

# 2017 網際網路程式設計全國大賽

## 高中組決賽

- 本次比賽共 8 題，含本封面共 24 頁。
- 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，以題目敘述為主。
- 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，英文字母大小寫不同或有多餘字元皆視為答題錯誤。
- 比賽中上傳之程式碼，使用 C 語言請用 `.c` 為副檔名；使用 C++ 語言則用 `.cpp` 為副檔名。
- 使用 `cin` 輸入速度遠慢於 `scanf` 輸入，若使用需自行承擔 Time Limit Exceeded 的風險。
- 部分題目有浮點數輸出，會採容許部分誤差的方式進行評測。一般來說「相對或絕對誤差小於  $\epsilon$  皆視為正確」， $\epsilon$  值以題目敘述為主。  
舉例來說，假設  $\epsilon = 10^{-6}$  且  $a$  是正確答案， $b$  是你的答案，如果符合  $\frac{|a-b|}{\max(|a|,|b|,1)} \leq 10^{-6}$ ，就會被評測程式視為正確。

# 2017 網際網路程式設計全國大賽

## 輸入輸出範例

C 程式範例：

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

C++ 程式範例：

```
1 #include <iostream>
2 int main()
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 }
```

# A. A + B Problem <sup>-1</sup>

Problem ID: aplusb

還記得瀚瀚小時候想搶首殺的故事嗎？沒想到，今年 NPSC 初賽第一題就寫著大大的「A + B Problem」，瀚瀚又熱血沸騰的想要搶這場比賽的首殺！

如果你沒參加今年 NPSC 初賽的話，以下是今年 NPSC 初賽第一題「A + B Problem」的題目：

A + B Problem 往往是大家在各 Online Judge 通過的第一道題目，剛學會寫程式的繪里也不例外。然而，單純地讀入兩個數字  $A, B$  並輸出它們的總和  $C$ ，對於聰明伶俐又可愛的繪里來說實在太簡單了。富有挑戰精神的她忍不住想：「假如只有  $A, B, C$  三個數字視為字串接起來後的結果  $s$ ，我有辦法還原出當初的數字  $A, B, C$ ，並滿足  $A + B = C$  嗎？」

舉例來說，如果  $s = "123"$ ，那麼可以還原出  $1 + 2 = 3$ 。注意冗餘的前導零或空字串是不被允許的，如  $s = "1023"$ ，則不能被還原成  $1 + 02 = 3$ 。

瀚瀚很開心的在一分鐘內把程式碼敲完，準備上傳搶首殺。但瀚瀚突然想起小時候爲了搶首殺，卻不小心敲錯鍵的慘痛經驗，他決定這次要好好的測試他寫完的程式再上傳。因此，瀚瀚想找出  $N$  組有  $K$  種還原方式的輸入字串  $S$ 。但瀚瀚剛寫完一份程式，已經精疲力盡。身爲他的隊友，你，便自告奮勇的來幫他找出  $N$  組有  $K$  種還原方式的輸入！

## Input

測試資料第一行包含兩個整數  $K, N$ ，代表瀚瀚想找出  $N$  組有  $K$  種還原方式的輸入字串。

- $0 \leq K \leq 2$
- $1 \leq N \leq 100$

## Output

請輸出  $N$  行，每行包含一個字串  $S_i$ ，代表  $S_i$  有  $K$  種還原的方法。 $S_i$  必須滿足以下的條件：

- $1 \leq |S_i| \leq 30$
- $S_i$  由數字 0123456789 組成
- $S_i \neq S_j \forall i \neq j$ ，任兩個輸出字串都不相等
- $S_i$  必須恰有  $K$  種還原的方法

如果有超過  $N$  個滿足條件的  $S_i$ ，請輸出任意  $N$  個。輸出的  $N$  個字串可以按照任意順序輸出。

Sample Input 1	Sample Output 1
0 1	1023

Sample Input 2	Sample Output 2
1 2	123 101

Sample Input 3	Sample Output 3
2 1	11111122

## B. 魔法鏈

Problem ID: gem

NPSC 魔法學院在今年正式成立囉！

從小就展現過人天賦的天才兒童殿王，如今已成長茁壯並成為 NPSC 魔法學院第一屆的學員了。一年級的課程對於殿王來講根本就是小菜一碟，所以他跟著學院裡的教授一起衝鋒陷陣，研究最新穎的魔法技術：魔法鏈。

