

# 送分題 – solution

---

(30 分)

## 前言

比賽開始了！

趕快驗證一下，  
網路是否設定正確？  
上傳競賽程式是否順利？  
檔案是否用 `STDIN` 輸入？  
程式解答是否用 `STDOUT` 輸出？

都沒問題，30 分就到手了！ 繼續 ... 衝！衝！衝！

## 問題敘述

試寫一程將輸入的兩個整數相加後輸出。

### 輸入格式

A, B

兩個正整數，中間以空白區隔

### 輸出格式

C

兩個正整數之和

### 資料範圍

A, B, C 皆為小於 9999999 的正整數

### 資料範例

#### 輸入範例 1

2756 707

#### 輸出範例 1

3463

#### 輸入範例 2

23024 19109

#### 輸出範例 2

42133

### 輸入範例 3

6253 14484

### 輸出範例 3

20737

#### 範例說明：

兩個整數 2756 及 707 相加後輸出 3463

#### 補充說明：

以下為 STDIN /STDOUT 的方式輸入輸出的 Sample Code,

#### [Sample Code: C]

```
#include <stdio.h>

int main() {
    int a, b;
    scanf("%d %d", &a, &b);
    printf("%d\n", a+b);
    return 0;
}
```

#### [Sample Code: C++]

```
#include <stdio.h>

int main() {
    int a, b;
    scanf("%d %d", &a, &b);
    printf("%d\n", a+b);
    return 0;
}
```

#### [Sample Code: Java]

```
import java.util.Scanner;

public class solution {
    public static void main(String args[]) {
```



```
Scanner in = new Scanner(System.in);
int a, b;
a = in.nextInt();
b = in.nextInt();
System.out.println(a + b);
}
}
```

[Sample Code: Python]

```
import sys

a, b = (int(x) for x in sys.stdin.readline().strip().split())
sys.stdout.write("%d\n" % (a + b))
```

# 問題 1 – 旗幟繪製問題 (1\_Flag)

(10 分)

## 問題敘述

有一個猴子爭島的遊戲，目標是爭奪一面旗子，首先牠們要繪製一面共同認同的旗子，能畫層數越多代表牠們越愛這個島，這個旗子樣式如右圖：

旗子正中有半徑 7.5 公分的圓，圓心位置(0,0)，往外有「若干層」，每層往外間距 7.5 公分，每層 12 道光芒，每道光芒頂部夾角 30 度。在此，先就芒頂，以及芒腳定義如下：

芒頂：如圖例(每道光芒)頂點 A 所在位置為芒頂，每層 12 個芒頂，1~12 的排列與時鐘位置相同。

芒腳：如圖例(每道光芒)凹點 B 所在位置為芒腳，每層 12 個芒腳，起始點位置如圖示 1，順時針方向依序為 1~12 個芒腳。

第一層 12 道光芒的芒頂到圓心距離為 15 公分，第二層為 22.5 公分...依此類推。

請注意，示意圖中第一層的芒腳和半徑 7.5 公分的圓，並沒有重疊。勿為其解析度有限而誤解

## 輸入格式

M, N

M 層數，N 為第幾個芒腳，M, N 皆為正整數，並以逗號間隔。

## 輸出格式

x, y

芒腳的位置座標，四捨五入到小數點後 2 位輸出，並以逗號間隔，座標值若為負數，則需輸出“-”號(詳輸出範例)

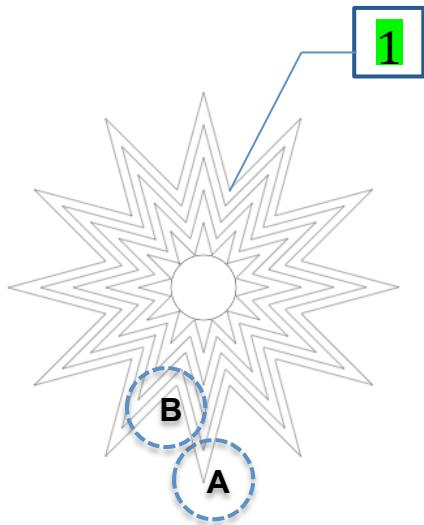
## 資料範圍

$1 \leq M \leq 99$ ,  $1 \leq N \leq 12$   
 $0.00 \leq |x, y| \leq 9999.99$

## 資料範例

### 輸入範例 1

1, 1



**輸出範例 1**

2.01, 7.50

**輸入範例 2**

3, 5

**輸出範例 2**

10.98, -10.98

**輸入範例 3**

10, 12

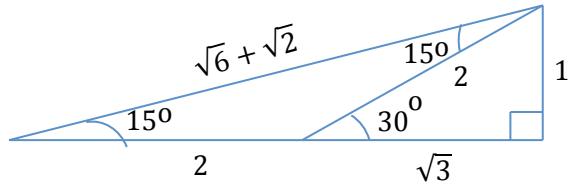
**輸出範例 3**

-11.05, 41.25

## 參考資料

 $\sin(x)$ 

x	$\sin(x)$
0	0.0
15	$(\sqrt{6}-\sqrt{2})/4$
30	0.5
45	$\sqrt{2}/2$
60	$\sqrt{3}/2$
75	$(\sqrt{6}+\sqrt{2})/4$
90	1.0



$$\sqrt{6} = 2.449489742783178$$

$$\sqrt{2} = 1.4142135623730951$$

## 問題 2 – 存量問題 (2\_Remainings)

(15 分)

### 問題敘述

猴子島上有個傳統，猴王專向其他猴子從採收的香蕉中抽成，但自己不採香蕉。

島上除猴王外有 20 隻猴子，每隻猴子每天固定採收 80 根香蕉。

第 1 天猴王向 20 隻猴子各收 1 根香蕉，第 2 天也向 20 隻猴子各收 1 根香蕉，第 3 天之後每天各收先前兩天所抽香蕉數的總和(即：第 3 天，各收 2 根( $1+1$ )；第 4 天，各收 3 根( $1+2$ )；第 5 天，各收 5 根( $2+3$ )，... 依此類推)。



然而，猴王抽成來的香蕉不新鮮，隔天有一半(或近一半)，即  $\lfloor \text{香蕉數} \times 0.5 \rfloor$  (註 1) 會壞掉。例如：

第 4 天時，前 3 天猴王抽成來的 55 根香蕉，會壞掉 27 根，剩 28 根，  
第 4 天，新收 60 根，  
第 4 天結束時，猴王有 88 根香蕉。

其他猴子擁有的香蕉比較新鮮，所以不會壞掉。

日子如此這般，當某天猴王收走其他猴子們所有的香蕉後，猴子們立即會發生暴動並選出新猴王。此時，舊猴王必須發給新猴王以外的每一個猴子(包括自己共 20 隻)各 N 根香蕉，剩餘的香蕉則由新猴王擁有。而 N 符合這樣條件：

$$\begin{aligned} N * 20 &\leq \text{舊猴王原擁有總數} * 95\% \\ (N+1) * 20 &> \text{舊猴王原擁有總數} * 95\% \\ N &\text{為正整數} \end{aligned}$$

例如：第 14 天原猴王有 10912 根香蕉，其他猴子各有 134 根香蕉，第 15 天每隻猴子又採了 80 根香蕉變成 214 根，但猴王依照算法當天要收 610 根，全部香蕉給猴王之後，都還不夠，所以引發暴動並產生新猴王。如前述之方式重分配後，新猴王拿到 496 根香蕉，其他猴子(包括舊猴王)則留下 462 根香蕉，然後下一天新猴王又向大家收 1 根香蕉，再一天 1 根香蕉，再來 2 根香蕉，依此類推。

島上資源有限，所以總猴子數為固定 21 隻。

註 1：下取整函數，在數學中一般記作， $\lfloor x \rfloor$ ，例如：

$$\begin{aligned} \lfloor 55 \times 0.5 \rfloor &= 27, \\ \lfloor 55 \times 0.95 \rfloor &= 52. \end{aligned}$$

## 輸入格式

輸入第 n 天，因為猴子百萬年前就有，因此假設  $n \leq 100,000,000$ 。

## 輸出格式

猴子(非猴王)所擁有的香蕉數量。

## 資料範圍

$0 < n \leq 100,000,000$ ；所有使用到的數字皆屬於長整數(long integer)。

## 資料範例

### 輸入範例 1

2

### 輸出範例 1

158

### 輸入範例 2

28

### 輸出範例 2

893

### 輸入範例 3

65

### 輸出範例 3

893

# 問題 3 – 優希的麻將牌(3\_Mahjong)

(10 分)

## 問題敘述

相信在比過 YTP 初賽的你/妳，已經知道，優希(Kataoka Yuuki)特別喜歡日麻中的「七對子」跟「國士無雙」這兩個役種了。

也因此，優希對於每張日麻牌的喜好程度都不一樣。讓我們回顧一下，日麻使用的牌為萬子牌（一萬到九萬，以  $1m$  到  $9m$  表示），餅子牌（一餅到九餅，以  $1p$  到  $9p$  表示），索子牌（一索到九索，以  $1s$  到  $9s$  表示），四風牌（東風、南風、西風、北風，以  $1z$ ， $2z$ ， $3z$ ， $4z$  表示），三元牌(白、發、中，以  $5z$ ， $6z$ ， $7z$  表示)，每種牌各有四張，總共 136 張牌、34 種牌。

現在，優希挑出了  $N$  種牌，每一種牌都挑選了「恰好一張」。她挑出這些牌後，會把這些牌排成一列放在桌子上。最左邊的牌是第一張牌，最右邊的牌是第  $N$  張牌。

排好牌之後，優希會在心中想像一個序列  $a_1, a_2, \dots, a_N$ 。其中  $a_i$  代表她對於第  $i$  張牌的喜好程度。因為優希實在太熱愛日麻了，所以喜好程度一定是一個不超過 1000 的正整數。

