

# 2009 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（不含本封面共 18 頁）。
- 題目輸入：全部題目的輸入都來自輸入檔，請依題目以及下表指示開啓指定的輸入檔讀取輸入。每個輸入檔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指定的輸出檔。檔名如下表，請注意都是小寫。
- 時間限制：裁判將限制每次執行時間為十秒。其間執行的電腦上不會有別的動作、也不會使用鍵盤或滑鼠。

表一、題目資訊

	題目名稱	輸入檔名	輸出檔名
題目 A	草地上的松鼠	pa.in	pa.out
題目 B	KjumpingCube	pb.in	pb.out
題目 C	跑跑卡丁車	pc.in	pc.out
題目 D	有限狀態自動機	pd.in	pd.out
題目 E	糾結的水管	pe.in	pe.out
題目 F	風鈴	pf.in	pf.out
題目 G	整理 CD	pg.in	pg.out

## 題目 A

# 草地上的松鼠

輸入檔: pa.in / 輸出檔: pa.out

前一陣子松鼠王爬到樹林中最高的地方，他發現遠處有一片優美舒適的草地，於是他帶著整群松鼠去那邊玩耍。

玩了幾天，大家都累了，現在松鼠王想要帶著整群松鼠回到他們原本住的樹林。為了避免有些松鼠太貪玩而落單，松鼠王決定叫大家先整隊排成一列，確定全部到齊以後再一起出發。

現在大家已經集合並且排好隊了，正要出發的時候卻發現有一隻小松鼠不見蹤影。在這焦急的當下，松鼠王眼尖地發現這隻小松鼠還在草叢裡舒服地睡覺。為了維持松鼠王國的紀律，松鼠王必須把它叫醒。不過為了維持親民的形象，松鼠王不能生氣。

在把這隻小松鼠叫醒以後，松鼠王要讓他排到隊伍的某個地方。他不能排在隊伍的最前面，因為需要專業的嚮導帶領大家回去。他也不能排在最後面，以免因為打瞌睡而倒在半路上。由於其他松鼠已經排好隊，要在他們之中挪出一些空間給這隻小松鼠也有點困難。

聰明的松鼠王想到了一個好辦法：在所有相鄰的松鼠中找出距離最遠的兩個，然後就讓這隻小松鼠排在他們中間。這裡距離的定義是兩隻松鼠所在位置的差的絕對值。

由於松鼠的數量有點多，隊伍也很長，松鼠王需要你的幫忙才能很快地知道這隻小松鼠應該排在哪裡。

### ■ 輸入檔說明

第一行有一個整數  $T$ ，代表接下來有幾組測試資料。

每一筆測試資料有兩行，第一行有一個數字  $N$ ，代表隊伍上排了幾隻松鼠。

第二行有  $N$  個整數， $x_1, x_2, \dots, x_N$ ，分別代表這  $N$  隻松鼠的位置。

數字越小的表示在隊伍的越前面。

( $2 \leq N \leq 10000$ ,  $0 \leq x_1 < x_2 < \dots < x_N \leq 200000000$ )

**■ 輸出檔說明**

對每筆測試資料請輸出一個整數  $k$ ，代表這隻小松鼠應該被排在第  $k$  隻松鼠和第  $k + 1$  隻松鼠中間。若有距離相同的情形，請讓這隻小松鼠排在越前面越好。