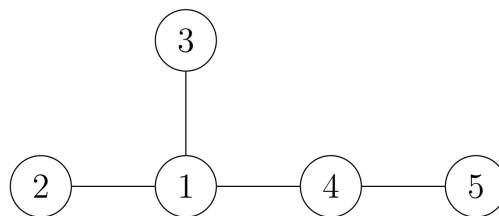
施展這種法術前，殿王要先擁有一串由魔法寶石組成的魔力串珠。一串魔力串珠會由  $N$  顆魔法寶石組成，每顆魔法寶石都是相異的，用數字 1 到  $N$  表示。一串魔力串珠中還會有  $N - 1$  條線，每條線連接兩顆相異的魔法寶石，使得這  $N$  顆魔法寶石都會被串在一起（任兩顆魔法寶石都會透過一或多條線的串接而連接著）。

這項法術的施展，就是利用這串魔力串珠來製造一條魔法鏈。

製造魔法鏈時，若這串魔力串珠只有 2 顆魔法寶石，就直接用魔力剪刀剪斷中間唯一的那條線，並且把這 2 顆魔法寶石依任意順序放在魔法鏈的第 1 跟第 2 個位置。

否則，殿王要挑選魔力串珠中的一條線並用魔力剪刀把它剪斷，使得在剪斷之後會形成一串有多顆魔法寶石的魔力串珠以及一顆獨立的魔法寶石（如果剪斷後不會變成這種結果，那麼殿王就不能剪這條線），接著把這獨立的魔法寶石放在魔法鏈的下一個位置（剪第一刀時得到的魔法寶石放在第一個位置、剪第二刀時得到的魔法寶石放在第二個位置，依此類推），重複上述動作直到這串魔力串珠只剩下兩顆魔法寶石（中間由一條線連接著）時，殿王就把最後這條線剪斷，並依他的選擇將最後的兩顆魔法寶石放在魔法鏈的第  $N - 1$  跟第  $N$  個位置。

不難發現，就算是同樣的魔力串珠，也可以藉由不同的剪取順序而排出許多不同的魔法鏈，以下圖的魔力串珠（與範例一相同）為例，2, 3, 5, 1, 4、5, 2, 4, 1, 3 跟 5, 2, 4, 3, 1 都是可能創造出的魔法鏈，不過 2, 1, 3, 4, 5 跟 2, 3, 4, 5, 1 就是不可能創造出來的魔法鏈。



現在給你一串魔法串珠，請你計算總共可以創造出多少種不同的魔法鏈。兩個魔法鏈只要任一個對應的位置不相等，就視為不同。因為答案可能很大，你的程式只需要輸出答案除以  $10^9 + 7$  的餘數即可。

## Input

測試資料第一行包含一個正整數  $N$ ，表示這串魔法串珠是由幾顆魔法寶石所組成。測試資料接下來包含  $N - 1$  行，每行包含兩個正整數  $a_i, b_i$ ，表示第  $i$  條線連接了魔法寶石  $a_i$  跟魔法寶石  $b_i$ 。

- $2 \leq N \leq 2000$
- $1 \leq a_i, b_i \leq N$
- 輸入的魔法串珠一定會讓  $N$  顆魔法寶石都被串在一起

## Output

輸出一行，包含一個整數，表示該魔法串珠所能創造出的魔法鏈種類數除以  $10^9 + 7$  的餘數。

Sample Input 1	Sample Output 1
5 1 2 1 3 1 4 4 5	28

Sample Input 2	Sample Output 2
2 1 2	2

Sample Input 3	Sample Output 3
4 1 2 2 3 3 4	8

## C. 兮陣

Problem ID: matrix

**注意，本題的記憶體限制為 1 MB。**

兮克出生於娑婆世界東勝神洲傲來國通靈山，自開闢以來的仙石孕育而生。但仙石並非毫無來歷，處於十洲三島的祖脈上，其高圍按二十四氣，其上竅孔對應九宮八卦。一天仙石迸裂，產一石卵，經風一吹，化作一兮克，兮克出世後，眼裡冒出兩道神光，射沖斗府，驚動了天上的玉皇大帝。之後因為成功量子開竅，被通靈山諸人拜為「通靈王」。