有了這些資訊之後，優希想要算出一個序列  $b_1, b_2, \dots, b_N$ ，其中  $b_i$  代表：「當優希想要挑一個長度為  $i$  的連續的牌時，她能夠挑選到的最大喜好程度平均值的數值」。當優希挑選第  $L$  張、第  $L+1$  張、……、第  $R$  張牌時，她得到的喜好程度平均值就是  $\frac{a_L + a_{L+1} + \dots + a_R}{R-L+1}$ 。注意到優希並不太喜歡小數，請用分數的方式表達  $b_1, b_2, \dots, b_N$ 。詳細的輸出方法請參考輸出格式以及範例輸入。

## 輸入格式

$N$

$a_1 \ a_2 \ \dots \ a_N$

輸入的第一行包含兩個正整數  $N$ ，代表優希挑出來的牌的數量。

接下來的一行，有  $N$  個以一個空白隔開的正整數  $a_1 \ a_2 \ \dots \ a_N$ ，其中  $a_i$  代表優希對第  $i$  張牌的喜好程度。

## 輸出格式

請輸出  $N$  行。

對於第  $i$  行，請輸出一個分數，代表  $b_i$ ，輸出的格式為「分子/分母」(不含引號)，分數必須化簡到最簡分數。

## 資料範圍

$1 \leq N \leq 34$

$1 \leq a_i \leq 1000$

## 資料範例

### 輸入範例 1

1  
1000

### 輸出範例 1

1000/1

### 解釋 1

優希只有挑出一張牌，而那張牌的美麗程度是 1000。  
所以符合題目條件的  $b_1$  就是 1000，輸出最簡分數 1000/1。

### 輸入範例 2

3  
60 161 28

### 輸出範例 2

161/1  
221/2  
83/1

### 解釋 2

優希挑出了三張牌，喜好程度分別是[60, 161, 28]。  
考慮計算  $b_1$ ，有三個可能的拿法[60], [161], [28]，平均最大的是[161]，平均值為 161/1。  
考慮計算  $b_2$ ，有兩個可能的拿法[60, 161], [161, 28]，平均最大的是[60, 161]，平均值為 221/2。  
考慮計算  $b_3$ ，只有一種可能的拿法[60, 161, 28]，平均最大的是[60, 161, 28]，平均值為 249/3，輸出最簡分數 83/1。

### 輸入範例 3

5  
1 2 3 4 5

### 輸出範例 3

5/1  
9/2  
4/1  
7/2  
3/1



## 問題 4 – 凱薩解密 (4\_Caesar)

---

(10 分)

### 問題敘述

凱薩大帝是羅馬由共和轉向帝國的重要人物，鼎鼎大名的凱薩加密法更是以他的名字來命名的，凱薩加密法的做法如下：

首先，我們有一個要拿來加密的鑰匙 K，以 K=3 為例子，

再來對要加密的字串 S 的每個字母，向後移 K 格。若遇到 Z 則下一個是 A。

例如:A->D , B->E , Z->C

就完成了加密後的字串了

今天有一位大臣收到了凱薩寄來的加密信，

可是他忘記鑰匙是甚麼了，身邊只有一份以前凱薩寄來的信跟他加密的結果，而且他知道凱薩每次都用一樣的鑰匙來加密，請你利用之前的結果來幫他解密

### 輸入格式

S1: 上次的明文

S2: 上次的密文

S3: 這次的密文

保證測試資料是合法的

### 輸出格式

這次的密文(S3)解密後的明文字串 S

### 資料範圍

$0 < |S1|, |S2|, |S3| \leq 1000$  ( $S1, S2, S3$  的長度 $\leq 1000$ )

### 資料範例

#### 輸入範例 1

TEST  
WHVW  
KHOORZRUOG

#### 輸出範例 1

HELLOWORLD

#### 輸入範例 2

QDOYKLHZH  
ERCMYZVNV  
WSBHMKLAT



### 輸出範例 2

IENTYWXMF

### 輸入範例 3

GXR SUTMX XVM EQX JW

V MG H J I B MM K BT F M Y L

C O I O A Z A Q G I S M P J F Y

### 輸出範例 3

N Z T Z L K L B R T D X A U Q J

# 問題 5 - 史萊姆 (5\_Slime)

---

(15 分)

## 問題敘述

當有  $N$  隻史萊姆整齊排列時，每經過一秒，原來相鄰的兩隻史萊姆會提供養分產生一隻新的史萊姆在中間的空隙中，並使自己消失。換句話說，當有  $N$  隻史萊姆排成一列，此時從前面數第  $i$  個史萊姆的養分數值為  $A[i]$ ，經過一秒後，新的  $N - 1$  隻史萊姆排成一列，其中第  $i$  隻的養分值為  $A[i] + A[i+1]$ 。這個現象會持續發生，直到場上只剩一隻史萊姆。

定義：一個史萊姆的壞度是他的養分值除以 1,000,000,007 的餘數。

現在你想知道當場上只剩一隻史萊姆時，他的壞度是多少。

## 輸入格式

第一行有一個整數  $N$ ，表示開始時有  $N$  隻史萊姆在場上。

第二行有  $N$  個以空白區隔的非負整數數列  $A$ ， $A[i]$  代表第  $i$  個史萊姆的養分值。

## 輸出格式

輸出一個非負整數，代表最後一隻史萊姆的壞度。

## 資料範圍

$1 \leq N \leq 100,000$

$0 \leq A[i] \leq 1,000,000,000$

## 資料範例

### 輸入範例 1

2  
1 2

### 輸出範例 1

3

### 輸入範例 2

3  
1 2 2

### 輸出範例 2

7

### 輸入範例 3

3  
1 2 3



### 輸出範例 3

8

### 輸入範例 4

10

779819952 258948867 10118107 648116442 673906045 627578303  
625309921 751119352 847506985 618648389

### 輸出範例 4

715959710

## 範例解釋

第三筆範例：

剛開始：[1, 2, 3]

經過一秒後：[3, 5] 原來的 1+2 變成 3，原來的 2+3 變成 5。

經過兩秒後：[8] 原來的 3+5 變成 8

# 問題 6 – 螺旋丸 (6\_Rasengan)

(10 分)

## 問題敘述

你因為在家閒閒沒事，因此你開始學鳴人搓螺旋丸。但是你搓的過於走火入魔，導致你現在看到什麼東西都想搓成螺旋狀的...

現在你看到了一條字串，你想要把他搓成下面的樣子...

16	15	14	13	12		
17	4	3	2	11		
18	5	0	1	10		
19	6	7	8	9	...	
20	21	22	23	24	25	

## 輸入格式

一行字串，字串裡面不會有空白，只會有英文和數字

## 輸出格式

請輸出搓完的樣子

注意，請勿輸出行末空白，並且行頭要使用最少的空格對齊。

## 資料範圍

字串長度不會超過 10000

## 資料範例

### 輸入範例 1

1ba

### 輸出範例 1

a  
1b

### 輸入範例 2

C8763

### 輸出範例 2

367



C8

**輸入範例 3**

rdrrMeow

**輸出範例 3**

Mrr

erd

ow

# 問題 7—PPPPPPP (7\_PPPPPP)

(20 分)

## 問題敘述

- $P$ : 一個 16-bit (註 1) 且大於 100 的質數。
- $N$ :  $P^7$ 。
- $e$ : 固定是 65537。
- $d \equiv e^{-1} \left( \text{mod}(p^6 \times (p - 1)) \right)$   
(註 2; 註 3)

註 1: 16-bit, **16** 位元整數可以儲存 $2^{16}$  (或 65536) 的不同的數值。使用無正負號的表示方法，這些數值是介於 0 到 65535 的整數；

註 2:  $e^{-1}$  是一個值，會使得:  $e * e^{-1} = 1 \text{ (mod } N\text{)}$ 。

註 3:  $\equiv$  為同餘符號， $A \equiv B \text{ (mod } N\text{)}$  代表 A 與 B 除以 N 以後有相同的餘數。

## 輸入格式

一開始有三個整數，以空白間隔，分別是  $N$ ,  $Q$ ,  $d$ 。( $N$ ,  $d$  定義如上)  
接下來有  $Q$  行，每行一個整數  $c$

## 輸出格式

$Q$  行，每行一個整數  $m$ ，使得每個  $m$ ，相對應的  $c$  有如下的關係  
 $c \equiv m^e \text{ (mod } N\text{)}$ ，而且，輸出的  $m$  必須滿足  $0 < m < P$  這個限制

## 資料範圍

- $100 < P < 2^{17}$
- $0 < c < N$
- $0 < m < P$
- $0 < Q < 11$
- $0 < d < (p^6 \times (p - 1))$

## 資料範例

### 輸入範例 1

62764785704439251 1 2799555438492723  
3806193642843218

### 輸出範例 1

150



### 輸入範例 2

```
478564714635879102354249253493939 10
86178793801477300477479930707141
12758220350707378154868818083880
29260013564752897893858934306111
66643948912065995227756374274536
305749958937069957101396094291602
26888494424599108027243163189844
375340073210863366694031165886746
284170402693730499688845325858940
82798665271677178276261636999761
19176468761183329423864442652361
128794217885273988157866793866023
```

### 輸出範例 2

```
809
19063
14995
40743
30844
2270
15696
14092
7781
18191
```

# 問題 8 – 連續重複數字串 (8\_RepeatedNumber)

(10 分)

## 問題敘述

請寫一支程式，輸入一數字串。找出這串數字中連續重複最多次的子數字串，以及重複的次數。

取子字串時，重複的子字串，其位置不能相交。例如，23232，就不能視作 232 有重複。而 01、001、0003 … 這些 0 在前的數字串都是合法的子字串。

## 輸入格式

- 一行只有數字的字串

## 輸出格式

- 第一行輸出為，出現連續重複次數最多的子數字串
- 第二行輸出為，該子數字串出現的次數

## 資料範圍

- 字串長度不超過 100
- 字串由阿拉伯數字(0-9)組合而成
- 當有重複次數一樣的兩個子數字串時，輸出子數字串長度較短的。如果兩個輸出長度也一樣，則選較早在輸入數字串中出現的。