**■ 範例輸入**

```
1
5
1 3 6 8 9
```

**■ 範例輸出**

```
2
```

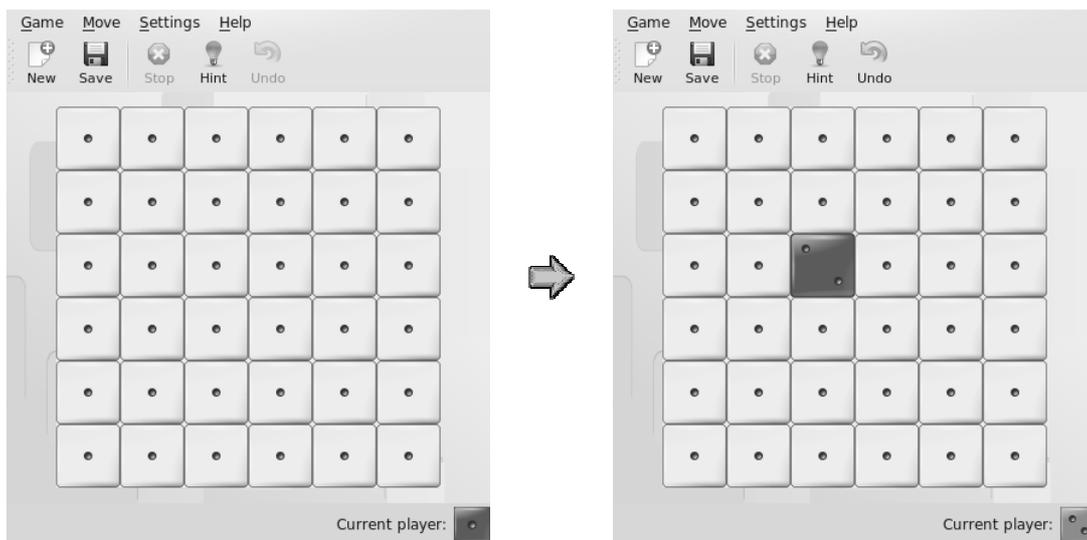
## 題目 B

# KjumpingCube

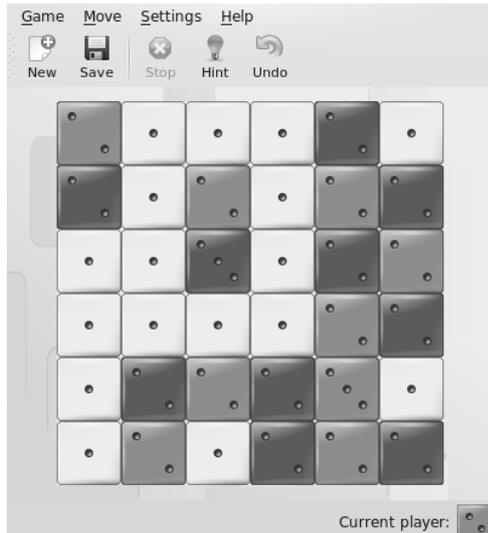
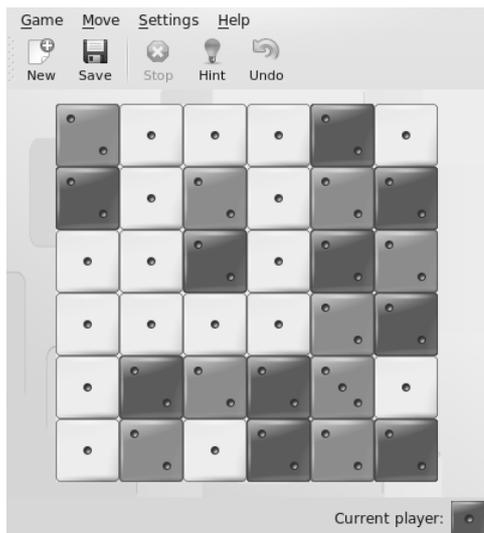
輸入檔: pb.in / 輸出檔: pb.out

Linux 裡有很多的內建小遊戲，大部份都以益智遊戲作為取向，例如 3D 數獨 (Ksudoku)，黑白棋(lagno)，數迴(Loopy)，骰子遊戲(Tali) 等，而 KjumpingCube 則是一個對弈的遊戲。

這個遊戲的規則是這樣的：一剛開始，每個格子都會放有一顆珠珠，但所有格子皆不屬於任何一人。接著，兩人輪流放一顆珠珠，放珠珠的位置只能是不屬於任何人的格子，或者是擁有權為自己的格子。如果這個格子本來是不屬於任何人的，那麼當你選擇要將珠珠放入該格子後，格子所有權就屬於你的。