由於兮克的出生太過驚天動地了，世界上出現了許多奇景，科學家們也從中獲得許多啟發，發現了許多有趣的東西，其中最為著名的即「兮陣」。

兮陣是一個有趣的矩陣，若一個矩陣滿足所有左上至右下對角線上的元素皆相同，我們便

稱他為兮陣。以下的矩陣皆為兮陣。

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 1 & 2 \\ 0 & 4 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 4 & 1 & 2 & 3 & 4 \\ 0 & 4 & 1 & 2 & 3 \end{pmatrix} \quad \begin{pmatrix} 1 & 2 \\ 4 & 1 \\ 0 & 4 \\ 5 & 0 \\ 5 & 5 \end{pmatrix} \quad (3) \quad \begin{pmatrix} 3 & 3 & 3 \\ 3 & 3 & 3 \\ 3 & 3 & 3 \end{pmatrix}$$

以下的矩陣皆不是兮陣。

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 1 & 1 \\ 0 & 4 & 1 \end{pmatrix} \quad \begin{pmatrix} 3 & 2 & 1 & 4 & 5 \\ 4 & 1 & 2 & 3 & 4 \\ 0 & 4 & 1 & 2 & 3 \end{pmatrix} \quad \begin{pmatrix} 1 & 2 \\ 2 & 3 \\ 3 & 4 \\ 4 & 0 \\ 0 & 5 \end{pmatrix} \quad \begin{pmatrix} 1 & 2 & 3 & 2 & 5 \\ 2 & 3 & 2 & 5 & 2 \\ 3 & 2 & 5 & 2 & 0 \end{pmatrix}$$

兮克在發現自己原來有這麼大的貢獻後，便對兮陣十分有興趣。現在兮克找到了很多矩陣，你能幫他確認這些矩陣是不是兮陣嗎？

## Input

測試資料第一行有一個正整數  $T$  表示今克想測試的矩陣數量。對於每個測試的矩陣，第一行有兩個整數  $N, M$ ，表示該矩陣是一個由  $N \times M$  個元素排列成的矩形陣列。接下來有  $N$  行，每行  $M$  個整數  $A_{i,j}$ ，表示矩陣裡的元素。

- $1 \leq N \times M \leq 10^6$
- $\sum N \times M \leq 10^6$
- $0 \leq A_{i,j} \leq 10^9$

## Output

請輸出  $T$  行，如果第  $i$  個矩陣是幻陣，請輸出 "Yes"，否則請輸出 "No"，皆不包含引號。



Sample Input 1	Sample Output 1
9	Yes
3 3	No
1 2 3	Yes
4 1 2	No
0 4 1	Yes
3 3	No
1 2 3	Yes
4 1 1	Yes
0 4 1	No
3 5	
1 2 3 4 5	
4 1 2 3 4	
0 4 1 2 3	
3 5	
3 2 1 4 5	
4 1 2 3 4	
0 4 1 2 3	
5 2	
1 2	
4 1	
0 4	
5 0	
5 5	
3 5	
1 2 3 2 5	
2 3 2 5 2	
3 2 5 2 0	
1 1	
3	
3 3	
3 3 3	
3 3 3	
3 3 3	
5 2	
1 2	
2 3	
3 4	
4 0	
0 5	

*This page is intentionally left blank.*

## D. 最大不團問題

Problem ID: clique

在圖論領域中，對於一個無向圖而言，滿足兩兩之間有邊連接的頂點集合，被稱為該無向圖的團。團是圖論中的基本概念之一，用在很多數學問題以及圖的構造上。計算機科學中也有對它的研究，儘管在一個圖中尋找給定大小的團達到了 NP 完全的難度，人們還是研究過很多尋找團的算法。

