

## 2011 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 20 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**)。
- 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動作、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。

表 1: 題目資訊

	題目名稱	執行時間限制
題目A	旅行地圖	5 秒
題目B	積木封裝	5 秒
題目C	摩斯電碼	5 秒
題目D	巨集危機	5 秒
題目E	神雕大俠的困境	5 秒
題目F	英雄 Fantasy for you	30 秒
題目G	請客的藝術	10 秒

## 2011 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;
}
```

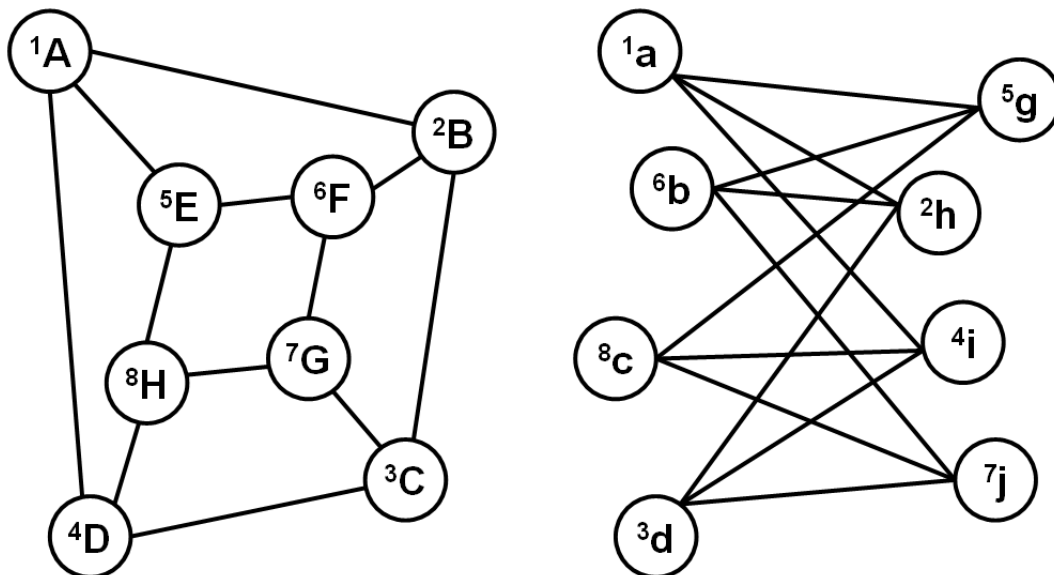
## 題目 A 旅行地圖

執行時間限制: 5 秒

規規和散散是感情十分要好的兄弟，他們兩個人都很熱愛旅行，所以常常一起出去旅遊。

最近他們的叔叔在整理規規和散散之前一起出去玩時寫下的筆記，發現規規在記錄事情之前總會先清楚地標明時間和地點；散散每次出門旅遊都會更換一本新的記事本，也不會順手記下日期，加上散散喜歡用「感覺」記錄一切，總是使用「天空很大的地方」或「海水很藍的沙灘」這一類的詞來代替地名，所以叔叔搞不清楚規規和散散的日記要如何對應起來。

在叔叔觀察了很久之後，發現規規和散散兩個人都有記錄大眾運輸工具路線圖的習慣，規規會根據地點之間的遠近和相對位置作圖，並且記錄地點之間是否有大眾運輸工具可直達；散散畫的地圖只在乎地點之間能否直達，加上散散獨家的「地名記錄法」，使得叔叔還是沒辦法迅速地把兩兄弟畫的地圖對應起來。（如果我們可以對兩張地圖上的地點找到一組一對一的配對，讓兩張地圖上的地點之間的直達關係都相同，我們就當作這兩張地圖能夠對應；對於下圖而言，節點上的數字即代表其中一組配對。）



聰明的你，能不能夠幫助叔叔把規規和散散畫的地圖對應起來呢？

## ■ 輸入檔說明

輸入的第一行有一個正整數  $T$ ，代表測試資料的組數 ( $1 \leq T \leq 100$ )。

每組測試資料的第一列有兩個整數  $N$  和  $M$ ，代表地圖內有  $N$  個地點 ( $1 \leq N \leq 8$ ) 和  $M$  個關係，關係不會重複也沒有方向性。

接下來的  $M$  行描述規規畫的地圖，地名皆由一個大寫字母所代表，每一行包含兩個地名，表示這兩個地點之間可以直達；再下來的  $M$  行描述散散畫的地圖，地名皆由一個小寫字母所代表。

## ■ 輸出檔說明

對於每組測試資料輸出兩個地圖能否對應，能就輸出 yes，否則輸出 no。

## ■ 範例輸入

