, 4, 8, 16  $\Rightarrow$  4 種情況會贏
  - 擲出 30，可贏過對手 2, 4, 8, 16  $\Rightarrow$  4 種情況會贏

因此共 20 種情況會贏。

- 第二筆測試資料，由於小櫻所有數字皆小於對方的數字，因此 0 種情況會贏。

# 題目 B

## 蚯蚓搬家問題

執行時間限制: 2 秒

魚會吃蚯蚓是眾所皆知的事情，而在可魚國的大可魚們也時常會捕食蚯蚓。

冬天會讓食慾變得很好，魚也不例外。所以大可魚們在每年的冬季會舉辦蚯蚓捕食大會。而住在可魚國的地底下的蚯蚓們就十分可憐了，在這個天敵環伺的季節，一時風聲鶴唳，蚯蚓自危。

因此，在天敵每年大規模嚴重地侵擾下，住在可魚國泥土中的蚯蚓們，終於不堪其擾，最近大舉搬遷到胖胖兮國。

你是其中一隻搬家的蚯蚓，平常喜歡收集各種年份出產的酒，並把酒按照年份在酒櫃上排成一排，十分壯觀。

可是好不容易搬完家後，發現你本來按照年份收藏的順序在搬家的過程中被弄的亂七八糟。在經過一番努力後，你把每一瓶酒應該要放在哪一格都重新算了出來。然而接下來才是最辛苦的事情 — 搬酒。

由於酒瓶對單一隻蚯蚓來說過於巨大，你得從搬家公司雇用更多的蚯蚓來搬酒。

可是雇用來的蚯蚓們對酒大多都沒什麼研究，所以他們發現如果一次搬動太多瓶酒的話，他們很容易搞混到底哪瓶是哪瓶。最後，搬家公司想出了一個辦法。他們的員工每次只會將櫃子上的兩瓶酒抽出來，交換位置後放回。每次搬動兩瓶酒的費用為這兩瓶酒的重量和。

由於請來的蚯蚓不會幫你省錢，也不清楚你想要做什麼，所以你需要自己規劃要搬移哪些酒瓶，然後告訴他們交換酒瓶的順序。你想知道在最佳的搬移策略下將酒瓶排好序需要花多少錢。

### ■ 註腳

- 未成年請勿飲酒。
- 飲酒過量，有害健康。

## ■ 輸入說明

輸入的第一行有一個正整數  $T(T \leq 15)$ ，代表測試資料的組數。

每一組測試資料有三行。第一行有一個正整數  $N(N \leq 100000)$ ，表示總共有  $N$  瓶酒。

第二行有  $N$  個正整數  $a_i$ ，各以一個空白隔開，表示第  $i$  瓶酒應該要放在第  $a_i$  格。保證 1 到  $N$  在  $a_i$  恰各會出現一次。

第三行有  $N$  個正整數  $w_i$ ，各以一個空白隔開，表示第  $i$  瓶酒的重量為  $w_i(1 \leq w_i \leq 100)$ 。

## ■ 輸出說明

對於每一筆測試資料輸出一行，包含一個整數表示將所有酒排好序需要花多少錢。

## ■ 範例輸入

```
2
4
2 1 4 3
10 20 30 40
3
2 3 1
10 20 30
```

## ■ 範例輸出

```
100
70
```

## ■ 範例說明

對於第二筆測試資料，一個最佳的搬移策略如下 (數字  $i$  表示一開始是第  $i$  瓶酒)，

$$[1, 2, 3] \rightarrow [2, 1, 3] \rightarrow [2, 3, 1]$$

花費為  $30 + 40 = 70$ 。

# 題目 C

## 紀念品排隊問題

執行時間限制: 2 秒

皮皮是一個名為「超 ★ 可魚」的偶像團體的忠實粉絲 (fans)，目前這個團體舉辦過的每一場演唱會他都有參加。

而最近這幾年的演唱會，除了本身的演出之外，販賣的週邊商品往往造成了所謂的「第二個戰場」。在「限量」二個字的面前，幾乎所有死忠粉絲都會為之瘋狂。然而，所謂的週邊商品，已經幾乎變成有錢也不一定能夠買得到的東西了。尤其是比較有名的歌手或團體 (例如「超 ★ 可魚」就是其中之一) 的演唱會週邊商品，更是時常「一份難求」。