即使尋找團的問題是如此的困難，但傳說中，某個小島舉行的 ACM-ICPC 區域賽卻常常有最大團問題。即給定一張無向圖，找出一個頂點數最多的團。

身為一個專業的競賽選手，艾迪準備了最大團的 codebook，以應付某小島賽區的題目。但今年該賽區居然不出最大團問題了，反而出了一道**最大不團問題**！即找一個最大的頂點集合，使得他並不是一個團。

由於艾迪已經驚慌失措了，你能夠幫助他解決這個問題嗎？

### Input

測試資料第一行有兩個數字  $N, M$ ，分別表示圖的頂點個數以及邊數。

接下來  $M$  行，每行包含兩個數字  $u_i, v_i$ ，表示  $u_i$  與  $v_i$  之間有一條邊。

- $1 \leq N \leq 10^5$
- $0 \leq M \leq 10^5$
- $1 \leq u_i < v_i \leq N$
- 不會有重邊，即任兩個頂點之間至多只有一條邊連接

### Output

輸出一個數字於一行，表示最大不團的頂點個數。

**Sample Input 1**

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

**Sample Output 1**

```
0
```

**Sample Input 2**

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

**Sample Output 2**

```
4
```

## E. When in doubt, C4!

Problem ID: bomb

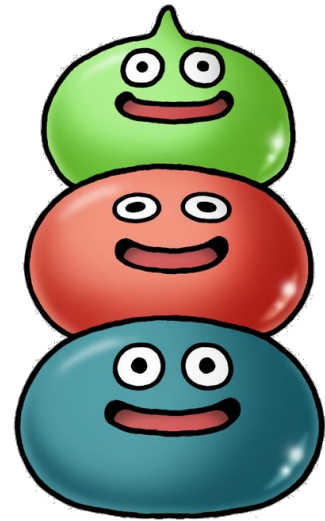
「不管遇上什麼問題，一顆炸彈總能解決你的困擾——如果不行就兩顆。解決不了問題那是你當量不夠，這是普世真理。」

當今的帝國審查官兼諸界巡視者暨煉獄星領主——郝仁，在去找女神渡鴉 12345 蹭飯的路上遇到一群不長眼的史萊姆擋路，江湖人稱炸彈超仁的他決定扔炸彈開路。

擋路的史萊姆們巧妙地排成了一個  $n \times m$  的格狀陣型。其中郝仁視角中左上角那格的座標為  $(1, 1)$ ，右下角為  $(n, m)$ 。每格中可能有若干隻史萊姆直直向上疊起來，也有可能一格內所有史萊姆都被嚇跑了，導致該格一隻史萊姆也沒有。

郝仁打算依序扔  $q$  顆炸彈來趕走史萊姆。第  $i$  顆炸彈會落於座標  $(x_i, y_i)$  的格子正中央，其爆炸半徑為  $r_i$ 。而當第  $i$  顆炸彈落下後，所有座標  $(x, y)$  滿足  $(x - x_i)^2 + (y - y_i)^2 \leq r_i^2$  的非空格子中，最上面的那隻史萊姆會被炸彈嚇跑。

身為一個專業的炸彈仁，忠實紀錄每顆炸彈的威力是他的職責所在。對於每顆炸彈，郝仁想要知道它嚇跑了幾隻史萊姆。但是史萊姆實在太多了，希望你可以寫個程式幫幫他。



### Input

測試資料第一行有三個整數  $n, m, q$ ，分別代表史萊姆陣型的長與寬，以及炸彈的數量。接下來  $n$  行，每行有  $m$  個字元，依序代表左上到右下每格包含的史萊姆數量。為了節省空間，郝人偷懶的使用了縮寫來記錄史萊姆數量。其中字元 0 到 9 代表 0 到 9 隻史萊姆、字元 a 到 z 代表 10 到 35 隻史萊姆、字元 A 到 Z 則代表 36 到 61 隻史萊姆。接下來  $q$  行，每行有三個數字  $x_i, y_i, r_i$ ，依序代表每顆炸彈的落點座標以及爆炸半徑。

- $1 \leq n, m \leq 1000$
- $1 \leq q \leq 100000$
- $1 \leq x_i \leq n$
- $1 \leq y_i \leq m$
- $1 \leq r_i \leq 5000$
- 每格中的史萊姆數量不超過 61 隻