```
3
8 12
A B
A D
A E
B C
B F
C D
C G
D H
E F
E H
F G
G H
a g
a h
a i
b g
b h
b j
c g
```

```
c i
c j
d h
d i
d j
4 3
A B
B C
C D
a b
a c
a d
3 1
A E
c d
```

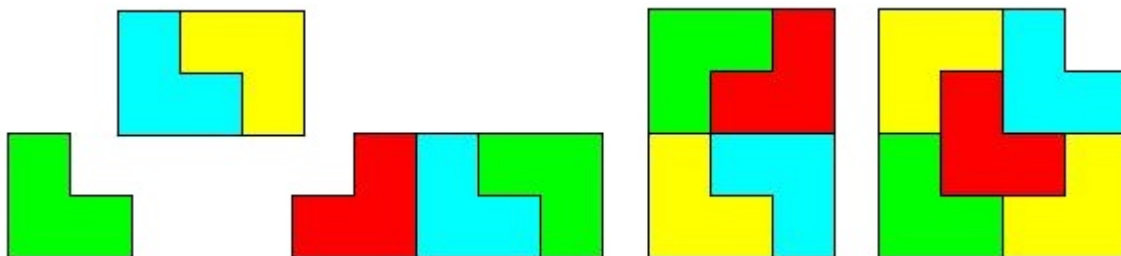
### ■ 範例輸出

```
yes
no
yes
```

本頁留白。

## 題目 B 積木封裝

執行時間限制: 5 秒



阿卡在一間玩具工廠擔任開發的工作，主要內容是具有數學意義的玩具。最近阿卡正熱衷於積木的設計，為此他也去買了許多不同工廠販賣的積木，但是他發現找不到非凸多邊形形狀的積木，這令他覺得非常可惜。

阿卡認為讓小朋友組合積木對認識各種形狀的性質很有幫助，由於 L 形在非凸多邊形中架構相對簡單並且容易組合，因此設計了一款由 L 形積木所構成的玩具，L 形積木是由邊長為 2 單位的正方形積木切掉角落的 1 單位正方形而成，每份玩具都包含了相同塊數的積木。

阿卡向上層提交了新玩具的設計方案之後，上層回覆他希望能將新玩具依據積木數量分成多種款式，以配合即將到來的耶誕優惠活動，並且儘可能地降低單份玩具的製作成本。阿卡一直以來對降低成本這件事感到十分不滿，因為他覺得應該要給小朋友最好的玩具，不過他為了明天還能吃到工廠裡提供的免費點心也只能努力滿足上層的要求。

過了幾天，阿卡決定把新玩具的封裝盒尺寸縮減成恰好能將積木裝進去以完成上層的要求，可是他最近正忙於參加 NPSC (Non-Professional Snack Contest: 業餘點心大賽)，沒有時間去計算各種數量的積木所需要的封裝盒尺寸，請你寫一個程式幫助阿卡計算當有  $N$  塊積木時封裝盒所需要的最小周長。

爲了美觀考量，封裝盒的形狀限定爲矩形，且積木只能以 L 形面朝上平放，不可以豎立。

## ■ 輸入檔說明

輸入的第一行有一個正整數  $T$ ，代表測試資料的組數 ( $1 \leq T \leq 100$ )。

每組測試資料只有一行包含一個數字  $N$ 。 ( $1 \leq N \leq 10^8$ )

## ■ 輸出檔說明

對每筆測試資料輸出一個整數，代表封裝盒所需要的最小周長。

## ■ 範例輸入

