

2012 網際網路程式設計全國大賽 高中組初賽

- 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**。
輸入中可能包含多組輸入，依題目敘述分隔。
- 題目輸出：全部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，英文字母大小寫不同視為答題錯誤。
- 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`。
- `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	神祕的地圖	20 秒
題目 B	老蚯的寶物	5 秒
題目 C	好多星星	15 秒
題目 D	長長的蚯蚓天天長長	5 秒
題目 E	出牌的抉擇	5 秒
題目 F	畫蚯蚓	5 秒
題目 G	電梯向上	30 秒

2012 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：

```
#include <stdio.h>
int main(void){
    int cases, a, b, i;
    scanf("%d", &cases);
    for(i = 0;i < cases;i++){
        scanf("%d %d", &a, &b);
        printf("%d\n", a+b);
    }
    return 0;
}
```

C++ 程式範例：

```
#include <iostream>
using namespace std;
int main(void){
    int cases, a, b, i;
    cin >> cases;
    for(i = 0;i < cases;i++){
        cin >> a >> b;
        cout << a+b << endl;
    }
    return 0;
}
```

題目 A

神秘的地圖

執行時間限制: 20 秒

相傳，有個古老而神秘的帝國，曾經有過非常繁榮的時代，但是經過幾次戰爭以及災禍之後，帝國便在歷史中消失了。但也有另外一派的人相信現在的都市其實是從帝國逐漸演變而成，只是有幾個城市以及連接那些城市的道路消失了。

一天，小小郭在家中的深處找到了一張神秘的地圖，有點像是他所居住的地方，但是在地圖上卻多了好幾個城市，因此他懷疑這就是傳說中古老帝國的地圖。

為了驗證這個說法，小小郭想到了一個辦法：如果帝國是都市的前身，那麼每個現在的新城市一定都是古老帝國中的某個舊城市（彼此有對應關係）。若這些舊城市之間的道路連接狀況與新城市之間的道路連接狀況相同的話，則我們就說帝國與都市有完整對應，代表帝國真的是都市的前身。要注意的是，若兩個城市之間有 k 條道路的話，對應之後也必須要有 k 條道路，才可以說有對應關係！（ $k \geq 0$ ）

但是小小郭覺得這個工作太麻煩了，你能幫幫他嗎？

■ 輸入檔說明

測試資料第一行有一個正整數 T ($T \leq 20$) 表示接下來總共有幾筆測試資料。

每一筆測試資料的第一行有兩個整數 N_1, M_1 以一個空白隔開， N_1 代表地圖上城市的數量，並由 0 編號到 $N_1 - 1$ 、 M_1 代表道路的個數。城市接下來有 M_1 行，每行有兩個整數 a_i, b_i ，代表該條道路連接城市 a_i 及城市 b_i 。（ $0 \leq a_i, b_i < N_1$ ）接下來的一行有兩個整數 N_2, M_2 分別以空白隔開，代表有現在的城市的數量，並由 0 編號到 $N_2 - 1$ ，以及道路的個數。接下來有 M_2 行，每行有兩個整數 p_i, q_i ，代表該條道路連接城市 p_i 及城市 q_i 。（ $0 \leq p_i, q_i < N_2$ ）

我們保證 $N_2 \leq N_1 \leq 10$ 以及 $M_1, M_2 \leq 50$ 。

■ 輸出檔說明

對每筆測試資料輸出一行，如果可以成功的對應，輸出“YES”（不含引號），若無法對應，則輸出“NO”（不含引號）。

■ 範例輸入

```
2
4 4
0 1
1 2
2 3
0 3
3 2
0 1
1 2
4 4
0 1
1 2
2 3
0 3
3 3
0 1
1 2
0 2
```

■ 範例輸出

```
YES
NO
```

■ 範例說明

第一筆測試資料中，將 0 對應到 0，1 對應到 1，2 對應到 2，即可成功對應。

第二筆測試資料中，無法找到任何對應。

題目 B

老蚯蚓的寶物

執行時間限制: 5 秒

傳說中，地球深處住著一隻老蚯蚓—老蚯，他可以藉由把自己砍成兩半而多產生一隻蚯蚓，所以能從一隻蚯蚓變成兩隻，前半段為本尊 (老蚯蚓)，後半段為產生的新蚯蚓。老蚯蚓會持續地把自己砍成兩半，而新產生的每一隻蚯蚓也都擁有一樣的能力來產生更多的蚯蚓。於是過了若干年後，蚯蚓已經遍佈世界的各個角落，泥土裡時常能找到蚯蚓的蹤跡。