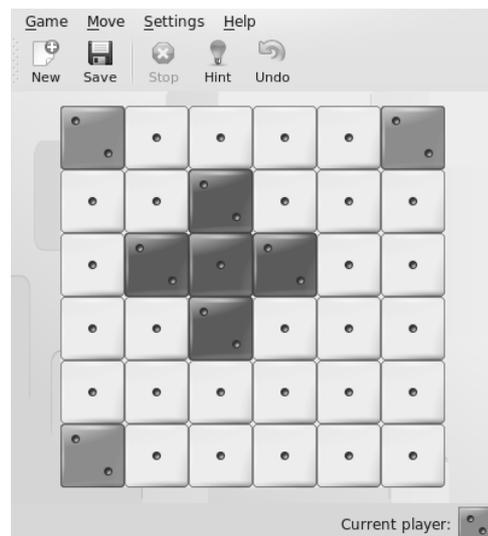
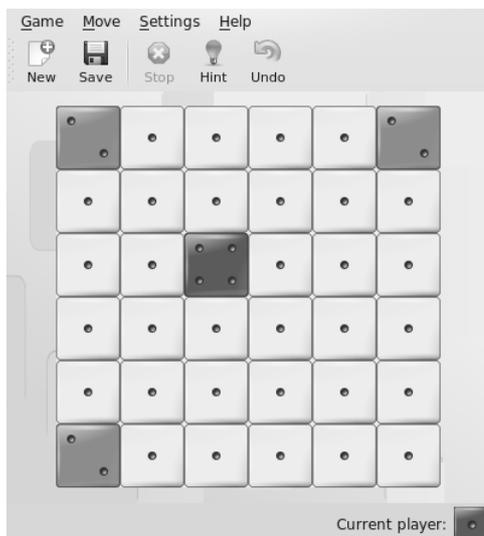


↑ player 1 在第 3 列第 3 行放珠珠



↑ player 1 在第 3 列第 3 行放珠珠

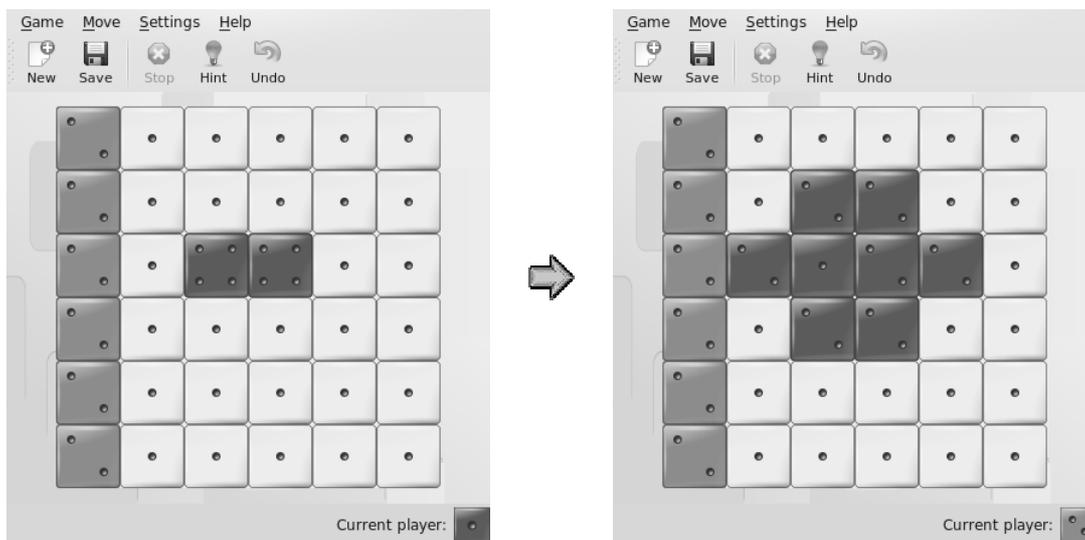
假若一個玩家將一個格子的珠珠放滿出來，則會進行擴張的動作。所謂滿出來的意思，是珠珠的個數超過自己的鄰居數，在這裡我們定義的鄰居一個格子至多只有上下左右四個，舉個例子，在角落的格子只會有兩個鄰居，因此當玩家在空格放珠珠後，使數量變成 3 個，那麼將會進行擴張。而擴張的方法如下，是將滿出來的那一格的珠珠各分一顆至他的附近鄰居，並且周圍的格子的擁有人將同時會變成該玩家，就算附近的格子可能已經屬對手的。讓格子擴張出去，是這個遊戲的重點，兩個人互相攻擊，互相將格子納入自己的勢力。以下是另一個擴張的例子。