這次，皮皮正打算參加 2013 年底所舉辦的「超 ★ 可魚之 2013 年終巡迴演唱會」，而他當然也不會錯過這次演唱會的週邊商品販賣。為了買到儘量多的週邊商品，皮皮 (以及其他忠實粉絲們) 往往需要一大清早就到場排隊才能夠買到想要的商品。

在冗長的等待過程中，皮皮突然發現到一個有趣的現象：「由於這次已經是巡迴演唱會的最後一站，有些在前面排隊的粉絲已經擁有一些準備要販賣的商品了！」例如皮皮就發現前面有一位粉絲已經穿著這次巡迴演唱版本的限量 T-shirt，而有另一位粉絲已經戴著專屬護腕。根據皮皮的認知，「超 ★ 可魚」的粉絲們都非常 nice，會傾向於把自己已經有的東西留給別人，故我們可以假設「如果一位粉絲已經有了某樣物品，則那位粉絲就不會再購買該物品」。

在排隊之餘，皮皮想要根據排在前面的人的「已購買」資訊，來推測出「運氣最差時至少可以買到多少種商品」。簡單來說，皮皮想要知道，「就算前面每個人都買了所有他/她還沒有擁有的商品，皮皮仍確定可以買到多少種商品」。

在這題中，我們假設每位粉絲每項商品最多只會**購買一份**；除此之外，皮皮會觀察到排在他前面的所有粉絲們所擁有的物品。

## ■ 輸入說明

輸入的第一行有一個正整數  $T$  ( $T \leq 50$ )，代表測試資料的組數。

每一組測試資料的第一行有兩個整數  $S$  ( $1 \leq S \leq 100$ ) 和  $N$  ( $1 \leq N \leq 100$ )，中間以一個空白隔開。 $S$  代表這次所販賣的商品總數， $N$  為排在皮皮前面的人數。

之後有  $S$  行，每一行即代表一項這次要販賣的演唱會週邊商品，包含商品名稱及該商品的總販賣數量，中間以一個空白隔開。

接著有  $N$  行，第  $i$  行包含第  $i$  位排在皮皮前面的粉絲的擁有物品清單。每一行首先有一個正整數  $k_i$ ，代表第  $i$  位粉絲已經擁有的商品數量。之後有  $k_i$  個字串，代表第  $i$  位粉絲擁有的商品名稱。商品數量與名稱各以一個空白隔開。

所有出現的商品名稱長度都不會超過 128，並且，所有的商品名稱都是以英文字母組成的 (大小寫視為不同)。另外，每種商品的販賣數量保證不會超過 1000。

## ■ 輸出說明

對於每一筆測試資料，請輸出一行。

- 如果皮皮很可憐，什麼都買不到的時候，請輸出 “Need to be lucky” (不包含雙引號)。
- 反之，請輸出一個正整數，表示皮皮至少可以買到幾種商品。

## ■ 範例輸入

```
3
5 4
Tshirt 10
Watch 4
Towel 3
HandLight 10
LimitedBook 1
1 Tshirt
1 Tshirt
2 Tshirt Watch
2 Tshirt HandLight
2 2
LimitedCD 1
Handring 1
1 Handring
1 LimitedCD
2 2
ABCDEFGHIJKLMNOPQRSTUVWXYZ 1
abcdefghijklmnopqrstuvwxyz 1
1 ABCDEFGHIJKLMNOPQRSTUVWXYZ
1 ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

## ■ 範例輸出

```
3
Need to be lucky
1
```

本頁留白。

# 題目 D

## 可樂，可樂，更多的可樂

執行時間限制: 2 秒

口渴可樂公司近期推出了兩種相同價格的新口味可樂，分別是香甜的蜂蜜可樂與溫暖的生薑可樂！為了回饋一直以來愛好可樂的廣大群眾，在新產品上市的一個月內各家超商都有下列的優惠活動！

- 蒐集  $A$  個蜂蜜可樂的空瓶子可兌換一瓶全新的生薑可樂
- 蒐集  $B$  個生薑可樂的空瓶子可兌換一瓶全新的蜂蜜可樂

可魯非常地喜歡喝可樂，但他身上的錢只能買  $N$  瓶可樂。因此，他希望可以利用優惠活動來讓自己喝到盡量多瓶的可樂。你能幫幫他嗎？

### ■ 輸入說明

輸入的第一行有一個正整數  $T$  ( $T \leq 100$ )，代表測試資料的組數。

每一組測試資料有三個正整數  $N$  ( $N < 10000$ ) 和  $A, B$  ( $1 < A, B < 100$ )，各一個空白隔開。

### ■ 輸出說明

對於每一筆測試資料請輸出一行，包含一個整數表示可魯最多能喝到幾瓶可樂。

## ■ 範例輸入