而蚯蚓們在地底挖挖挖，常常會找到一些寶物，身為最初的那一隻蚯蚓 ---老蚯，他偶爾會整理一下蚯蚓們找到的寶物。

最近老蚯發現他的收藏多了一台筆記型電腦，但是人類使用的輸入系統和蚯蚓習慣的不太一樣。

於是身為老蚯好友的你，想要幫他寫一個英文打字系統。一開始螢幕上會是一個空字串 (長度為 0 的字串)。接下來會有底下的指令：

- `pushfront L`, 表示在目前字串首插入一個字元 L 。
- `pushback L`, 表示在目前字串尾插入一個字元 L 。
- `undo U`, 表示復原前 U 個指令。

保證每次給的 U 值都不會超過目前的指令數，且 L 將會是大小寫英文字母或數字 (`[A-Za-z0-9]`)。

值得一提的是，`undo` 這個指令，會從現在開始往前連續取消 U 個指令。

如果取消的是 `pushfront L` 這個指令，則把 L 從目前字串首刪除；

如果取消的是 `pushback L` 這個指令，則把 L 從目前字串尾刪除；

但是如果取消的是 `undo x`，則是要依**指令輸入的順序**重新依序執行當時被取消的 x 個指令。

舉個例子來說，如果依序執行了以下的指令，那螢幕上的字串則會這樣變化：

指令	螢幕上的字串
pushfront m	m
pushfront t	tm
pushback t	tmt
undo 2	m
undo 1	tmt
pushback 5	tmt5
pushback 1	tmt51
pushback 4	tmt514
undo 5	tmt

■ 輸入檔說明

輸入的第一行有個正整數 T ($T \leq 20$)，代表測試資料的組數。

對於每一組測試資料，該組第一行都會有個正整數 N ($N \leq 2000$)，表示這組測試資料有 N 個指令。

接下來會有 N 行，每行代表一個指令。

■ 輸出檔說明

對於每一個指令結束後，輸出一行螢幕上目前的結果。

■ 範例輸入

```
2
9
pushfront m
pushfront t
pushback t
undo 2
undo 1
pushback 5
pushback 1
pushback 4
undo 5
3
pushfront 1
pushfront 5
pushback 4
```

■ 範例輸出

```
m  
tm  
tmt  
m  
tmt  
tmt5  
tmt51  
tmt514  
tmt  
1  
51  
514
```

本頁留白。

題目 C

好多星星

執行時間限制: 15 秒

現代人總是過著繁忙的都市生活，偶爾離開都市遠離塵囂，在沒有光害的情形下看著夜空，那漫天星斗總是特別美麗。現在正值冬天，抬頭一看，夜空中最有名的莫過於「冬季大三角」了。而除此之外，夜空中還有著數不清的星星，彼此交織成各式各樣令人玩味的圖形，想必當初前人也是這樣望著夜空而構思出大家耳熟能詳的 88 星座的吧！

想到這裡，曉涵突然想起她的書房裡有一本「星座大百科」，於是她便趕緊找出那本塵封已久的書，並興奮地想要找看看現在夜空中是否有出現書中所介紹的星座。然而，夜空實在太過廣大，有數不清的星座在其中，要人眼一一比對實在太難了。身為曉涵的好朋友，你決定寫一支程式幫她找出同時是書中所介紹並在夜空中出現的星座。

每個在夜空中的「星座」都是由數個星星所組成的，且也為了方便起見，我們將每個星座都當作在二維座標平面上的一群點（彼此分開考慮，每個星座都是獨立的，彼此沒有關係）。注意，我們如果要說兩個星座一樣，必須要它們的所有點（星星）的位置都要匹配才算。也就是說，必須能夠完全對應星座百科中的其中一個星座的每一個點，我們才能視該星座有出現在夜空中。