## 資料範例

### 輸入範例 1

111112323232323111

### 輸出範例 1

1  
5

### 範例 1 說明

此數字串中 1 連續重複次數與 23 一樣，但是 1 長度較短所以輸出為 1 最多次，其次數為 5。

### 輸入範例 2

1212121

### 輸出範例 2

12  
3



## 範例 2 說明

12 與 21 出現次數相同為 3 次所以輸出較早出現的答案，12 比 21 早出現所以答案為 12 而連續重複 3 次。

# 問題 9—卡車載貨 (9\_Truck)

(15 分)

## 問題敘述

有一部卡車要載運貨物去賣，卡車有載重上限而貨物有各自的重量與價值，由於貨物過多可能無法全部載完，所以在有限的載重能力下找出最大載貨價值，是很重要的。

請寫一支程式，在卡車載貨重量限制下，選擇可載貨物中總價值最高的組合，並輸出此最高的總貨物價值。

## 輸入格式

第一行為卡車的重量上限  $W_t$  跟可被挑選貨物的件數  $N$ ，以空白間隔  
第二行之後，有  $N$  行，每行表示每件貨物的價值  $V_i$  跟重量  $W_i$ ，以空白間隔

## 輸出格式

一行，輸出載貨的總價值

## 資料範圍

$W_t \leq 100,000$

$N \leq 10,000$

$V_i \leq 2,500$

$W_i \leq 1,000$

以上皆為正整數

## 資料範例

### 輸入範例 1

50 3

60 10

100 20

120 30

### 輸出範例 1

220

### 範例 1 說明

$W_t$  為 50 而  $N$  為 3，最佳解為重量  $20+30=50$ ，價值為  $100+120=220$

# 問題 10 – 序列操作 (10\_Sequence)

(20 分)

## 問題敘述

你沒看錯，這就是一題序列操作題。

給你一個長度為  $N$  的序列  $a_1, a_2, \dots, a_N$ ，以及一個正整數  $K$ 。定義  $S$  為當前序列中，區間連乘積  $\leq K$  的區間的個數，亦即， $S$  為滿足  $a_L \times a_{L+1} \times \dots \times a_R \leq K$  的  $(L, R)$  個數。

接下來，會有  $Q$  筆操作，第  $i$  筆操作的形式為  $X_i Y_i$ ，代表把  $a_{X_i}$  改成  $Y_i$ 。請你在進行操作之前，以及每筆操作完後，輸出當前的  $S$  的值。

## 輸入格式

$N\ K$   
 $a_1\ a_2\ \dots\ a_N$   
 $Q$   
 $X_1\ Y_1$   
 $X_2\ Y_2$   
...  
 $X_Q\ Y_Q$

輸入的第一行包含兩個正整數  $N, K$ ，以空白間隔，代表序列的長度，以及一個參數。

接下來的一行，有  $N$  個以一個空白隔開的正整數  $a_1\ a_2\ \dots\ a_N$ ，其中  $a_i$  代表序列的第  $i$  項。

接下來的一行，包含一個整數  $Q$ ，代表操作的次數。

接下來的  $Q$  行，第  $i$  行包含兩個正整數  $X_i Y_i$ ，代表要把序列的第  $X_i$  項改成  $Y_i$ 。

## 輸出格式

輸出  $Q+1$  行。

第一行輸出為，未經任何操作前的  $S$  值。

第二行起的  $Q$  行，依序輸出執行前  $i$  個操作後，當前序列的  $S$  值。

## 資料範圍

$1 \leq N \leq 200000$   
 $1 \leq a_i, K, Y_i \leq 10^9$   
 $0 \leq Q \leq 200000$   
 $1 \leq X_i \leq N$

## 資料範例

### 輸入範例 1

3 4  
1 2 3  
0

### 輸出範例 1

4



### 範例 1 說明

因為  $Q = 0$ (沒有任何操作) , 所以只要輸出原本序列的  $S$  值就好了。

而在原本的序列中，符合條件的( $L, R$ ) 有  $(1,1), (2,2), (3,3), (2,3)$  ，總共有四個(也就是說，有四個區間的連乘積  $\leq 4$ )。

### 輸入範例 2

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

### 輸出範例 2

```
10
10
7
5
```

### 範例 2 說明

一開始的序列為 $[1, 1, 1, 1]$  , 可以發現所有的子區間的連乘積都  $\leq 5$  , 於是  $S = 10$  。

執行第一個操作後，序列變成 $[5, 1, 1, 1]$  , 所有子區間的連乘積都還是  $\leq 5$  , 於是  $S = 10$  。

執行第二個操作後，序列變成 $[5, 2, 1, 1]$  , 符合的( $L, R$ ) 只剩下  $(1, 1), (2, 2), (2, 3), (2, 4), (3, 3), (3, 4), (4, 4)$  , 總共七個，所以  $S = 7$  。

執行第三個操作後，序列變成 $[5, 2, 3, 1]$  , 符合的( $L, R$ ) 只剩下  $(1, 1), (2, 2), (3, 3), (3, 4), (4, 4)$  , 總共五個，所以  $S = 5$  。

### 輸入範例 3

```
6 2847598
24987 20937503 92831 2398594 23895 1938652
5
5 98435931
1 39864
3 985986
2 348
4 1
```

### 輸出範例 3

```
5
4
4
4
5
6
```

# 問題 11 - 賽程安排(11\_Tournament)

(15 分)

## 問題敘述

有天你上體育課的時候，老師介紹了正式比賽中，淘汰賽程的賽程排法。我們假設參賽者都有個種子序（例如：16 強中，有第 1 種子到第 16 種子），那麼雖然會有很多等價的賽程，但正式比賽中會有唯一一種賽程的排法。假設你觀察了許久後，發現在  $2^N$  強淘汰賽的賽程有以下這些規則：

- 每個位置都有一個籤號，依序為 1 號籤到  $2^N$  號籤，1 號籤會對上 2 號籤、3 號籤會對上四號籤，依此類推。
- 為了不要讓兩個種子序很小的選手太早相遇，因此第 1 種子會對上第  $2^N$  種子、第 2 種子會對上第  $2^N-1$  種子，依此類推。
- 奇數種子序的人要在奇數籤號、偶數種子序的人要在偶數籤號。
- 當  $N > 0$  時，經過一輪比賽後，如果種子序小的人都勝過了種子序大的人，將 1, 2 號籤的贏家當成新的 1 號籤、3, 4 號籤的贏家當成新的 2 號籤，依此類推，會產生出一個新的  $2^{N-1}$  強的淘汰賽程，這個新的淘汰賽程裡面成員的種子序順序也必須滿足上述的條件。

為了複習方便以及順利拿到這題的分數，你想要寫一個程式：給定兩個正整數  $N, X$ ，輸出在一個  $2^N$  強的淘汰賽當中，第  $X$  種子會是哪一個籤號？

## 輸入格式

第一行包含一個整數  $T$ ，表示接下來有幾筆測試資料

接下來的  $T$  行中，每行有兩個整數  $N, X$  (以空白間隔)，表示有一個詢問希望你回答在一個有  $2^N$  個參賽選手的淘汰賽程當中，第  $X$  種子會在哪一個籤號。

## 輸出格式

對於每一個詢問請輸出一個整數表示該種子的籤號，詳情請見資料範例。

## 資料範圍

$$1 \leq T \leq 100, \quad 0 \leq N \leq 60, \quad 1 \leq X \leq 2^N$$

## 資料範例

### 輸入範例 1

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

### 輸出範例 1

1  
1  
2

### 輸入範例 2

4  
2 1  
2 2  
2 3  
2 4

### 輸出範例 2

1  
4  
3  
2

### 輸入範例 3

4  
3 1  
3 2  
3 5  
3 6

### 輸出範例 3

1  
8  
3  
6

### 範例 3 說明

在八強賽中，1 號籤到 8 號籤的種子序依序為：

1, 8, 5, 4, 3, 6, 7, 2

輸入 3 1 -》所以，2 的 3 次方的種子球員中，第 1 種子，會在籤號 1

輸入 3 2 -》所以，2 的 3 次方的種子球員中，第 2 種子，會在籤號 8

輸入 3 5 -》所以，2 的 3 次方的種子球員中，第 5 種子，會在籤號 3

輸入 3 6 -》所以，2 的 3 次方的種子球員中，第 6 種子，會在籤號 6



## 問題 12 - 水星逆行(12\_Reverse)

(20 分)

### 問題敘述

水星逆行會導致運氣不順，遭遇到一些不幸的事情，例如上廁所剛好沒衛生紙、買 21 元的東西結果手上只有 20 元的零錢跟一張千元大鈔等等，是很可怕的事情。

但水星逆行會導致壞事的原理並沒有那麼簡單。在西元 8787 年，宇宙學家發現這世界上一共有  $N$  個平行時空，在思維上可以把他們當成  $N$  層的甜甜圈，由內而外排列，每一個時空也都有一個水星。同時，會有  $N \times (N-1)/2$  個人，每一個人都同時身處在兩個平行時空，而且任意兩個人所處的平行時空至少有一個是不同的，當某個人的內圈時空的水星移動速度比外圈時空的水星快的時候（稱作「水星逆行」），那個人的運氣就會變得超級差。

但人們運氣差的時候也還好，因為他們會跨時空互相討拍。兩個運氣差的人彼此討拍後，運氣就會變好。但總可能會有落單的人，因為運氣不好的人都倆倆互相拍拍了，落單的人沒有人可以討拍，他將會變成這個世界上最不幸的一個人。

幸好，不知道為什麼，時空每一天都會有些變化，使得這世界上不一定一直都存在落單的可憐人。

每一天的開始，恰一個連續的時空區段會被翻轉，翻轉的時空區段內，時空的內外順序會被逆轉。例如，原先由內而外的時空依序是 1 到 6 號。若是翻轉 2 到 5 號後，由內而外依序將變為 1,5,4,3,2,6 號。為了判斷每一天有沒有運氣差的落單人，科學家找上了你，希望請你判斷在每一次翻轉過後，這世界上有沒有運氣差的落單人。