```
3
1
3
5
```

## ■ 範例輸出

```
8
14
16
```



## 題目 C

# 摩斯電碼

執行時間限制: 5 秒

摩斯電碼是一種傳送文字訊息的方式，在電報通訊的時代被廣泛地使用。它使用兩種長短不同的訊號產生而成的序列來表示各個英文字母、數字和標點符號。我們用點（.）來表示短訊號，用劃（—）來表示長訊號，在兩個訊號間以很短的空白（無訊號的狀態）來隔開。英文字母對應的摩斯電碼如下：

A .—	B —...	C —.—.	D —..	E .	F ..—.
G —.—.	H ....	I ..	J .— — —	K —.—	L .— ..
M — —	N —.	O — — —	P .— —.	Q — — .—	R .— .
S ...	T —	U ..—	V ...—	W .— —	X —..—
Y —.— —	Z — — ..				

在摩斯電碼中，我們需要使用很短的空白來分隔兩個訊號，用稍長的空白來分隔兩個字母，用更長的空白來分隔兩個英文單字。但是這樣有些不方便，如果收訊號的人沒辦法精準的判斷空白的長短，就會搞混 EI 和 S，或是 AN 和 P。這個問題的原因是 S 的電碼開頭和 E 的電碼重複，P 的電碼開頭和 A 的電碼重複。為了解決這個問題，我們修改摩斯電碼的編碼規則，讓每一個字母的電碼都不會和其他字母的電碼開頭重複，我們稱為「字首差異電碼」。使用字首差異電碼，我們不需要空白就可以分辨兩個字母。這個電碼與英文字母的對應如下：

A — — —.	B — — ....	C .— ..—	D — — — — —
E .— —	F ..— .—	G — — ..— —	H —...
I —.— —	J ..— ..— .— —	K ..— ..— —	L — — — — .
M ..— — —	N —.—.	O — — .—	P — — ...—
Q ..— ..— ..—	R .— .—	S —..—	T ...
U .— ...	V ..— ...	W ..— —.	X ..— ..— .— .
Y — — ..— .	Z ..— ..— ...		

現在，給你一段字首差異電碼，請你解讀這段電碼代表的英文字句。

## ■ 輸入檔說明

輸入的第一行有一個正整數  $T$ ，代表測試資料的組數 ( $1 \leq T \leq 500$ )。

接下來每一行有一個以點（‘.’’）、劃（‘-’）和空白（‘ ’）組成的字串，代表一組測試資料。點和劃就如題目所說，分別代表短訊號和長訊號。我們忽略訊號之間的空白，而用空白字元代表兩個單字的分隔。題目保證每一行的長度不超過 200，而且都能正確解碼，沒有多餘的訊號。一個字母的電碼不會被空白字元切開，測試資料中也不會有連續的空白字元。

## ■ 輸出檔說明

對每一筆測資輸出一行，即這段字首差異電碼代表的英文字句。電碼對應的英文字母以大寫表示，兩個英文單字間以一個空白字元隔開。

## ■ 範例輸入

[illegible]

## ■ 範例輸出

THIS IS AN  
EASY  
PROBLEM  
YOU CAN DO IT

## 題目 D

# 巨集危機

執行時間限制: 5 秒

相信大家都知道 C/C++ 語言有提供巨集(macro)的功能，我們可以藉由 `#define` 來實作之，寫出各式各樣五花八門的巨集。若不知道也沒有關係，我們可以參考以下的程式碼：

```
#define SQUARE(x) ((x) * (x))  
#define CUBE(x)    ((x) * (x) * (x))
```

有了這兩行巨集宣告，我們就可以在往後的程式碼中書寫如以下的述句，就如同我們宣告了兩個名叫 `SQUARE` 和 `CUBE` 的函式(function)一樣：

- 對於 C 語言：

```
scanf("%d", &x);  
printf("%d %d\n", SQUARE(x), CUBE(x));
```

- 對於 C++ 語言：

```
cin >> x;  
cout << SQUARE(x) << " " << CUBE(x) << "\n";
```

其概念為，在編譯時我們便會將所書寫的巨集代入程式碼中，你可以當作是一種置換(replace)的動作，置換之後的程式碼大概會長的像下面這樣：

- 對於 C 語言：

```
scanf("%d", &x);  
printf("%d %d\n", ((x) * (x)), ((x) * (x) * (x)));
```

- 對於 C++ 語言：

```
cin >> x;  
cout << ((x) * (x)) << " " << ((x) * (x) * (x)) << "\n";
```

然而，書寫巨集需要非常小心，一旦有一點疏忽，很容易就會造成錯誤的結果。舉例來說，如果我們因為一時偷懶，將以上的程式碼寫成：