在夜空中的星座的大小比例可能和百科上的不同，角度也可能不同（也就是說，可能經過旋轉）。只有比例或觀測角度的不同的話，仍視為一樣的星座。（只要經過旋轉縮放後能跟百科上的一樣，便視為一樣，但**不包括**鏡射後相同。）

為了讓題目更簡單一些，我們把每顆星星都當做是「相同的」，也就是說，夜空中的哪一顆星星對到書上的哪顆星星並不重要，只要星座中所有的星星都有所對應就可以了。而對於給定的夜空中的星座，我們不考慮他們的「子集合」。

■ 輸入檔說明

輸入檔的第一行有一個正整數 T ($T \leq 80$)，代表共有幾組測試資料。

對於每組測試資料，我們第一行有一個整數 N ($1 \leq N \leq 88$)，代表曉涵的星座百科中有幾個星座。之後有 N 行，每一行代表書中的一個星座，其中首先會有一個整數 V_i ，代表書中所示的第 i 個星座是由 V_i 個星星組成的。該行後面有 $2 \times V_i$ ($3 \leq V_i \leq 20$) 個整數 $x_1, y_1, x_2, y_2, \dots, x_{V_i}, y_{V_i}$ ，分別代表其共 V_i 個星星的 x, y 座標。

第 $N + 2$ 行有一個整數 M ($1 \leq M \leq 88$)，代表夜空中有幾個星座。之後有 M 行，每一行代表夜空中的一個星座，其中首先會有一個整數 W_j ，代表夜空中的第 j 個星座是由 W_j 個星星組成的。該行後面有 $2 \times W_j$ ($3 \leq W_j \leq 20$) 個整數 $x_1, y_1, x_2, y_2, \dots, x_{W_j}, y_{W_j}$ ，分別代表其共 W_j 個星星的 x, y 座標。

每個星座中的星星的座標都是獨立考慮的，寫成座標只是為了方便描述他們的形狀。（座標和星座之間的位置沒有關係，我們也不用考慮星座的位置）所有給定的 x, y 座標的絕對值都不超過 1000。

■ 輸出檔說明

請告訴曉涵書上有哪些星座有出現在夜空中，請輸出他們的編號（百科中的第幾個星座，按照輸入順序從 1 開始編號）。對於一組測試資料，編號請輸出在同一行，編號請按照順序從小到大輸出。

如果都沒有出現書中的星座的話，請輸出一行 “so sad”（不含引號）。

（註：出現在書中和夜空中的星座不一定是現實中有的星座噢！）

■ 範例輸入

```
3
2
4 1 4 4 2 4 0 0 0
4 0 0 1 1 0 1 1 0
2
4 1 1 3 9 9 5 9 1
4 0 -3 3 0 0 3 -3 0
1
5 2 7 8 3 1 -1 2 -2 7 -7
2
3 -1 -1 2 -1 0 0
5 2 7 8 3 1 -1 2 -2 6 -7
4
4 0 -2 0 4 4 0 -2 0
4 0 0 1 1 2 2 3 3
5 6 -7 2 7 2 -2 8 3 1 -1
5 0 0 0 8 0 10 2 10 2 8
4
4 -1000 -1000 -500 -500 500 500 0 0
4 1 -2 1 4 4 1 -2 1
5 6 -7 2 7 2 -2 8 3 1 -1
5 0 0 0 8 0 10 -2 10 -2 8
```

■ 範例輸出

```
1 2
so sad
2 3
```

題目 D

長長的蚯蚓天天長長

執行時間限制: 5 秒

長長的蚯蚓，斟滿了月光。

這裡有一隻剛出生的小蚯蚓，一開始牠真的非常非常小隻，所以我們可以將牠的長度視為 0 公分。然而身為一隻健康成長的蚯蚓，只要牠每天早上起床的時候長度還沒有達到 N 公分，牠就會骰個公平的 M 面骰 (有相等的機率骰出 1 到 M 之間的任一整數)。假設得到的點數為 $X(1 \leq X \leq M)$ ，則他便會外出覓食吸收天地精華，使得牠的長度增加 X 公分！若牠早上起床的時候長度有達到 N 公分，牠就會停止成長。