### 輸入格式

第一行包含一個整數  $N$ ，代表這個世界上有  $N$  個平行時空與  $N \times (N-1)/2$  個人。每個人所處的平行時空組合皆不一樣。

第二行包含  $N$  個非負整數  $v_i$ ，依序代表由內而外的時空中，水星的移動速率，保證每個時空間的水星移動速率皆不相同。

第三行包含一個整數  $Q$ ，代表科學家的機器翻轉了  $Q$  次時空。

接下來有  $Q$  行，每行有兩個整數  $L, R$ ，代表機器翻轉了由內數來第  $L$  個至第  $R$  個的時空。

### 輸出格式

對於每次翻轉時空事件，請輸出一行，若該次翻轉後有運氣差且落單的人，請輸出 "QQ"(不含引號)，若沒有，請輸出 "Happy"(不含引號)。

### 資料範圍

$$1 \leq N \leq 300000$$

$$0 \leq v_i \leq 10^9$$

$$1 \leq Q \leq 10^6$$

$$1 \leq L \leq R \leq N$$



## 資料範例

### 輸入範例 1

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

### 輸出範例 1

```
Happy
QQ
QQ
```

### 範例 1 說明

翻轉前：序列 1,3,2,4,5，共有 1 個人運氣差

第一次翻轉後，序列變為 2,3,1,4,5，共有 2 個人運氣差，故沒有人落單

第二次翻轉後，序列變為 2,3,5,4,1，共有 5 個人運氣差，故有一個人落單

第三次翻轉後序列不變，故依然有一個人落單。

### 輸入範例 2

```
7
1 2 3 4 5 6 7
3
3 6
2 5
1 5
```

### 輸出範例 2

```
Happy
Happy
Happy
```

### 輸入範例 3

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

### 輸出範例 3

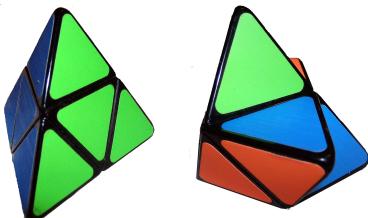
```
QQ
Happy
Happy
```

# 問題 13 – 魔粽 (13\_Rubix)

(15 分)

## 前言

自從經典的 3x3 魔術方塊發明以來，各玩具廠就爭相推出不同形狀的魔術方塊，而魔粽便是其中之一。魔粽看似簡單，卻能有形狀上的變化，在書局中總能帶給消費者開封試轉的衝動，是一樣相當成功的商品。



圖片來源：<https://en.wikipedia.org/wiki/Pyramorphix>

有趣的是，魔粽跟經典的 2x2 魔術方塊同構。也就是說如果你會解六面的 2x2 方塊，那你就用同樣的方法解魔粽。

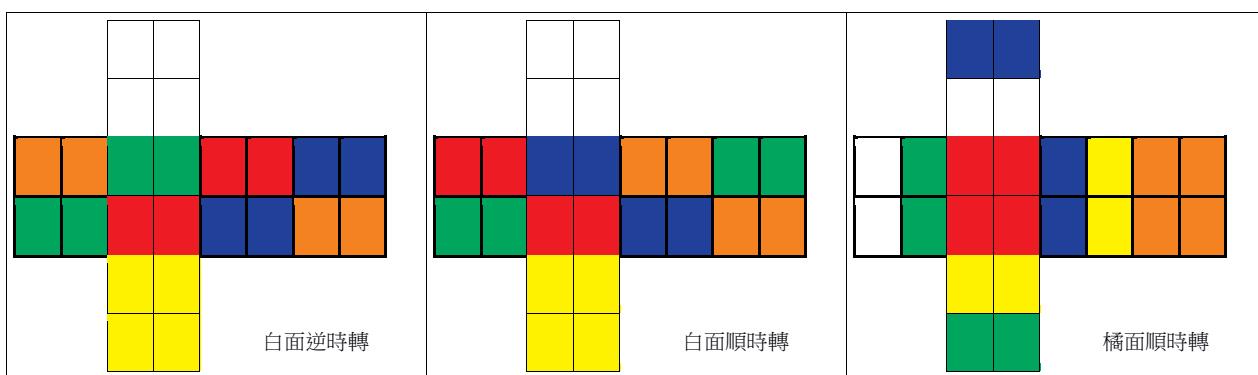
## 問題敘述

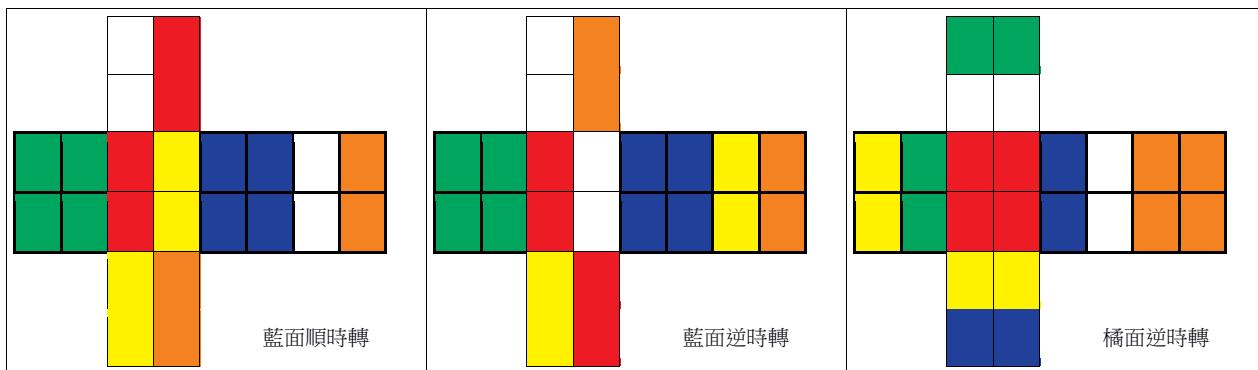
身為一個天才魔粽新手，你常常不確定自己直覺想到的解法是不是最快的。於是決定每次都將魔粽解讀成 2x2 方塊，輸入程式中計算最少要轉幾下才能解出來，然後再檢查目前的解法有沒有超過最少步數。

請寫一支程式，計算某個 2x2 方塊最少可以在幾步內解出六面來。這隻程式的輸入是各位置顏色（白(w)、綠(g)、紅(r)、藍(b)、橘(o)、黃(y)）所組成的字串，輸出則是復原該方塊所需的最少步數（每轉 90 度就算一步）。每步有 6 種可能：

	S <sub>0</sub>	S <sub>1</sub>																						
	S <sub>2</sub>	S <sub>3</sub>																						
S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	S <sub>7</sub>	S <sub>8</sub>	S <sub>9</sub>	S <sub>10</sub>	S <sub>11</sub>																	
S <sub>12</sub>	S <sub>13</sub>	S <sub>14</sub>	S <sub>15</sub>	S <sub>16</sub>	S <sub>17</sub>	S <sub>18</sub>	S <sub>19</sub>																	
	S <sub>20</sub>	S <sub>21</sub>																						
	S <sub>22</sub>	S <sub>23</sub>																						

原方塊





## 輸入格式

$C_0C_1C_2C_3C_4C_5C_6C_7C_8C_9C_{10}C_{11}C_{12}C_{13}C_{14}C_{15}C_{16}C_{17}C_{18}C_{19}C_{20}C_{21}C_{22}C_{23}$

輸入共一行，24 個字母，沒有分隔，每個字母  $C_i$  代表上圖中  $S_i$  位置的顏色。

## 輸出格式

L

輸出共一行，數字 L 代表最小步數。

## 資料範圍

$C_i \in \{w, g, r, b, o, y\}$

顏色必為合法的字母

$C_{13} = g, C_{14} = r, C_{20} = y$

為簡單起見，輸入值必符合  $C_{13} = g, C_{14} = r, C_{20} = y$ ，

而且輸入保證有解

## 資料範例

### 輸入範例 1

wwwwggrrbbooggrrrbbooyyyy

### 輸出範例 1

0

### 輸入範例 2

wwwwbbooggrrggrrrbbooyyyy

### 輸出範例 2

2

### 輸入範例 3

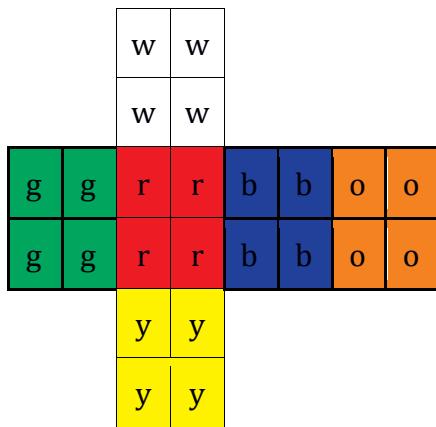
wowgrrbwobygggrwobyoybyr

### 輸出範例 3

2

### 範例 1 說明

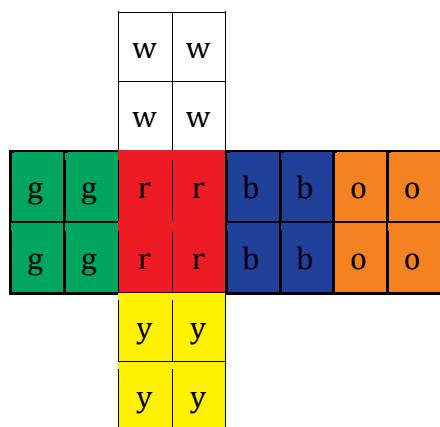
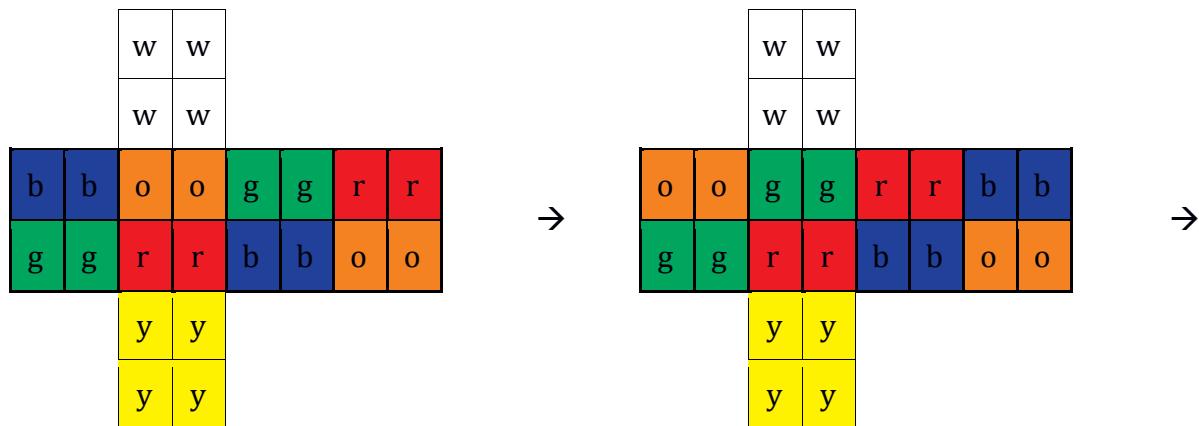
wwwggrrrbbooggrrbbooyyyy