```
3
10 5 5
11 3 5
20 50 60
```

## ■ 範例輸出

```
12
15
20
```

# 題目 E

## 可魚果餵魚問題

執行時間限制: 2 秒

你是活在胖胖兮國的胖胖兮，你最近說服了胖胖天國的胖胖天開始養胖可魚。胖可魚是一種很不可思議的生物，它唯一會吃的食物便是從可魚國來的可魚果。胖可魚孵化的第一天需要吃一顆可魚果，而且若是它前一天吃了  $k$  顆，隔天就必須吃  $k + 1$  顆，一旦胖可魚沒有吃到所需的可魚果顆數就會馬上餓死。

因為可魚國很遙遠，所以東西並不好買。於是胖胖天下單訂購了  $N$  顆可魚果與 1 顆胖可魚卵，並打算養到胖可魚餓死為止。

然而胖胖天很快就發現如果剩下的可魚果不足以餵飽胖可魚時，有餵跟沒餵是一樣的，胖可魚都會死掉。所以胖胖天決定，當剩下的可魚果不足以餵飽胖可魚時，就放棄餵食讓他死掉，直接再訂購  $N$  顆可魚果與 1 顆新的胖可魚卵。也就是說，胖胖天會拿原本剩下的可魚果 + 新買的  $N$  顆可魚果來餵食新的胖可魚。胖胖天會持續這個策略到有天可魚果剛好吃光，再停止養胖可魚，這樣就完全不會浪費食物了啊！

正當胖胖天沾沾自喜發現了如此絕妙的策略時，他突然想到一個問題：萬一沒有可魚果剛好吃完的一天，該怎麼辦啊！不幸地，胖胖天對於計算大薯價格以外的數學並不在行。所以他決定回來問數學很厲害的你。按照這個策略，有沒有可能把可魚果剛好吃完？如果可能的話，他總共會養過幾隻胖可魚呢？

### ■ 輸入說明

輸入的第一行有一個正整數  $T(T \leq 1000)$ ，代表測試資料的組數。

每一行有一個正整數  $N(N \leq 10^8)$ ，代表胖胖天一次會訂購  $N$  顆可魚果。

### ■ 輸出說明

對於每一筆測試資料，請輸出一行。

- 如果沒有可魚果剛好吃完的一天，請輸出 “Infinite Fish” (不包含雙引號)。
- 反之，請輸出一個整數，代表胖胖天總共會養幾隻胖可魚。

## ■ 註腳

胖可魚卵在運輸的過程中會被適當的保存，所以不必擔心胖可魚提早孵化的問題。另外胖胖天的壽命非常長，也不必擔心胖胖天老死的問題。

## ■ 範例輸入