↑ player 1 在第 3 列第 3 行放珠珠

也許你接著會有個問題，萬一某個格子珠珠滿出來而擴張，丟一顆珠珠到旁邊的格子後也使旁邊格子的珠珠滿出來會發生什麼事？答案是繼續擴張。一個好的玩

家，會利用這一點做大規模的攻擊，將大部分的格子吃掉；也有人利用這點，一舉把別人殲滅。



↑ player 1 在第 3 列第 4 行放珠珠

這個遊戲的目的，就是讓對手無法再做任何動作，換句話說，就是你要佔有所有格子。

需要注意的一件事情，當遊戲「全部都被佔領」時，就直接結束。換句話說，在連鎖擴張的過程假若已經分出勝負，則不繼續進行擴張的動作，因為已經沒有意義了。

## ■ 輸入檔說明

測試資料包含多組測資，每組測資的第一行包含了 3 個數字， $R C (2 \leq R, C \leq 10) T (1 \leq T)$ ， $R$  代表著這個棋盤幾行， $C$  代表著這個棋盤幾列，而  $T$  代表著兩位玩家共進行了幾個回合（一個玩家放一顆珠珠就算一個回合）。接下來會有  $T$  行，每行有兩個數字  $r c$ ，代表每個回合該玩家將珠珠放在第  $r$  列第  $c$  行。請注意我們是以 1 開始計數，因此左上角為第 1 列第 1 行。為了方便標示，我們假設第一位玩家(先手)為綠色，第二位玩家(後手)為藍色。

$R=C=T=0$  代表測試資料結束。

我們保證所有測資一定合法，也就是每個玩家不會放珠子在別人的格子上，也不會遊戲判出勝負卻還繼續進行。

## ■ 輸出檔說明

對於每組測資，請輸出兩個數字及遊戲狀態，第一及第二個數字代表著目前綠色及藍色的格子個數有幾個，而狀態可能有三個：“GREEN”代表遊戲已終局，且是綠色勝出；“BLUE”代表遊戲已終局，且是藍色勝出；“processing”代表遊戲還未結束。

## ■ 範例輸入

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

## ■ 範例輸出

```
1 0 processing
1 1 processing
3 1 processing
0 4 BLUE
```

## 題目 C 跑跑卡丁車

輸入檔: [pc.in](#) / 輸出檔: [pc.out](#)



你玩過跑跑卡丁車嗎？這是款遊戲橘子代理的線上賽車遊戲，分成道具賽跟競速賽兩種模式，道具賽中可以丟水球、香蕉皮等等來擾亂對手，而競速賽則是純粹的比誰快！

但你知道卡丁車是真的存在的東西嗎？



卡丁車是賽車界最初級的賽事，所有有名的車手小時候都是從這項賽事中開始學習的，有名的 F1 世界冠軍舒馬赫退休後甚至重溫童年時光以參加卡丁車賽事為樂。

可別小看小小的一台車，輕量化的車身加上強力引擎的搭配，能夠輕鬆跑出百公里的時速，刺激度不輸一般賽車，能否良好的駕馭這小怪獸也是對車手的考驗。

國內的卡丁車賽事通常分為測時賽、複賽 1、複賽 2、決賽四個階段，測時賽不需同時出發，每個車手必須在有限的時間內盡量達到最快的單圈時間，測時賽的排名決定複賽 1 的起跑位置，從複賽 1 開始則都是同時起跑，複賽 1 的排名決定複賽 2 的起跑位置，複賽 2 的排名決定決賽的起跑位置。

根據研究發現，如果能在測時賽中拿到前 1/3 的排名，則贏得比賽的機率會大增，身為 NPSC 車隊技術員的你，找出測時賽前 1/3 的車手提供給車隊研究吧！

假設有 N 個車手，前 1/3 的定義是排名前(N/3, 無條件捨去)的車手，例外條件是當有未列入前 1/3 的車手成績與第 N/3 名的車手成績相同的話，也必須將他算入前 1/3