轉動 0 步即解

### 範例 2 說明

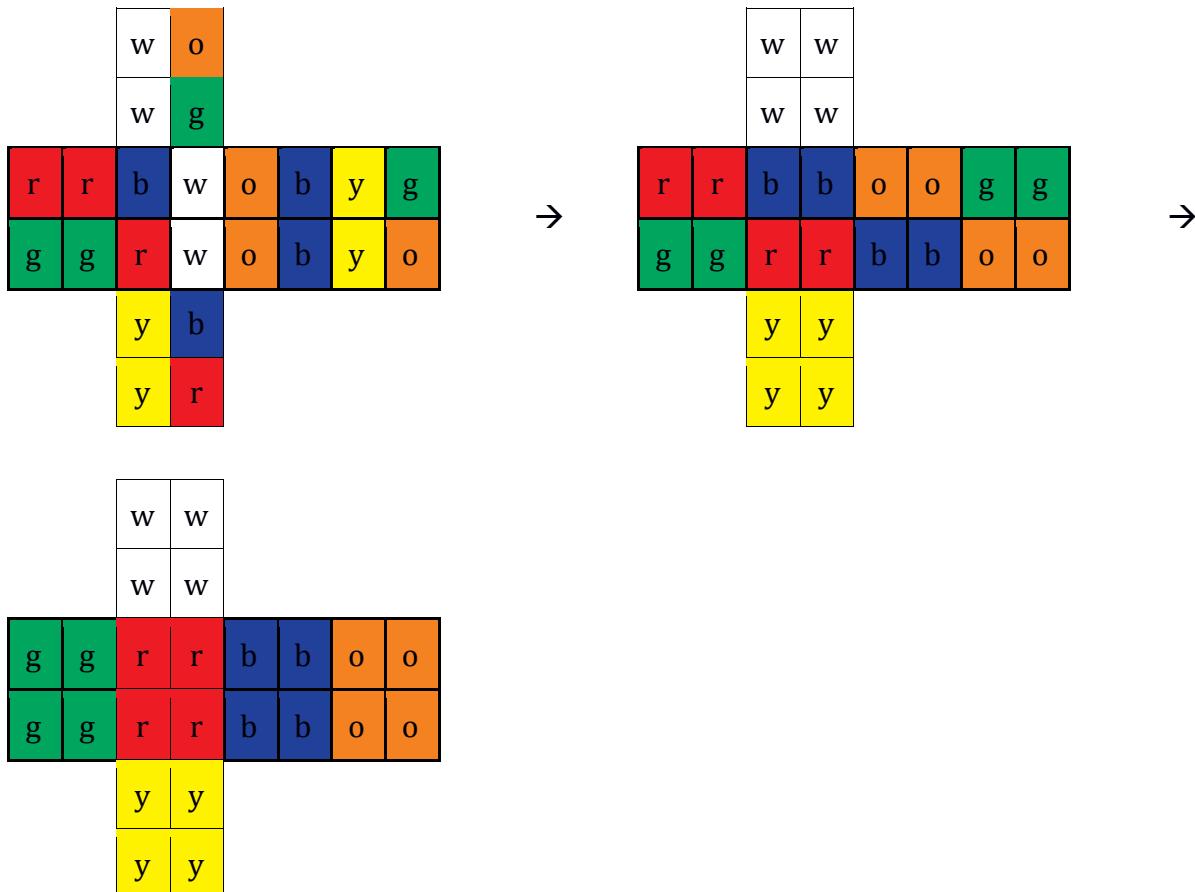
wwwbbooggrrrgrrbbooyyyy



轉動 2 步即解

## 範例 3 說明

wowgrrrbwobygggrwoboybyr



轉動 2 步即解

# 問題 14 – 神奇的變換(14\_MarvelousTransform)

(20 分)

## 問題敘述

我們來考慮一個轉換。給你一個字串  $s$ ，你要對他做些神奇的事：

第一步：照順序把  $s$  所有的循環一列一列排。

第二步：將那  $\text{len}(s)$  (也就是  $s$  的長度) 個字串照字典序(備註 1)排序。

第三步：回傳每列最後一行的字元所組成的字串，以及原字串  $s$  所在列之序號。若有很多列與原字串相同，會隨機回傳一列。

例如(範例 4)： $s = \text{guava}$

第一步：找出所有循環

guava  
uavag  
avagu  
vagua  
aguav

第二步：排序

aguav  
avagu  
guava  
uavag  
vagua

第三步：回傳結果

aguav<sup>v</sup>  
avagu<sup>u</sup>  
guava<sup>a</sup>  
uavag<sup>g</sup>  
vagua<sup>a</sup>

因為  $\text{guava}$  在第 3 列  
所以回傳(**vuaga, 3**)

我們把經過這個變換所回傳的結果記為  $(\tilde{s}, id)$ 。

你以為這題就是要做這個嗎？才不呢。

你要做的事是：給你  $(\tilde{s}, id)$ ，請你求出原本的  $s$ 。

## 輸入格式

$\tilde{s} \ id : \tilde{s}$  為小寫英文字串。 $id$  為正整數，與  $\tilde{s}$  以空白分隔。

## 輸出格式

$s : s$  為原本的那個字串。

## 資料範圍

$1 \leq |\tilde{s}| \leq 1000, 1 \leq id \leq |\tilde{s}|$

## 資料範例

### 輸入範例 1

ntaoronef 1

### 輸出範例 1

afternoon

### 輸入範例 2

rsptuo 4

### 輸出範例 2

sprout

### 輸入範例 3

pvxswdfguhjknbomearycqztl 17

### 輸出範例 3

qwertyuiopasdfghjklzxcvbnm

### 輸入範例 4

vuaga 3

### 輸出範例 4

guava

備註 1: 字典序的定義是：以 a、b、c … z 的順序排列；如果第一個字母一樣，那麼比較第二個、第三個乃至後面的字母。如果比到最後兩個單詞不一樣長（比如，sigh 和 sight），那麼把短者排在前。（From 維基百科）



# 問題 15 - 火車站 (15\_TrainStation)

---

(20 分)

## 問題敘述

事情是這樣的，小莫最近沉迷了一款手機遊戲，而這遊戲的內容，正式小莫最喜愛的火車！你可以利用各式各樣不同的火車頭、車廂，組起來後開出去賺錢、運物資，然後賺到的錢可以拿來.....額，沒錯，買房子！

問題就在這裡！房子可以用來賺錢，但是可以拿來蓋房子的街道長度有限，數量也有限。目前小莫的火車站擁有 2 條街。小莫現在很煩惱，到底要蓋那些房子，才能賺最多錢呢？請你幫幫他吧！

## 輸入格式

第一行3個數字**L**, **N**和**M**，表示街道長度**L**，有**N**種建築物可以建造，而另外還有**M**種比較特別的建築物，每一種最多只能蓋一棟。2條街是一樣長ㄉ！