## Output

請輸出  $q$  行，每行包含一個整數。第  $i$  行中的數字表示第  $i$  顆炸彈嚇跑了幾隻史萊姆。

### Sample Input 1

```
2 3 3
240
222
1 1 2
2 3 2
1 2 3
```

### Sample Output 1

```
4
4
3
```

### Sample Input 2

```
2 4 10
np5c
6es7
1 3 0
2 1 1
1 3 1
2 4 2
1 1 4
2 3 3
2 3 1
1 4 0
1 3 0
1 4 4
```

### Sample Output 2

```
1
3
4
5
8
8
3
1
0
7
```

## F. 輸贏吧！在序列征戰中

Problem ID: sequence

「想成為序列操作之神嗎？在序列的世界輸贏一番吧！」

埃迪的意識漸漸清楚，思考著剛剛所聽到的話的同時，逐漸張開眼睛注視著四周，發現了自己身處於一個未知的世界之中，並且旁邊躺著一位蘿莉。

突然間，蘿莉說話了：「來自異世界的勇者呀！請你拯救這個世界，戰勝這個世界的神—胖胖兮吧！」

「這種小說才會發生的事情，居然發生在自己的身上了！我果然有主角光環！」埃迪心想。

這是一個充滿著序列的世界，序列征戰的排行將決定在這個世界的地位，弱肉強食的情況也隨處可見。想要提升排行，必須透過序列征戰系統與他人對決。而序列征戰排行第一的人，也就是這個世界的神，可以改變世界一切法則。

充滿著正義感的埃迪想要改變這個世界，於是他決定挑戰序列操作之神—胖胖兮。

見到胖胖兮後，埃迪發現了，原來召喚埃迪到這個世界的人，不是他人，正是這個世界的神—胖胖兮。胖胖兮長年以來一直在等待擁有強大序列實力的人來挑戰他，然而等了 5487 年，還是等待不到，因此他決定用召喚的方式，找異世界的強者輸贏，而召喚出來的這位強者就是埃迪。

「你想成為序列操作之神嗎？來輸贏吧！來自異世界的強者！」

在序列征戰中，系統將會隨機產生一道跟序列相關的題目，先解出該題的人即為贏家。這場異世界的頂尖對決終於要開始了，究竟序列征戰系統會出現怎麼樣的題目呢？

給定四個整數  $N, Q, C, P$  以及一個長度為  $N$  的序列  $A_1, A_2, \dots, A_N$ ，序列中每個數值皆介於  $[0, 10^6]$ ，接下來有  $Q$  次操作，每次操作形式為以下三種中的一種：

- 1  $l\ r$ : 代表要把序列中第  $l$  項至第  $r$  項的每個數字  $A_i$  變成  $C^{A_i}$  ( $l \leq i \leq r$ )
- 2  $l\ r$ : 代表要詢問  $(A_l + A_{l+1} + \dots + A_r) \bmod P$  的結果
- 3  $p\ v$ : 代表要把序列的第  $p$  項數字變成  $v$ ，即  $A_p = v$

埃迪看到這道題目後整個驚呆了，完全不知從何著手，可以請你幫幫埃迪，讓他可以戰勝這場序列征戰，改變這個世界嗎？

## Input

以下所有符號意義皆與序列征戰系統題目中的意義相同。

測試資料第一行包含四個正整數  $N, Q, C, P$ 。

接下來的一行，有  $N$  個整數，依序為序列中的每一項數值  $A_1, A_2, \dots, A_N$ 。

接下來  $Q$  行，每行表示一個操作「1 l r」、「2 l r」或「3 p v」。

- $1 \leq N \leq 10^5$
- $1 \leq Q \leq 5 \times 10^4$
- $1 \leq C \leq 10^6$
- $1 \leq P \leq 10^3$
- $0 \leq A_i \leq 10^6$
- $1 \leq l \leq r \leq N$
- $1 \leq p \leq N$
- $0 \leq v \leq 10^6$

## Output

對於每個第 2 種操作，輸出一行，表示該次詢問的結果。

Sample Input 1	Sample Output 1
5 7 2 3	1
0 1 2 3 4	1
2 1 5	0
1 1 5	0
2 1 5	
1 1 5	
2 1 5	
3 1 5	
2 1 5	