#### ■ 輸入檔說明

測資會有多組，每組測資的第一行是一個整數 N， $3 \leq N \leq 1000$ ，當 N=0 的時候表示測資結束，接下來的 N 行分別是車手的姓名及測時賽的最佳單圈成績，姓名只會包含大小寫英文字母，成績的格式為 HH:MM:SS.SSS(小數點後可能有 0 到 3 個位數)，秒數精確到三位數，姓名與成績用一個空白分隔。

#### ■ 輸出檔說明

對每組測資，請先輸出一行 LIST START，接下來輸出符合條件的車手名稱，輸出順序請依照出現在測資的順序，車手輸出完畢後請輸出一行 LIST END。

#### ■ 範例輸入

```
3
Schumacher 00:01:46.532
Alonso 00:01:47.581
DE 00:01:46.531
3
Schumacher 00:01:00
Alonso 00:01:00
DE 00:01:00
0
```

## ■ 範例輸出

```
LIST START
DE
LIST END
LIST START
Schumacher
Alonso
DE
LIST END
```

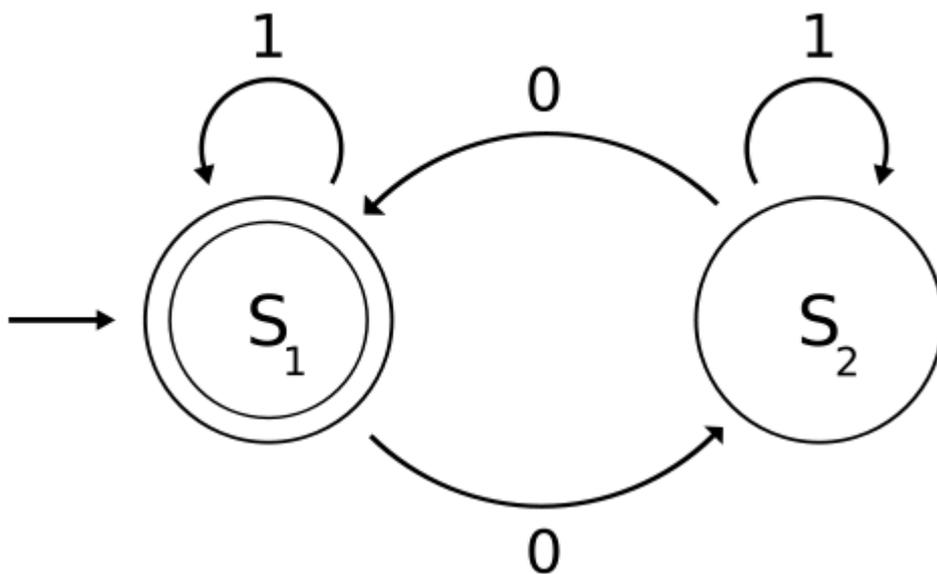
## 題目 D

### 有限狀態自動機

輸入檔: pd.in / 輸出檔: pd.out

有限狀態自動機 (Finite State Machine) 是由有限個狀態以及在這些狀態之間的轉移和動作等行為所組成的數學模型。

有限狀態自動機可以使用狀態轉移圖 (State Transition Diagram) 來表示, 例如下面的狀態轉移圖, 表示的是一個可以用來判斷二進位數是否具有偶數個 0 的有限狀態自動機。「沒有起點的箭頭」指向狀態  $S_1$ , 表示狀態  $S_1$  是開始狀態 (Start State); 狀態  $S_1$  為雙重圓圈, 表示狀態  $S_1$  是接受狀態 (Accept State)。