```
#define SQUARE(x) (x * x)
```

則我們將會得到錯誤的結果。為什麼呢？因為假設我們的  $x$  為  $5 + 7$ ，則我們會認為  $SQUARE(5 + 7)$  的結果應該為  $(5 + 7)^2 = 144$ ；可是，如果我們因為一時的疏失而少寫了一些括號，則結果將會是出乎意料的(如果你不知道原因的話)：你會得到 47。原因在於，我們在展開巨集的過程中，編譯器所做的動作是「直接代換  $x$  的值」。就剛剛上面錯誤的例子會是：

- 對於 C 語言：

```
printf("%d\n", SQUARE(5 + 7));  
=> printf("%d\n", (5 + 7 * 5 + 7));
```

- 對於 C++ 語言：

```
cout << SQUARE(5 + 7) << "\n";  
=> cout << (5 + 7 * 5 + 7) << "\n";
```

發現到什麼問題了嗎？

爲了避免這種錯誤的發生，有些人會開始瘋狂的加括號，每一步都加括號！當然，這樣並沒有不好，某方面來說這樣的編程習慣或許還是比較好的。但是，達克皮(Darkpi)很不喜歡有一堆括號的程式碼，他覺得只要答案對就好了何必要那麼多的括號？因此，他想要問說，對於一個巨集運算式，在合理(不影響正確性)的情況下，最多可以拿掉幾對括號？

我們保證對於所有輸入的巨集其名稱([MACRO\_NAME])都只包含英文大寫字母和底線 `_`，並且長度不超過 50。而對於所有輸入的巨集其運算式([OPERATION])，都只包含變數 `x`、左括號 `(`、右括號 `)`、加法二元運算子 `+` 和乘法二元運算子 `*`，並且長度不超過 40。爲了進一步的簡化題目，我們假設輸入的 `x` 都是一個只包含加(+)和乘(\*)的整數運算式(例如：`2 + 5 * 3 + 1`)。意思是你只要保證對於所有 `+` 和 `*` 運算子組合出來的算式都能得出**正確**的答案，那麼就是一組合法的方案。所謂的「正確的」定義爲：假設你把 [OPERATION] 中的幾組括號拿掉了，得到一個新的式子 [NEW\_OP]，則下面兩個函式所回傳的結果在無溢位狀態發生時必須是一樣的。

```
int [MACRO_NAME](int x) { return [NEW_OP]; }
int [MACRO_NAME](int x) { return [OPERATION]; }
```

## ■ 輸入檔說明

輸入的第一行有一個正整數  $T$ ，代表測試資料的組數 ( $1 \leq T \leq 100$ )。

之後緊接著有  $T$  行(因此整個輸入檔案將有  $T + 1$  行)，每一行都恰好有一個已經寫好的巨集。

我們保證原本給定的巨集都是「安全的」，只是可能有多餘的不必要之括號。輸入的巨集程式碼格式如下：

```
#define [MACRO_NAME](x) [OPERATION]
```

如果在 [OPERATION] 中有出現 `+` 或 `*` 等運算子，則其前後必定會有且僅有各一個空白。

上面說所說的「安全的」是指，所給定的巨集會跟下面的函式有相同的效果。(在題目對 `x` 的條件下)

```
int [MACRO_NAME](int x) { return [OPERATION]; }
```

## ■ 輸出檔說明

對於每一個輸入的巨集(共  $T$  組(行))，請輸出一個整數代表最多可以拿掉幾對括號而仍保持答案的正確性。

重要的是，我們規定無論你是怎麼消括號的，**最外層的括號一定要保留**，不可以刪除。

## ■ 範例輸入

```
4
#define SQUARE(x) (((x) * (x)))
#define PLUS_TWO(x) ((x) + (x))
#define ADD_THREE_TIMES(x) (x + (x + x))
#define CANNOT_REMOVE(x) ((x) * (x + x))
```

## ■ 範例輸出