接下來有**N**行，每行兩個數字**w**和**a**，代表建築物的賺錢能力和佔地**units**

接下來還有**M**行，以同樣的方式描述**M**種特別建築物

## 輸出格式

輸出一個數字 **P**，代表小莫的火車站最多能用這些建築物賺多少錢

## 資料範圍

$0 \leq N \leq 100$

$0 \leq M \leq 100$

$2 \leq L \leq 500$

$2 \leq w \leq 100$

$2 \leq a \leq 100$

## 資料範例

### 輸入範例 1

```
10 2 3
3 2
5 3
2 3
7 4
11 6
```

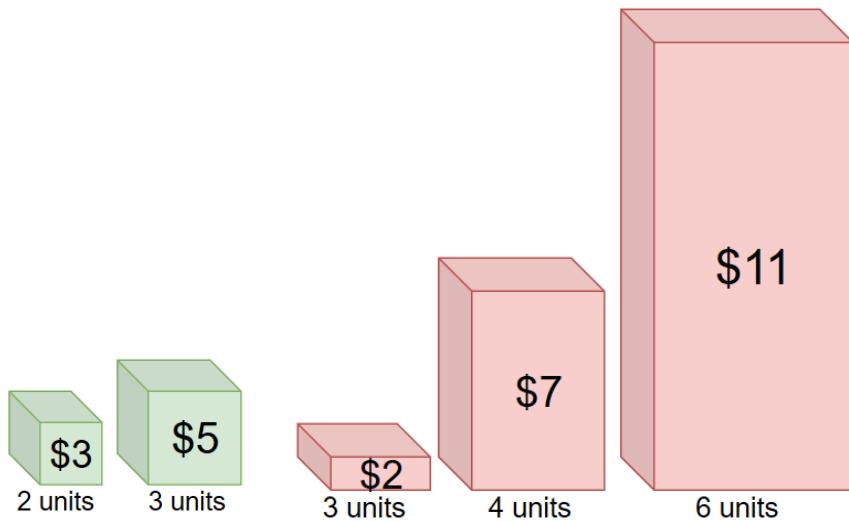
### 輸出範例 1

34

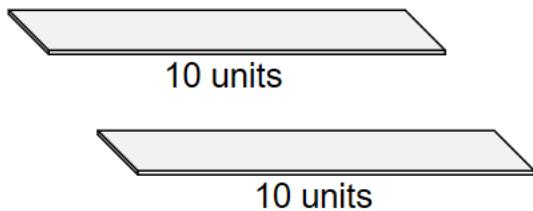
**範例 1 說明**

這些是你可以蓋的建築物：

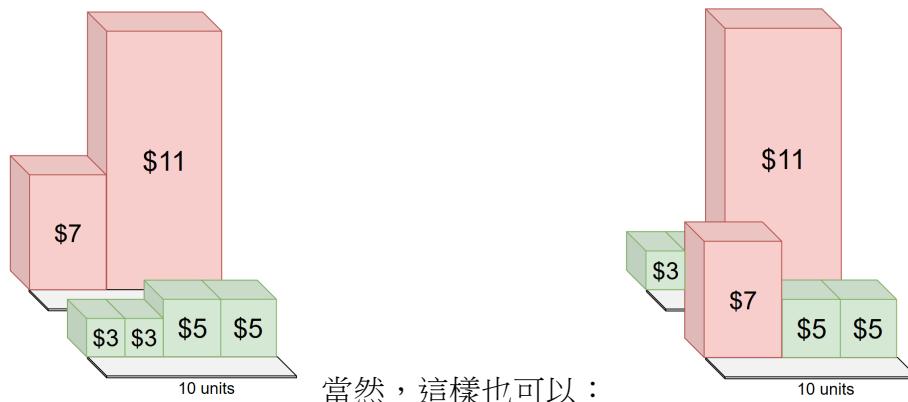
左邊綠色的可以隨便你蓋，右邊紅色的每一種最多只能蓋一棟



這是你可以用來蓋建築物的街道，有 2 條：



利用這些建築物，你最多可以賺 34 元，這是其中一種方式：



當然，這樣也可以：

不管怎樣，最多就是賺 34，所以答案是 34

**輸入範例 2**

11 2 0  
2 2  
4 3

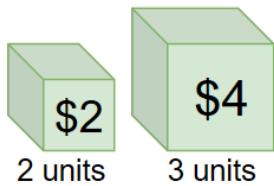
**輸出範例 2**

28

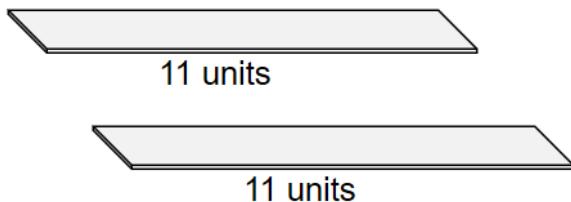
**範例 2 說明**

這是你可以蓋的建築物：

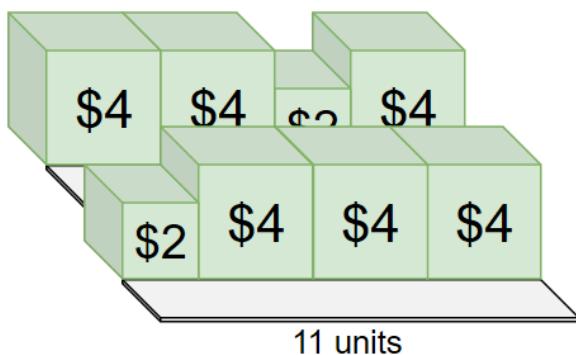
在這筆測資中，2 種建築物的數量都是沒有限制的



這是街道：



你最多可以賺 28 元，方式如下：



# 問題 16 – 打牌 (16\_Card)

(25 分)

## 問題敘述

「枯，臺式麻將，屁胡」

看到身邊的小 Y 熱愛麻將，這天小 P 又遇到他在打麻將，小 P 再也受不了了，你決定來搗亂，趁小 Y 摸牌後上廁所之際，幫他出牌。

讓我們來簡化小 Y 在玩的遊戲，注意牌的組成與現實麻將牌並不相同：

一共有 **50** 種種類的牌，分別是 1 條、2 條……、50 條，且每一種牌有無限多張。

而一副胡牌的麻將共有 17 張牌，要胡牌須組合完成 5 組加 1 對的形式：

一組：一組 3 張，三張相同或連號皆可謂完成組合。

例如：(42 條、42 條、42 條) 或 (20 條、21 條、22 條)

一對：2 張牌相同，謂為一對。

例如：(50 條、50 條)

可以胡牌的例子如下(省略「條」字)：

1、1、1、2、2、2、4、5、6、1、2、3、9、10、11、42、42

現在小 P 已經決定好幫小 Y 出牌的幾個選項，請你算算打掉指定的牌之後，剩下 16 張牌的“向聽數”，和“進張”種類，定義“聽牌”、“向聽數”、“進張”種類如下：

聽牌：若這 16 張牌加上一張牌 A 後就可以胡牌，則稱為聽牌，並稱牌 A 為候選牌。

向聽數：若以最佳的方法更換 16 張牌中最少的 B 張牌即可聽牌，則稱為 B 向聽。

進張：若已經聽牌，則所有候選牌都是進張；若沒聽牌，則進張為可讓向聽數少一的牌。

以下舉例說明，若小 P 幫小 Y 打牌後的牌如下(省略「條」字)：則

1、1、1、2、2、2、3、3、3、5、5、7、7、9、9、12

向聽數 = 1：若將「12 條」打掉，換成以下任一進張種類的牌，便可聽牌

進張種類 = 5：若得到「5 條、6 條、7 條、8 條、9 條」並換掉「12 條」，便可以聽牌  
(使原本向聽數 = 1 變成向聽數 = 0)



## 輸入格式

$T_1, T_2, \dots, T_{17}$

N

$D_1, D_2, \dots, D_N$

輸入總共有三行。

第一行共 17 個數字(以空白間隔)，第  $i$  個數字代表小 Y 現在的第  $i$  張牌是  $T_i$  條。

第二行有一個正整數 N，代表小 P 考慮的方案數。

第三行共有 N 個數字，第  $i$  個數字代表考慮把  $D_i$  張牌打出去。

## 輸出格式

共輸出  $2N$  行，每行依序處理對於輸入第三行的每個數字  $D_i$

在第  $2i-1$  行輸出第  $D_i$  張牌打出去後，剩下來的牌的向聽數與進張種類。

在第  $2i$  行輸出第  $D_i$  張牌打出去後，剩下來的牌的每一種進張。若有多種進張，請依數字大小由小到大輸出。

## 資料範圍

$1 \leq T_1 \sim T_{17} \leq 50$

$1 \leq N \leq 17$

$1 \leq D_i \leq 17$

## 資料範例

### 輸入範例 1

```
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
1  
1
```

### 輸出範例 1

```
0 1  
1
```

### 輸入範例 2

```
1 1 1 2 3 4 5 6 7 8 9 9 9 42 42 42 50  
2  
16 17
```

### 輸出範例 2

```
1 14  
1 2 3 4 5 6 7 8 9 10 42 48 49 50  
0 10  
1 2 3 4 5 6 7 8 9 10
```

### 輸入範例 3

```
10 11 13 14 16 17 20 20 1 1 1 4 4 4 7 7 7  
3  
1 2 7
```



### 輸出範例 3

```
1 3
12 15 18
1 3
12 15 18
2 11
9 10 11 12 13 14 15 16 17 18 20
```

### 輸入範例 4

```
1 1 1 2 2 2 3 3 3 5 5 7 7 9 9 12 13
8
1 4 7 10 12 14 16 17
```

### 輸出範例 4

```
2 9
1 4 5 6 7 8 9 11 14
2 9
2 4 5 6 7 8 9 11 14
2 9
3 4 5 6 7 8 9 11 14
1 5
6 7 9 11 14
1 6
5 6 8 9 11 14
1 5
5 7 8 11 14
1 5
5 6 7 8 9
1 5
5 6 7 8 9
```



## 問題 17 – 英靈召喚 (17\_Fate)

(25 分)

### 前言

みたせ　みたせ　みたせ　みたせ　みたせ  
閉じよ。閉じよ。閉じよ。閉じよ。閉じよ。  
くかえ　ごど  
繰り返すつどに五度。  
み　とき　はきやく  
ただ、満たされる刻を破却する。  
す　ぎん　てつ　そ　いし　けいやく　たいこう  
素に銀と鉄。礎に石と契約の大公。  
そ　わ　だいし  
祖には我が大師シュバインオーグ。  
お　た　かぜ　かべ  
降り立つ風には壁を。  
しほう　もん　と　おうかん　い　おうこく　いた　さんさろ　じゅんかん  
四方の門は閉じ、王冠より出で、王国に至る三叉路は循環せよ。  
つ  
告げる。  
なんじ　み　わ　した　わ　めいうん　なんじ　けん  
汝の身は我が下に、我が命運は汝の剣に。  
せいはい　よ　したが　い　ことわり　したが　こた  
聖杯の寄るべに従い、この意、この理に従うならば応えよ。  
ちか　こ　こ  
誓いを此処に。  
われ　とこ　よ　すべ　ぜん　な　もの  
我は常世総ての善と成る者、  
われ　とこ　よ　すべ　あく　し　もの  
我は常世総ての悪を敷く者。  
なんじさんだい　ことだま　まと　しちてん　よくし　わ　き　てんびん　まも　て  
汝三大の言霊を纏う七天、抑止の輪より来たれ、天秤の守り手よ！

聖杯戰爭就要開打了，小凜已經把召喚英靈的咒語背得滾瓜爛熟，只差在正確的時機詠唱，以正確的召喚小凜想要的英靈，不過不知道什麼緣故，小凜家的時鐘比正確的時間快了一個小時。

召喚英靈除了咒語不能唱錯之外，也要十分精準的掌握自己的魔力狀況，然而一個人的魔力是十分難以掌握的，不僅左右手的施法魔力不同，每分鐘魔力起伏都是不同的，但幸虧小凜家世世代代都在研究如何推算一個人在特定時間的魔力值，特別透露公式如下！