小姜身為蚯蚓控，預言這隻蚯蚓停止生長後的長度至少會有 S 公分，請問預言正確的機率有多少呢？

■ 輸入檔說明

測試資料第一行有一個正整數 T 表示小姜預言了多少隻蚯蚓。接下來 T 行，每行依序為三個整數 N, M, S 。

保證 $1 \leq T \leq 514$ ， $1 \leq N, S \leq 10^7$ ， $1 \leq M \leq 50$ 。

■ 輸出檔說明

對每個預言輸出一行，代表小姜猜對的機率。請以百分比輸出並四捨五入至小數點後第一位。

■ 範例輸入

```
3
5 6 9
1 7 3
4 8 6
```

■ 範例輸出

```
12.6%
71.4%
64.4%
```

本頁留白。

題目 E

出牌的抉擇

執行時間限制: 5 秒

大老二是一種流行的撲克牌遊戲，每人各發十三張牌後輪流出牌，每次出牌的組合必須要比前一個人還大，最先打完手上的牌便贏得遊戲。出牌的組合有下列數種：

1. 單張：任何一張牌
2. 對子：兩張同點數的牌
3. 三條：三張同點數的牌
4. 順子：五張點數各差一的連續牌，包含 TJQKA，但所有帶 2 的組合皆不包含
5. 葫蘆：三條加上對子所組成的五張牌，比較大小時先比三條再比對子
6. 鐵支：四張同點數的牌加上任意單張牌，比較大小時先比四張同點數的牌
7. 同花順：相同花色且為順子的五張牌，同花順不計入順子

現在我們有一副手牌，請找出所有可以出牌的組合。而相同種類組合則從小到大輸出，牌面大小原則上先比較組合中最大的牌，若仍一樣則比較次大的牌並依此類推，但葫蘆與鐵支不完全適用此規則。單張牌的大小先看點數再看花色，這兩者的順序如下：

Spade > Heart > Diamond > Club

$2 > A > K > Q > J > T > 9 > 8 > 7 > 6 > 5 > 4 > 3$

■ 輸入檔說明

測試資料第一行有一個正整數 $T (T \leq 500)$ 表示接下來有幾組測試資料。

每組測試資料共有一行，每行共有 13 張牌，每張牌由兩個字元構成，分別表示該張牌的花色與點數。花色使用原英文名的第一個大寫字母表示。

■ 輸出檔說明

對於每組測試資料，首先輸出一行表示為第幾組測試資料，接著依前述順序輸出七行分別表示七種組合，每行依序輸出手牌中該組合可能的情況，每個組合以一個空白鍵分隔，對於一個多張牌的組合，輸出時請照牌面由小到大輸出，如果手牌中無該種組合請輸出 “No exists”(不含引號)。

■ 範例輸入

```
2
SA S2 C7 H3 H4 S4 C5 C6 C8 C9 CJ CK HK
SA S2 S3 H3 H4 S4 C4 S6 C6 C9 CQ CJ CK
```

■ 範例輸出

```
Case #1:
H3 H4 S4 C5 C6 C7 C8 C9 CJ CK HK SA S2
H4S4 CKHK
No exists
H3H4C5C6C7 H3S4C5C6C7 H4C5C6C7C8 S4C5C6C7C8
No exists
No exists
C5C6C7C8C9
Case #2:
H3 S3 C4 H4 S4 C6 S6 C9 CJ CQ CK SA S2
H3S3 C4H4 C4S4 H4S4 C6S6
C4H4S4
No exists
H3S3C4H4S4 C4H4S4C6S6
No exists
No exists
```

題目 F 畫蚯蚓

執行時間限制: 5 秒

小喇魚是一隻很喜歡畫畫的魚，尤其喜歡特寫自然界中嬌小的動植物。然而除了喜歡畫畫外，小喇魚也很懶惰，懶惰到甚至連畫畫時也必得用最簡易的方法來畫。

一天，一隻小蚯蚓找上了小喇魚，希望小喇魚能幫自己畫幅素描。喜愛畫畫、愛護小動物的小喇魚自然是義不容辭，馬上準備了畫具。但由於小喇魚的惰性，他並不打算直接將蚯蚓畫出來。經過小喇魚觀察，他發現蚯蚓身上有許多長得一模一樣的環節。這代表著，他只要作出一部份的蚯蚓的模板，並將模板複印多次、疊起來，便可以將整隻蚯蚓給畫出來了！