<b>Sample Input 2</b>	<b>Sample Output 2</b>
3 4 10 13 9 17 33 1 2 3 2 1 3 3 1 8 2 1 3	12 11

*This page is intentionally left blank.*

## G. 最大不連續和問題

Problem ID: sum

在數學中，某個數列的子數列是從最初數列通過去除某些元素，但不破壞餘下元素的相對位置（在前或在後）而形成的新數列。例如， $1, 3, 4$  即是  $1, 2, 3, 4, 5$  的一個子數列。

最大連續和問題是一個子數列相關的經典問題，目標是在數列中找到一個連續的子數列，使該子數列的和最大。例如，對一個數列  $-2, 1, -3, 4, -1, 2, 1, -5, 4$ ，其連續子數列中和最大的是  $4, -1, 2, 1$  其和為 6。

在某個小島舉行的 ACM-ICPC 區域賽中，常常會有許多經典問題。身為一個專業的競賽選手，艾迪十分熟練各種經典題的做法，因此也常常在該賽區神速般獲得各種 **Accepted**。

但就在這次競賽中，艾迪稍微遇到了小波折。這次艾迪遇到的問題已經不是經典題了，而是只有一字之差的**最大不連續和問題**！即要在數列中找到一個**不連續的子數列**，使得該子數列的總和最大。

例如，對一個數列  $1, 1, 1, -1, -3, 10$ ，其不連續的子數列中和最大的是  $1, 1, 1, 10$  其和為 13。

由於艾迪已經驚慌失措了，你能夠幫助他解決這個問題嗎？

### Input

測試資料第一行有一個數字  $N$ ，表示數列的長度。

第二行有  $N$  個整數  $A_1, A_2, \dots, A_N$ ，表示數列。

- $3 \leq N \leq 10^6$
- $|A_i| \leq 10^9$

### Output

輸出一個數字於一行，表示最大不連續和的值。

**Sample Input 1**

3 -1 -2 -3
---------------

**Sample Output 1**

-4
----

**Sample Input 2**

6 1 1 1 -1 -3 10
---------------------

**Sample Output 2**

13
----

**Sample Input 3**

10 9 4 3 8 1 5 10 7 2 6
----------------------------

**Sample Output 3**

54
----

## H. 決賽時你在做什麼呢？

Problem ID: painting

「決賽時你在做什麼呢？」

眾所皆知，全國網際網路程式設計大賽（NPSC）的決賽會場會有很多好吃的點心，瑤瑤是一位很會畫畫且喜歡吃點心的可愛高中生，一直很希望能夠吃到 NPSC 的點心，而夜夜為了獲得瑤瑤的歡心，便努力地練習程式，經過日以繼夜的苦練，終於順利取得了晉級決賽的資格，並於今天參加了 NPSC 高中組的決賽。

俗話說得好，「眾人皆醉我獨醒，舉世皆濁我獨清」。此時 NPSC 的決賽正在進行中，所有選手都在奮力寫 code、思考題目的同時，有一名少女也在思考，雖然同樣是思考，不過她腦海裡浮現的內容卻與其他參賽者所浮現的有些不一樣。

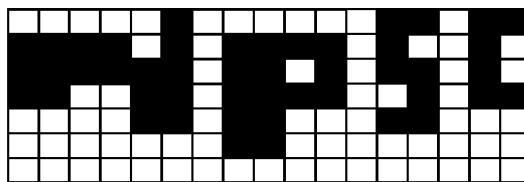
「有沒有空？」

瑤瑤想畫畫，雖然瑤瑤對程式設計一竅不通，但她仍然想用自己所擅長的方式—畫畫，來為夜夜加油打氣。

現在她手邊有一張神奇的圖畫紙，由長、寬分別為  $W$  和  $H$  單位長度所構成的矩形方格圖。（為了方便大家理解，瑤瑤定義「長為平行  $x$  軸方向、寬為平行  $y$  軸方向」）