```
2
2
1
0
```

## 題目 E

# 神雕大俠的困境

執行時間限制: 5 秒

在襄陽抵禦蒙古軍一戰，『神雕大俠』楊過苦戰金輪國師。雙方激戰了數千餘招，修練了數十年的真力激盪在戰場上。最後，楊過終於使出畢生絕技「黯然銷魂掌」。

在出掌的那一剎那，楊過頓感真氣微塞，心中暗叫不妙，尤其在這種驚險時刻，勝負生死多在一招之間。情急之下楊過奮力擊出全身的內力，終於打敗金輪國師。

待戰事結束，大俠郭靖隨即為楊過把脈。這脈不把還好，一把之下郭靖猛然驚覺：楊過因為長年在外與人拼鬥，身上早就為許多奇異內功所傷；又因楊過喜好行俠仗義，鮮少安靜下來靜心調養，多處瘀血阻塞了真氣。若不儘速治好，少則內功修為停滯不前，多則可能減壽數年。所幸郭靖征戰多年，熟知行醫之道，連忙找來襄陽城中最好的畫師為楊過臨摹出一幅「五臟六腑圖」。根據黃藥師長年的研究，只要依據這張「五臟六腑圖」從體外施以猛烈的掌擊即可清除經脈內的瘀血。

如下圖A即為「五臟六腑圖」，其中 A 和 B 代表兩個不同的臟腑，而 \* 則代表瘀血所在之處。郭靖可從上方或左方的任一處以垂直或水平方向施以「降龍十八掌」，則可清除掌力觸及的瘀血(如圖B)，同時，若掌力觸及臟腑則會令之受傷。

			A	A	
			*	A	*
				A	
				*	
	B	B	B	*	
*					

圖A

			A	A	
			*	A	*
				A	
				*	
	B	B	B	*	
*					

圖B

爲了避免造成過大的副作用，郭靖決定在不傷及太多臟腑的前提下清除所有經脈中的瘀血，他是否能成功呢？

## ■ 輸入檔說明

輸入的第一行有一個正整數  $T$ ，代表測試資料的組數 ( $1 \leq T \leq 100$ )。

每組測試資料以三個整數  $N, M, K$  開頭，其中  $N$  代表「五臟六腑圖」的大小， $M$  代表臟腑的數量， $K$  代表最多可以傷及幾個臟腑。  
( $1 \leq N \leq 20, 0 \leq M \leq 10, K \leq M$ )

接下來  $N$  行爲楊過的「五臟六腑圖」，其中臟腑只會由大寫英文字母表示，順序是由 'A' 到 'Z'。

## ■ 輸出檔說明

對於每筆輸入資料，如果郭靖能在最多傷及  $K$  個臟腑的狀況下清除所有瘀血則輸出 yes，否則輸出 no。

## ■ 範例輸入

```
2
5 2 1
....*
*AA..
*A**
...B
..*BB
3 1 0
AAA
A*A
AAA
```

## ■ 範例輸出

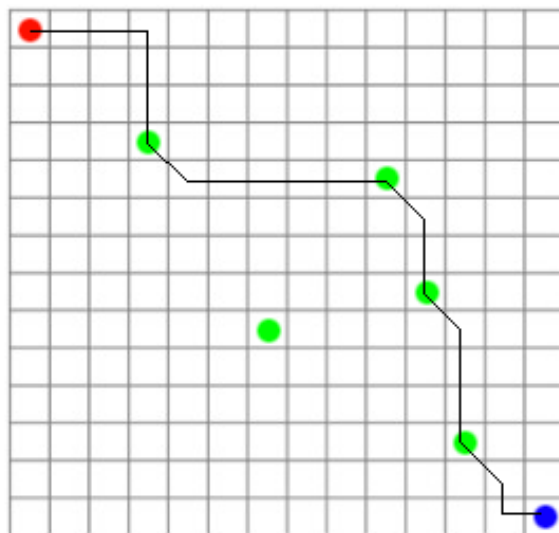
```
yes
no
```



## 題目 F

# 英雄 Fantasy for you

執行時間限制: 30 秒



英雄是一款線上遊戲，在這款遊戲中的地圖是以  $N \times N$  的格子方式呈現。在一般的情況下，每次玩家只能往上、下、左、右四個方向移動一格。但是，在某些格子之中備有一些傳送門（如圖中有綠色點的格子），這個傳送門讓你也可以選擇要往左上、左下、右下和右上移動一格，也就是可以前往周圍八格。

現在你的公會被攻打了，你必須要趕到你的公會去幫忙，然而你在這個地圖的左上角（如圖中有紅色點的格子），而公會卻在右下角（如圖中有藍色點的格子）。幸運的是，你有所有傳送門的座標，你希望能夠妥善的利用傳送門讓你縮短到公會的時間。每次從一格移動到另一格則時間加一，你希望能夠用最短的時間到公會的位置。

## ■ 輸入檔說明

輸入的第一行有一個正整數  $T$ ，代表測試資料的組數 ( $1 \leq T \leq 100$ )。

每一組測試資料的第一行有一個整數  $N$  ( $1 \leq N \leq 10000000$ )。第二行有一個數字  $K$  ( $0 \leq K \leq 50000$ )，代表總共有幾個傳送門。第三行有  $K$  個數字  $X_1, X_2, \dots, X_K$  ( $1 \leq X_i \leq N$ )，代表這  $K$  個傳送門的  $x$  座標。第四行有  $K$  個數字  $Y_1, Y_2, \dots, Y_K$  ( $1 \leq Y_i \leq N$ )，代表這  $K$  個傳送門的  $y$  座標。座標從左上角開始算，左上角為  $(1, 1)$ 。

## ■ 輸出檔說明

對每筆測試資料輸出一個整數，代表你要到公會所需的最短時間。

第一筆測資最短的走法如上圖。

第二筆測資最短的走法為  $(1, 1) \rightarrow (2, 1) \rightarrow (3, 2) \rightarrow (4, 3) \rightarrow (4, 4)$ 。

## ■ 範例輸入