### 問題敘述

一個人左手與右手的魔力值可以分別用一個  $N \times N$  的矩陣來表示，先將矩陣由左到右，由上到下編號，則

$$\begin{bmatrix} a_1 & a_2 & a_3 & \dots & a_N \\ a_{N+1} & a_{N+2} & \dots & & \vdots \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & \vdots \\ a_{N(N-1)+1} & \dots & \dots & \dots & a_{N \times N} \end{bmatrix} \text{ 有 } a_x = f(a_{x-1}, h, m)$$

其中  $a_0$  是小凜查表計算的值，但是左右手的  $a_0$  可能會是不同的， $h, m$  是個正整數，表示現在正確時間的時與分，採用 24 小時制，即  $0 \leq h < 24$  且  $0 \leq m < 60$ ，而函數  $f$  的計算方法如下：

$$f(a, h, m) = a \times 1103515245 + (h \times 7122 + m) * 2 + 1$$

為了方便，我們假設小凜可以在瞬間詠唱完咒語，不影響時間。當小凜詠唱完咒語後，就能根據當下的左手與右手的魔力值矩陣  $M_L, M_R$ ，召喚出靈基魔力值矩陣為  $M_L \times M_R$  的英靈！計算乘法時，每個結果都要取其除以  $2^{32}$  的餘數作為答案。(若不知道如何對矩陣乘法請參閱篇末的公式 )

而小睡一會兒後準備後，小凜發現自己家的時鐘就要到達  $h$  點  $m$  分了，於是小凜匆忙的趕緊準備召喚鎮，恰好在家裡的時鐘到達  $h$  點  $m$  分時，詠唱咒語開始召喚儀式！詠唱完畢後移陣狂風襲過，隨之揚起了大量的煙塵，在煙塵消散之際，小凜終於發現自己召喚了一個.....怪怪的英靈？

根據測定，小凜召喚的英靈的靈基是一個  $N \times N$  的矩陣  $M_S$ ，見到此狀的你，在得到小凜計算出的左右手的  $a_0$  後，能幫忙檢查小凜是否有正確的計算  $a_0$ ，使其能召喚  $M_S$  的英靈呢？

別忘了！”不知道什麼緣故，小凜家的時鐘比正確的時間快了一個小時”

## 輸入格式

第一行有三個數字  $N \ h \ m$ ，如題目所示，表示了矩陣的大小，以及召喚時小凜家裡的時鐘是  $h$  點  $m$  分。

第二行有兩個數字  $L \ R$  (以空白間隔)， $L$  表示左手的  $a_0$ 、 $R$  表示右手的  $a_0$ 。

接下來有  $N$  行，每行有  $N$  個數字，表示矩陣  $M_S$ 。

## 輸出格式

請輸出一行文字，如果小凜計算的  $a_0$  確實會召喚  $M_S$ ，則輸出 `saber`，否則輸出 `archer`。

## 資料範圍

$0 < N \leq 1000$ 、 $0 \leq h < 24$ 、 $0 \leq m < 60$

對於  $0 \leq i \leq N^2$  有  $0 \leq a_i < 2^{32} - 1$

單次輸入資料最大約有 10 MB，請注意讀取資料的速度

## 資料範例

### 輸入範例 1

```
3 20 35
1306187360 3049283999
3209963332 254862218 3048104280
1274204117 3615761292 3262155383
179796570 3730562294 4060963170
```

### 輸出範例 1

saber

### 輸入範例 2

```
3 20 35
1494103018 2762772055
1891146886 258868878 4079655926
3901958345 277932297 2608173749
550833852 303997258 2699905984
```

### 輸出範例 2

archer

### 輸入範例 3

```
3 0 0
3293365268 4225411422
718078474 2092981540 2537296694
786859696 4079735015 4275252306
2540255246 2623462854 1975811358
```

### 輸出範例 3

saber

## 範例解釋

如果  $N$  是一個大小為  $a \times b$  的矩陣，而  $M$  是一個大小為  $b \times c$  的矩陣，則  $S = N \times M$  是一個大小為  $a \times c$  的矩陣，如果把矩陣  $S$  中第  $i$  排第  $j$  列稱呼為  $S_{i,j}$ ，則

$$S_{i,j} = \sum_{x=1}^b N_{i,x} \times M_{x,j} = N_{i,1} \times M_{1,j} + N_{i,2} \times M_{2,j} + \cdots + N_{i,b} \times M_{b,j}$$

以範例 1 來說，兩手的魔力值矩陣分別為

$$\begin{bmatrix} 29534291 & 1382094538 & 2458411381 \\ 121980484 & 3498296423 & 4238353230 \\ 2001268137 & 3157315688 & 2562493883 \end{bmatrix} \text{ 與 } \begin{bmatrix} 3768984614 & 3777567137 & 400103680 \\ 2735851123 & 2298391146 & 2757426325 \\ 3824489188 & 997631111 & 4279455982 \end{bmatrix}$$

兩者相乘，即為範例 1 給定的靈基矩陣，第二排第三個元素 3262155383 就等於

$$121980484 \times 400103680 + 3498296423 \times 2757426325 + 4238353230 \times 4279455982$$

的答案計算完後，取其除以  $2^{32}$  的餘數。

# 問題 18 – 小麥畝產一千八 (18\_Jinkela)

(25 分)

## 問題敘述

日月卦長是小麥農場的主人，為了種植出優良的小麥每天勤奮的在田裡撒金坷垃。可以把卦長的田想成是一個x,y 座標範圍是 $-10^9 \sim 10^9$ 的平面，卦長會選擇一些四邊平行座標軸的矩形區域撒金坷垃，這些矩形區域可能會重疊，例如圖 1 中塗色區域三個矩形  $q_1, q_2, q_3$ 。

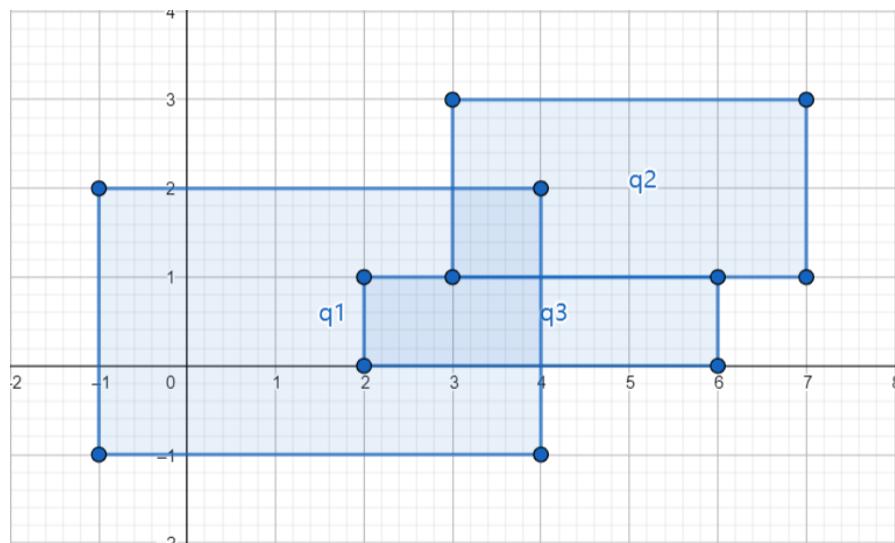


圖 1

但是撒了金坷垃的區域小麥會畝產一千八，大量的害蟲會從沒有金坷垃的區域往有金坷垃的區域移動，因此卦長想要請你幫忙噴灑最強殺蟲劑 - “四大名捕”。但是撒了金坷垃的區域如果噴了殺蟲劑那小麥很容易就會過於肥大而爆炸，因此噴灑殺蟲劑的地方只能在有撒金坷垃和沒撒金坷垃區域的邊界上。

我們可以把邊界分成水平(Horizontal)跟垂直(Vertical)兩種線段所組成，而水平邊界又可以分成下邊界(Down, D)、上邊界(Up, U)；垂直邊界又可以分成左邊界(Left, L)、右邊界(Right, R)。請參考以下圖例，

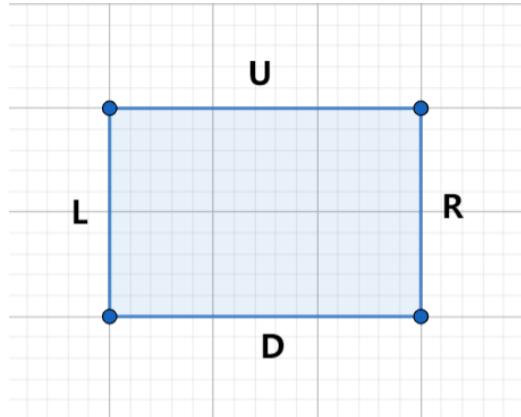


圖 2：四種邊界線

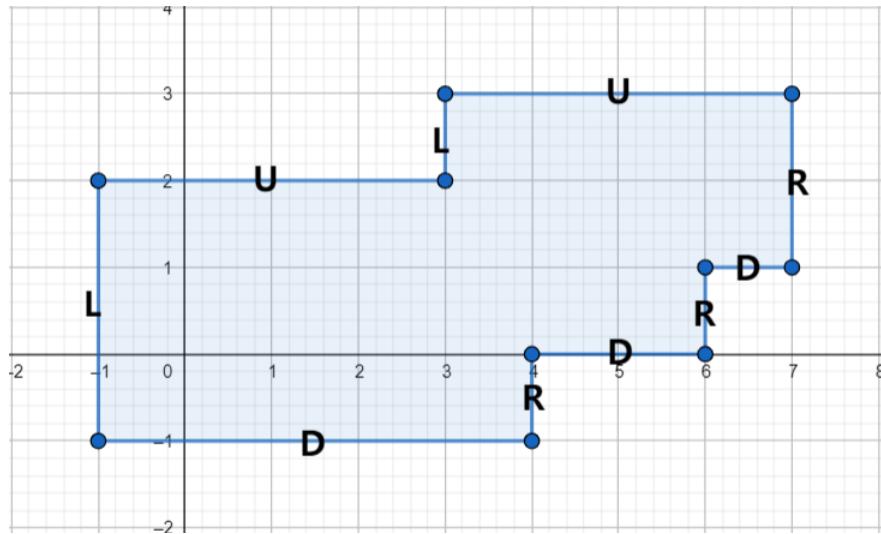


圖 3：圖 1 中的所有邊界線

由於”四大名捕”有四罐，必須分別噴灑在這四種邊界上才會有殺蟲效果，為了每個月都能領到足夠生活的氮磷鉀，你必須在卦長的田中將所有四種邊界找出來並噴上對應的殺蟲劑。

因為矩形太多，邊界太多，所謂上下左右邊界都是相對位置，並非絕對的上方或下方。例如，左側矩形可能和右側矩形在沒有重疊的情況下，左側矩形的右邊界，可能是在右側矩形的左邊。為免混淆，特將邊界定義如下：