為了方便，小喇魚已經把小蚯蚓身上的所有環節都表示成英文字母，一樣的環節會被標上一樣的字母。現在小喇魚會隨機找出小蚯蚓的一段環節，並在複印若干次後畫出整隻蚯蚓。注意懶惰的小喇魚為了方便對齊，所以在疊畫時兩張畫重疊的部分必須完全相等。由於小喇魚很聰明，你可以相信小喇魚選出的那段小蚯蚓一定有辦法組合出整隻小蚯蚓。然而複印畫也是需要時間的，假設該段蚯蚓長度為 p ，總共被複印 k 次，則小喇魚總共需要花掉 $p \times k$ 的時間。雖然小喇魚非常聰明，給定一段小蚯蚓後一定會在複印最少次的狀況下疊出整隻蚯蚓。但因為小喇魚運氣非常差，所以小喇魚一定會選到一段最差的小蚯蚓，使得該段蚯蚓的複印方案 (在複印最少次的狀況下疊出整隻蚯蚓) 所需要的時間最多。試問小可魚究竟會花掉多少時間？

■ 輸入檔說明

測試資料第一行有一個正整數 $T(T \leq 100)$ 表示接下來有幾組測試資料。

每筆測試資料包含一個字串，代表被表示為英文字母的蚯蚓，保證長度不超過 10^6 。

■ 輸出檔說明

每筆測試資料輸出一個整數，代表最差狀況下小喇魚會花掉多少時間。

■ 範例輸入

```
2
abcde
adadadada
```

■ 範例輸出

5 14

■ 範例說明

第一筆測試資料中 `abcde` 本身就是唯一合法的模板，故答案為 $5 \times 1 = 5$ 。

第二筆測試資料中，`adadada` 這個模板在最佳疊法下會用兩次 (中間 5 個字母重疊)，所需時間為 $7 \times 2 = 14$ 。

題目 G

電梯向上

執行時間限制: 30 秒

NPSC 百貨公司就快要重新開幕了！

NPSC 百貨公司的主打為頂樓的摩天輪！它是目前世界上最大的摩天輪，預計可以為 NPSC 百貨公司帶來大量的人潮。

為了因應龐大的人潮，NPSC 百貨公司的老闆決定要更換百貨公司裡的一台電梯，好讓大家不會被塞在樓梯間。但是因為營業時間的關係，電梯一天只能載 k 趟，因此電梯的載重量是一個相當重要的議題。

每天 NPSC 百貨公司都會發放搭乘電梯的號碼牌給民眾，好讓大家排隊從一樓搭電梯到頂樓搭乘摩天輪。

為了減少更換電梯的成本，NPSC 百貨公司希望盡量降低電梯的載重量，於是調查了某天所有顧客的號碼以及體重，想要知道電梯的載重量至少可以承受多少公斤才能在 k 趟內運送完顧客。

■ 輸入檔說明

測試資料第一行有一個正整數 T ($T \leq 30$) 表示接下來總共有幾筆測試資料。

每一筆測試資料的第一行有兩個整數 N, k 以一個空白隔開， N 代表顧客的數量、 k 代表電梯最多運送的趟數 ($1 \leq k \leq 10^3$)。接下來有 N 行 ($1 \leq N \leq 10^5$)，每行有兩個正整數 idx_i, w_i ，代表顧客 i 的號碼為 idx_i ，體重為 w_i 公斤。 ($0 < w_i \leq 100$)

我們保證顧客的號碼 $0 \leq idx_i < N$ ，且每名顧客的號碼皆不重複。

■ 輸出檔說明

對每筆測試資料輸出一行，每行只有一個整數 p ，代表電梯的最大載重量至少要 p 公斤才能在 k 趟內載完所有顧客。

■ 範例輸入

```
2
5 5
0 1
1 2
2 3
3 4
4 5
5 3
0 1
1 2
2 3
3 4
4 5
```

■ 範例輸出

```
5
6
```

■ 範例說明

第一筆測試資料中，每個顧客都各自搭一趟電梯，最大載重量需要 5 公斤。

第二筆測試資料中，號碼 0, 1, 2 的顧客搭乘一趟電梯，剩下兩名顧客各自搭乘一趟電梯，最大載重量需要 $1 + 2 + 3 = 6$ 公斤。