```
6
1
2
3
5
8
13
```

## ■ 範例輸出

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

# 題目 F

## 捷運路線

執行時間限制: 2 秒

螞蟻們討厭環狀的東西，所以螞蟻城裡面的捷運不會出現環狀線的捷運。特別的是，螞蟻城裏面的捷運都是沿著東西或南北向前進。也就是說，每次捷運轉彎都是直接轉 90 度的直角。

秉持著一年開通一條線的原則，捷運路線網越來越複雜了。而且每新增一條路線，就會造成更多的負擔。於是，在若干年後的今天，螞蟻城的首長打算好好整治一下現在紛亂的捷運路線。

每一條捷運的路線，都會從某個起點出發，然後沿著軌道**不重複地**經過一些站，抵達終點。為什麼要不重複呢？因為如果有重複經過同一站的話，搭捷運的螞蟻們便會覺得浪費時間，尤其是在這個**沒有環狀路線**的交通路網上。

請你幫忙算一算，至少需要幾條不同的捷運路線，才能使得每一段軌道上都至少有一條捷運路線經過？

### ■ 輸入說明

輸入的第一行有一個正整數  $T$  ( $T \leq 100$ )，代表測試資料的組數。

每一組測試資料有兩個正整數  $n, m$  ( $1 \leq n, m \leq 100$ )，中間以一個空白隔開。接下來有  $n$  行，每一行有一個長度為  $m$  的字串。這些字串組成了一張螞蟻城現在的捷運路網圖。

圖上的意義如下，

- **井字號** ('#')，表示捷運的軌道。
- **點** ('.')，代表這個位置沒有東西。
- **大小寫英文字母**，代表這是一個車站。可能有許多不同的車站標示著相同的英文符號。

保證對於任何一個軌道，它緊鄰的四格恰好有兩個字元是軌道或捷運站。

保證不會有兩個捷運站緊緊相鄰，且每一個捷運站周圍的四格之中，至少有一格是軌道。

## ■ 輸出說明

對於每一組測試資料請輸出一行，包含一個整數表示最少需要幾條不同的捷運線，才能讓任何一段軌道都有列車經過。

## ■ 範例輸入

```
2
4 15
.....##B#....C
.....##...#..####
.....##...###...
.A###.....
5 7
A###.D#
...#..#
.F.C##b
.#.#...
.##D...
```

## ■ 範例輸出

```
1
2
```

# 題目 G

## 小可魚買點心

執行時間限制: 2 秒

小可魚非常愛吃點心，每天都要吃很多點心才會開心。在小可魚居住的城市裡，有一間超市叫做豬豬超市，因為他專門販賣各種點心給像豬一樣愛吃東西的魚們 (例如小可魚就是其中之一)。

但是小可魚的家離豬豬超市有一段距離，所以小可魚每次去豬豬超市買點心時，都會一次買很大的量。

豬豬超市有很多個不同的點心攤位。每個攤位都有一份價目表，上面會列出該攤位所販賣的點心種類及單價。

為了方便起見，小可魚會為每個點心攤位各列一張購買清單以方便採買。購買清單上會列出所需點心的名稱和數量。而小可魚到了豬豬超市後，會將每一個攤位所屬的購買清單與其價目表核對，算出在每一個攤位各要花多少錢。

然而，小可魚很粗心大意，他有時候會將攤位上沒有販賣的點心列入購買清單。在這個情況下，他想知道扣掉那些沒有販賣的點心之外，仍要在攤位上花多少錢。

注意：購買清單上的種類名稱和價目表上的種類名稱需要**大小寫都相同**才當做同一種點心。

### ■ 輸入說明

輸入的第一行有一個正整數  $T(T \leq 1000)$ ，代表測試資料的組數，也代表豬豬超市有幾個攤位。

每一組測試資料表示一個攤位以及該攤位所屬的購買清單。每組測試資料的第一行有一個正整數  $N(N \leq 10)$ ，代表攤位有  $N$  種點心。接下來會有  $N$  行，每行會有一個字串  $s$ ，以及一個正整數  $p(p \leq 5000)$ ，中間以一個空白隔開，代表攤位上有賣一種點心叫做  $s$ ，以及它的單價為  $p$ 。

接下來會有一行，有一個正整數  $M(M \leq 10)$ ，表示小可魚打算在這個攤位買  $M$  種點心。接下來會有  $M$  行，每行會有一個字串  $b$ ，以及一個正整數  $a(a \leq 100)$ ，中間以一個空白隔開，代表小可魚打算買  $a$  個  $b$  這種點心。

保證  $s$  和  $b$  只由英文大小寫字母組成，且長度不會超過 20。

## ■ 輸出說明

對於每一組測試資料，輸出一行。

- 如果小可魚的購買清單上的所有食物都有在價目表上出現，請輸出應付的金額。
- 反之，請輸出一個“NO”(不包含雙引號)，與扣掉那些沒有販賣的點心後應付的金額，中間以一個空白隔開。

## ■ 範例輸入

```
2
3
PUFF 30
chips 20
cake 1000
2
chips 10
cake 1
3
OREO 50
pie 20
waffle 100
3
pie 10
oreo 2
waffle 20
donut 10
```

## ■ 範例輸出

```
1200
NO 2200
```