1. 下邊界：為一條水平線  $\overline{L(x_1, y) R(x_2, y)}$ ,  $x_1 < x_2$  滿足：對於任意點  $P(x, y)$ ,  $x_1 < x < x_2$ ，存在兩個點  $A(x, y + e)$ 、 $B(x, y - e)$ ,  $e > 0$  使得線段  $\overline{AP}$  會完全落在有撒金坷垃的區域中、線段  $\overline{PB}$  會完全落在沒撒金坷垃的區域中。並且水平線  $\overline{L'(x_1 - f, y) R'(x_2 + g, y)}$ ,  $f, g > 0$  不滿足上面條件
2. 上邊界：為一條水平線  $\overline{L(x_1, y) R(x_2, y)}$ ,  $x_1 < x_2$  滿足：對於任意點  $P(x, y)$ ,  $x_1 < x < x_2$ ，存在兩個點  $A(x, y + e)$ 、 $B(x, y - e)$ ,  $e > 0$  使得線段  $\overline{AP}$  會完全落在沒撒金坷垃的區域中、線段  $\overline{PB}$  會完全落在有撒金坷垃的區域中。並且水平線  $\overline{L'(x_1 - f, y) R'(x_2 + g, y)}$ ,  $f, g > 0$  不滿足上面條件
3. 左邊界：為一條垂直線  $\overline{D(x, y_1) U(x, y_2)}$ ,  $y_1 < y_2$  滿足：對於任意點  $P(x, y)$ ,  $y_1 < y < y_2$ ，存在兩個點  $A(x, y + e)$ 、 $B(x, y - e)$ ,  $e > 0$  使得線段  $\overline{AP}$  會完全落在有撒金坷垃的區域中、線段  $\overline{PB}$  會完全落在沒撒金坷垃的區域中。並且垂直線  $\overline{D'(x, y_1 - f) U'(x, y_2 + g)}$ ,  $f, g > 0$  不滿足上面條件
4. 右邊界：為一條垂直線  $\overline{D(x, y_1) U(x, y_2)}$ ,  $y_1 < y_2$  滿足：對於任意點  $P(x, y)$ ,  $y_1 < y < y_2$ ，存在兩個點  $A(x, y + e)$ 、 $B(x, y - e)$ ,  $e > 0$  使得線段  $\overline{AP}$  會完全落在沒撒金坷垃的區域中、線段  $\overline{PB}$  會完全落在有撒金坷垃的區域中。並且垂直線  $\overline{D'(x, y_1 - f) U'(x, y_2 + g)}$ ,  $f, g > 0$  不滿足上面條件

## 輸入格式

第一行有一個正整數  $T$ ，表示接下來有  $T$  個 case(測試資料)，case 編號由  $1 \sim T$ ，每個 case 的第一行有一個正整數  $n$ ，表示在田中撒金坷垃的區域數量，接著會有  $n$  行，每行有四個整數  $x_1, y_1, x_2, y_2$ ，表示每次撒金坷垃的矩形區域其左下點  $(x_1, y_1)$  以及右上點  $(x_2, y_2)$  的座標。

## 輸出格式

對於每個 case，首先請輸出 Case  $t$ :然後換行， $t$  表示該 case 的編號。

接下來依序輸出四行，每行分別表示如下：

D: D\_size Dlen

U: U\_size Ulen

L: L\_size Llen

R: R\_size Rlen

$D_{size}$  為下邊界(Down, D)有幾段， $D_{len}$  為所有下邊界的長度總和。

$U_{size}$  為上邊界(Up, U)有幾段， $U_{len}$  為所有上邊界的長度總和。

$L_{size}$  為左邊界(Left, L)有幾段， $L_{len}$  為所有左邊界的長度總和。

$R_{size}$  為右邊界(Right, R)有幾段， $R_{len}$  為所有右邊界的長度總和。

## 資料範圍

$1 \leq T \leq 5$

$1 \leq n \leq 100000$

$-10^9 \leq x_1, y_1, x_2, y_2 \leq 10^9$

$x_1 < x_2$

$y_1 < y_2$

保證每個 case 邊界線的數量小於 500000

## 資料範例

### 輸入範例 1

```
3
3
-1 -1 4 2
2 0 6 1
3 1 7 3
3
-1 -1 1 1
1 -3 3 -1
-1 1 1 3
4
-2 -2 3 -1
-1 -3 0 2
-2 0 3 1
1 -3 2 2
```

## 輸出範例 1

Case 1:

D: 3 8

U: 2 8

L: 2 4

R: 3 4

Case 2:

D: 2 4

U: 2 4

L: 2 6

R: 2 6

Case 3:

D: 8 8

U: 8 8

L: 8 8

R: 8 8



# 問題 19 - 好ㄏㄢˋ 好ㄏㄢˋ 奇號 (19\_Curiosity)

(25 分)

## 問題敘述

好好奇號（英語：**Curiosity**，念法：ㄏㄢˋ ㄏㄢˋ ㄑㄧˇ ㄏㄢˋ ）是一輛我國太空總署**R**星科學實驗室轄下的**R**星探測車，主要任務是探索**R**星的欸哎山城，為我國太空總署**R**星科學實驗室計劃的一部份。

好好奇號在 2511 年 11 月 26 日標準時間 10:02 於維拉爾納卡角空軍基地進入**R**星科學實驗室太空船，並預計在 2512 年 8 月 6 日協調世界時 05:17 左右於預定地附近著陸。好好奇號距離**R**星還有 56,300 億公里的旅程，**R**星科學實驗室的科學家們目前正在著手規劃好好奇號確切的著陸地點。

好好奇號搭載最先進的 AI 系統，想當然爾會有自己的想法，根據它最新的回傳訊息，好好奇號表示想要在欸哎山城裡面逛透透，但是它不走重複的路，也不走上坡。

請寫一支程式，輸入欸哎山城的地形高度資訊，這支程式要告訴好好奇號，它該在哪裡降落，才能探索最長的路徑。

## 輸入格式

第一行兩個數字**N**和**M**，表示欸哎山城的寬和長。（欸哎山城是長方形的！很棒吧！）

第二行起有**N**行，每行**M**個數字，以空白間隔，第*i*行第*j*個數字**H<sub>ij</sub>**代表格子座標(*i,j*)的地形高度，數字越大代表高度越高。左上角格子座標是(1,1)，右下角格子座標是(N,M)

## 輸出格式

第一行輸出一個數字**L**，代表好好奇號最長可以走多遠(走過幾格)

第二行輸出 2 個數字 *i* 和 *j*，代表好好奇號應該從座標點(*i,j*)這個位置開始它的旅程。如果有解，請選擇 *i\*10+j* 最小的那個。**請注意，好好奇號只能上下或左右走，不能斜著走。**

## 資料範圍

**1 ≤ N, M ≤ 10**

**1 ≤ i ≤ N, 1 ≤ j ≤ M**

**1 ≤ H<sub>ij</sub> ≤ 10**

相同高度的連續區域面積 ≤ 20

## 資料範例

### 輸入範例 1

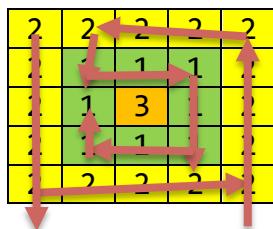
```
5 5
2 2 2 2 2
2 1 1 1 2
2 1 3 1 2
2 1 1 1 2
2 2 2 2 2
```

### 輸出範例 1

```
24
1 1
```

### 範例 1 說明

沒有上坡的前提下，最長可以走 24 格，而最小可能的出發點是(1,1)（雖然從右上角那個格子(1,5)出發，也可以走出 24 格的路徑，但是(1,1)比較小！），你無法找到更長的路徑或者更小的出發點了。（註：出發點比大小的依據是  $i*10+j$ ）



若你從中間那個 3 出發，最遠只能走 9 格，因為你一旦走到高度 1 的點，就爬不上周圍的 2 了。

### 輸入範例 2

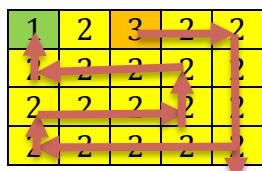
```
4 5
1 2 3 2 2
2 2 2 2 2
2 2 2 2 2
2 2 2 2 2
```

### 輸出範例 2

```
19
1 3
```

### 範例 2 說明

沒有上坡的前提下，最長可以走 19 格，而唯一可能的出發點是(1,3)



賭你不從 3 出發就走不到 19 格  
再加碼！

賭你不走 1 就走不到 19 格



### 輸入範例 3

```
8 8
1 1 1 1 2 2 2 2
1 1 1 1 2 2 2 2
1 1 1 1 2 2 2 2
1 1 1 1 2 2 2 2
3 3 3 3 4 4 4 4
3 3 3 3 4 4 4 4
3 3 3 3 4 4 4 4
3 3 3 3 4 4 4 4
```

### 輸出範例 3

```
48
5 5
```

### 範例 3 說明

不玩了，自己畫畫看，然後跟評審對答案！