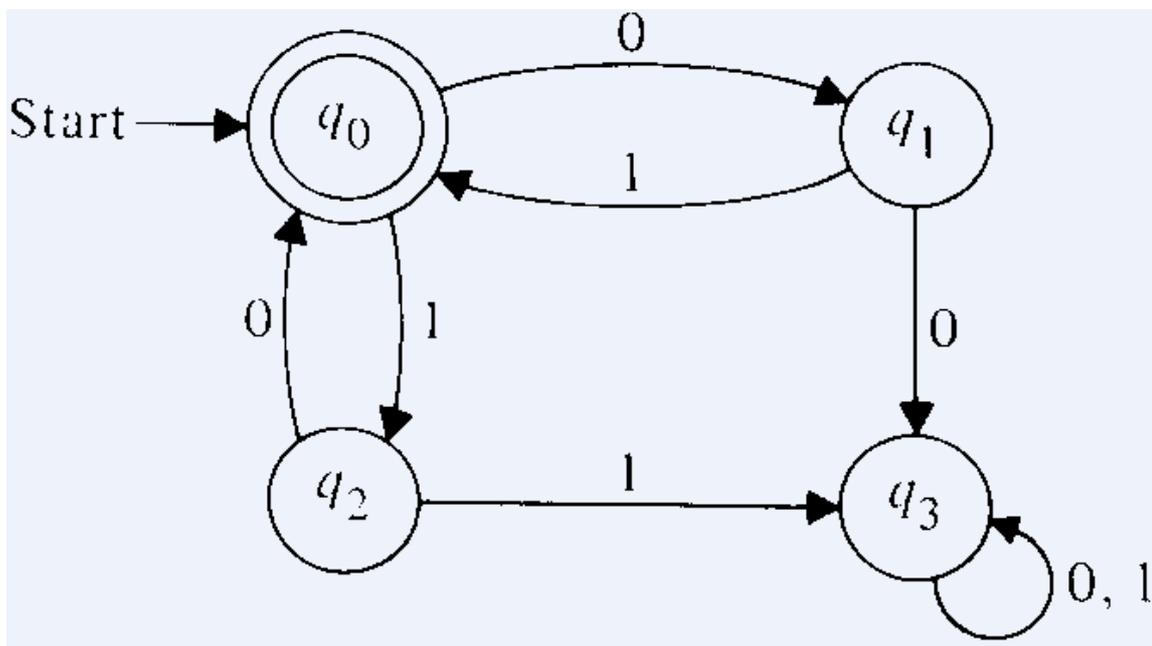
輸入二進位數, 例如 **10101**, 一開始的狀態是狀態  $S_1$ ; 讀到第一個 **1** 的時候, 依然停留在狀態  $S_1$ ; 讀到第一個 **0** 的時候, 移動到狀態  $S_2$ ; 讀到第二個 **1** 的時候, 依然停留在狀態  $S_2$ ; 讀到第二個 **0** 的時候, 移動到狀態  $S_1$ ; 讀到第三個 **1** 的時候, 依然停留在狀態  $S_1$ ; 因為狀態  $S_1$  是接受狀態, 表示輸入的二進位數 **10101** 具有偶數個 **0**。

如果輸入的二進位數是 **10010**, 最後會停留在狀態  $S_2$ 。因為狀態  $S_2$  不是接受狀態, 所以我們可以知道二進位數 **10010** 不具有偶數個 **0**。

有限狀態自動機也可以使用狀態轉移表（State Transition Table）來表示，例如上面(上一頁)的狀態轉移圖可以表示為：

當前狀態 → 條件 ↓	狀態 $S_1$	狀態 $S_2$
讀到 0	狀態 $S_2$	狀態 $S_1$
讀到 1	狀態 $S_1$	狀態 $S_2$

你現在的任務是模擬下圖表示的有限狀態自動機（如果你有興趣的話，你也可以想一想這個有限狀態自動機的目標是找出怎樣的二進位數）：



#### ■ 輸入檔說明

第一行有一個整數  $N$  ( $1 \leq N \leq 100$ )，代表總共有  $N$  筆測試資料。

每一筆測試資料各佔一行，各為一個二進位數  $B$ ，位數最少為 1、最多為 100。請不要忽視開頭的 0（不論有幾個），舉例而言：二進位數 1 和二進位數 001 在此題目中是不相同的。