```
2
14
5
4 9 5 8 12
4 7 10 11 12
4
3
2 2 3
2 1 2
```

## ■ 範例輸出

```
22
4
```

## 題目 G 請客的藝術

執行時間限制: 10 秒

三五好友又要一起聚餐了。不過每一次聚餐的時候，大家對於要吃什麼總是沒有辦法達成共識。於是他們決定透過抽牌比大小的遊戲來決定這次的聚餐活動。

他們總共有  $m$  個人，拿了一疊上面分別編有 1 到  $n$  號總共  $n$  張牌。發給大家以後，每一個人至少得到一張牌。接著，每一個人從中選一張牌，蓋在桌上。然後大家一起攤開。他們另外訂了兩個數字  $a$  和  $b$ 。

根據這次聚餐的規模，攤開來牌上數字的大小可以決定每個人要做的事情。牌上數字前  $a$  大的人(也就是數字是最大到第  $a$  大總共  $a$  個人)可以共同決定這次聚餐所在的餐廳。而第  $a + 1$  大到第  $b$  大的人，則是要負責將大家載往聚餐地點。可憐的第  $b + 1$  到最小數字的人，則必須負擔吃飯的所有費用。

現在給你每一個人手上所有的牌。請你幫忙算算，總共有多少種可能的情况會發生呢？我們說所謂兩種不同的情况是指：在這兩種情形中，至少有一人所做的事情(決定餐廳、負責交通或是負責出錢)不同。

### ■ 輸入檔說明

輸入的第一行有一個正整數  $T$ ，代表測試資料的組數 ( $1 \leq T \leq 100$ )。

每一組測試資料的第一行包含四個正整數  $m, n, a, b$  ( $3 \leq m \leq n \leq 100; 1 \leq a < b < m$ ) 表示這團人的人數、牌的總數、以及決定做不同的事情的兩個分界點。接下來有  $n$  個正整數  $a_1, a_2, \dots, a_n$ ，每一個數字都介於 1 和  $m$  之間，其中  $a_i$  表示編號為  $i$  的牌在  $a_i$  這個人的手上。輸入保證每一個人至少會拿到一張牌。

## ■ 輸出檔說明

對於每一筆測試資料請輸出一行，包含一個整數表示有多少種可能的情形。  
你可以假設所有輸入的測試資料答案都不會超過  $10^{18}$ 。

## ■ 範例輸入

```
4
3 3 1 2
1 2 3
3 7 1 2
1 1 1 2 2 3 3
3 9 1 2
1 2 3 1 2 3 1 2 3
15 40 5 10
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
1 3 5 7 9 11 13 15 17 19
```

## ■ 範例輸出

```
1
1
6
113862
```