#### ■ 輸出檔說明

針對每一筆測試資料，輸出最後會停留在哪一個狀態，如果停留在接受狀態就輸出 Accept，否則輸出 Reject，格式請參照範例輸出。

## ■ 範例輸入

```
3
101011
0110
1011011
```

## ■ 範例輸出

```
q3: Reject
q0: Accept
q3: Reject
```

## 題目 E 糾結的水管

輸入檔: pe.in / 輸出檔: pe.out

正直 DE 是個很正直的青年，每天他都會到鄰近的大街小巷中幫老爺爺老奶奶整理院子和花圃。

有天他正要澆花時發現李爺爺院子裡的好多條塑膠水管全部都打結成一團了，怎麼解都解不開。好在這些水管還是能通，只要找出某一頭的另一端在哪就可以使用。

```

.....
.A--+. ....
....|.B+..
+---#--#-C
|...|..|..
+-D.|..|..
.+--+..|..
.|.E---#-F
.+---G.|..
.....+-H

```

上圖代表多條糾結的水管，其中：

‘.’代表這個位置是空的，沒有水管

‘A’~‘H’代表水管的端點，也就是出入口

‘-’和 ‘|’各代表橫向和縱向的水管

‘+’代表水管的轉彎處

‘#’代表在這個點上有兩條水管交叉，兩條水管之間不會互相影響

若我們從 A 注水進去，水會從另一端 G 出來；若從 F 端注水則水會從 E 端出來。

因為 DE 實在是太正直了，面對這種糾纏不清的問題反而不知道怎麼解決，所以他現在要脅你在五個小時以內給他一個程式解決這個問題：問你從某個端點注水進去，水會從哪個端點出來？

## ■ 輸入檔說明

第一行有一個整數  $T(T \leq 20)$ ，代表總共有幾筆測試資料。

每筆測試資料的第一行包含  $N(0 < N \leq 20), M(0 < M \leq 20)$

接下來  $N$  行，每行有  $M$  個字元，描述水管的糾纏狀況。

接下來會有一個數字  $Q(1 \leq Q \leq 20)$ ，代表問題的數量。

接下來  $Q$  行每行包含一個字元(在 A 到 Z 之間)，代表要從這個端點注水進去。

你可以假設：

不會有兩條同方向且重疊的水管，也不會在端點上交叉

不會有兩條水管在同一個點上轉彎

兩條不同水管的轉彎點不會相鄰

一個端點的上下左右必定剛好只有一個管子和它相連

水管除了端點以外最少有 1 的長度

每兩筆測試資料之間會有一個空白行。

## ■ 輸出檔說明

對於每個問題，輸出一行包含：

注水端點 -> 出水端點

端點名稱同 input 描述。

每組測試資料間要有一個空白行。

## ■ 範例輸入

```

2
10 10
.....
.A--+. ....
...|.B+..
+---#--#-C
|...|..|..
+-D.|..|..
.+--+..|..
.|.E---#-F
.+---G.|..
.....+-H
2
A
C

3 3
...
A-B
...
1
A

```

■ 範例輸出

```
A->G
```

```
C->D
```

```
A->B
```

## 題目 F

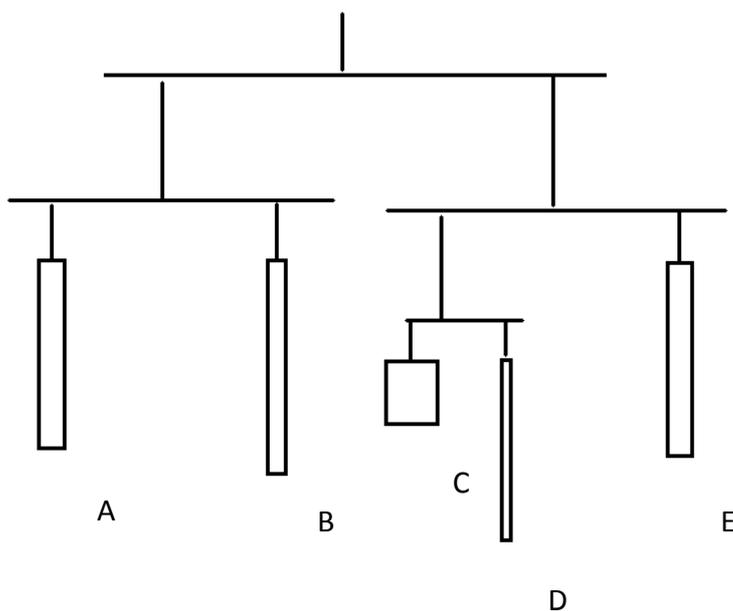
### 風鈴

輸入檔: pf.in / 輸出檔: pf.out

風鈴，一種鈴。其可在風的吹動下，通過各種鈴鐺或其它物體的碰撞來發出聲音。

阿啾大師製造的風鈴是由線、木棒、銅管構成。

一個表示為((A,B),(C,D),E)的風鈴，畫成圖後就像下圖所示：



一個好的風鈴，在每一個懸吊點都要保證是平衡的，也就是不能有左邊比較重或是右邊比較重的情況。以上圖為例，假設五個銅管的重量分別為 ABCDE，則：

$$A=B, C=D, C+D=E, A+B=C+D+E$$

反之，只要有一個懸掛點左右兩邊重量不均衡，這個風鈴就是壞掉的。

為了精確校正風鈴的音色，阿啾大師使用了沒有重量的木棒和線，所以可以只考慮銅管的重量。現在你得到了好幾個風鈴，請你驗證它們到底是好的還是壞的。

#### ■ 輸入檔說明

第一行有一個整數 N，代表總共有幾組測試資料。

接下來共有 N 行，每一行有一個字串描述風鈴的形狀(長度不超過 100)。

描述方式如下：

每對括號都代表一根木棒，左右兩邊各懸吊一根木棒或是一根銅管。  
每個數字都代表一根銅管，數字代表這個銅管的重量，數字的範圍在 0 到 100 之間(包含 0 及 100)。

舉例來說，(10,20)意思是這根木棒左邊懸吊了重量為 10 的銅管，右邊懸吊了重量為 20 的銅管，而整個木棒+銅管的重量為 30。

#### ■ 輸出檔說明

對每組測試資料輸出一行，若風鈴是好的則輸出 Yes，否則輸出 No。

#### ■ 範例輸入

```
3
((1,1),(1,1))
(3,2)
(((1,1),2),4),8)
```

#### ■ 範例輸出

```
Yes
No
Yes
```

## 題目 G 整理 CD

輸入檔: pg.in / 輸出檔: pg.out

彼得很喜歡聽音樂，因此他蒐集了  $M$  張 CD。有一天他心血來潮，打算把所有的 CD 放在一些架子上。

由於架子的規格有限，每一個架子上面最多只能擺放  $K$  張 CD。不過彼得不是很喜歡數字  $N$ ，因此他很不希望任何架子上的 CD 張數是  $N$  的倍數，當然，這也就意味著不能有空的架子出現。此外，由於彼得實在不太喜歡數字  $N$ ，因此擺放 CD 的架子數量也不能是  $N$  的倍數。

請問彼得至少要準備多少個架子才能把所有的 CD 放到架子上呢？舉個例子來說，如果現在彼得有 11 張 CD ( $M=11$ )，每個架子上面至多可以擺放 6 張 CD ( $K=6$ )，彼得很不喜歡數字 3 ( $N=3$ )。那麼至少要 4 個架子才能夠放得下彼得的所有 CD。因為 11 可以表示成  $4 + 4 + 2 + 1$ 。

### ■ 輸入檔說明

輸入檔的第一行有一個正整數  $T$ ，代表測試資料的組數。

每一組測試資料佔一行依序包含三個整數  $N, M, K$  ( $1 \leq N, M, K \leq 10^9$ )。

### ■ 輸出檔說明

對於每一筆測試資料，請輸出彼得必須準備的最少架子數。

如果無論如何都沒有辦法把所有的 CD 都放到架子上，這實在是太重要了，請輸出 **IMPORTANT** 提醒彼得。

### ■ 範例輸入

```
2
3 11 6
2 12 47
```

### ■ 範例輸出

```
4
IMPORTANT
```