圖畫紙的每個格子都有一個數值，瑤瑤想「從左至右」在這圖畫紙中畫出 "NPSC"（不含引號）這四個字，並且要求滿足一件事情—「任意兩字不能連起來」（即假設四個字的最左邊及最右邊的  $x$  座標分別為  $L_N, R_N, L_P, R_P, L_S, R_S, L_C, R_C$ ，那麼必須有  $R_N + 1 < L_P$ 、 $R_P + 1 < L_S$ 、 $R_S + 1 < L_C$ ）

以下為在  $W = 17, H = 7$  的神奇圖畫紙上繪製 N、P、S、C 四字的一種方式：



這幅寫著 NPSC 四個大字的繪畫看似普通，但其實擁有著顛覆眾人認知的力量，瑤瑤在圖畫紙上所畫的 N、P、S、C 四字覆蓋之格子數值總和越大，夜夜將會獲得越多的加持，甚至還有機會獲得夢幻的電神之力壓倒性地電全場呢！現在瑤瑤想知道她所畫的 NPSC 四字所覆蓋的格子數值總和最大可以是多少呢？

「可以來畫畫嗎？」

如果天真地以為畫在圖畫紙上的 N、P、S、C 四字可以隨便亂畫的話，那麼就大錯特錯啦！

對於 N、P、S、C 這四個字，在神奇圖畫紙上繪畫有些嚴格的定義，如果沒有按照定義好好的畫，夜夜不但獲得不了神秘力量的加持，反而會獲得負面效果！

為了避免這種悲慘的事情發生，瑤瑤早已有所準備。她在決賽前就已經整理好了一本 codebook，裡面收錄著各國字符在神奇圖畫紙上的嚴格繪畫定義。瑤瑤帶著這本 codebook 參加決賽，很快地就在這本 codebook 中找到了 N、P、S、C 四個字畫在神奇圖畫紙上的圖形定義，內容如下：

### 1. N 的定義

N 由至少三個矩形所組成（矩形四邊皆平行於座標軸），假設由  $K$  個矩形組成（編號 1 到  $K$ ），第  $i$  個矩形的左下角方格座標為  $(L_i, B_i)$ ，右上角方格座標為  $(R_i, T_i)$ ，要求滿足：

- $L_i \leq R_i$ 、 $B_i \leq T_i$
- 對於任意  $1 < i \leq K$ ，有  $L_i = R_{i-1} + 1$
- 對於任意  $3 \leq i < K$ ，有  $B_{i-1} - 1 \leq T_i \leq T_{i-1}$ 、 $B_i \leq B_{i-1}$
- $B_2 > B_1$ 、 $T_2 = T_1$ 、 $B_{K-1} = B_K$ 、 $T_{K-1} < T_K$

### 2. P 的定義

P 由兩個部分組成，第一個部分為一個平行於座標軸的矩形 X 構成，第二個部分為一個平行於座標軸的大矩形 Y 挖去一個平行於座標軸的小矩形 Z 構成，設三個矩形的左下角方格座標分別為  $(L_X, B_X)$ 、 $(L_Y, B_Y)$ 、 $(L_Z, B_Z)$ ，右上角方格座標分別為  $(R_X, T_X)$ 、 $(R_Y, T_Y)$ 、 $(R_Z, T_Z)$ ，要求滿足：

- $L_X \leq R_X$ 、 $B_X + 3 \leq T_X$
- $L_Y = R_X + 1$ 、 $B_Y > B_X$ 、 $T_Y = T_X$
- $L_Y + 1 \leq R_Y$ 、 $B_Y + 2 \leq T_Y$
- $L_Z = L_Y$ 、 $B_Z = B_Y + 1$ 、 $R_Z = R_Y - 1$ 、 $T_Z = T_Y - 1$

### 3. S 的定義

S 由一個大矩形 X，挖去兩個小矩形 Y、Z 得到，這三個矩形的邊都平行於座標軸。設大矩形 X 左下角方格座標為  $(L, B)$ 、長為  $U$  ( $U \geq 2$ ) 寬為  $V$  ( $V \geq 5$  且  $V$  為奇數)，則小矩形 Y 滿足左下角方格座標為  $(L, B + 1)$ 、長為  $U - 1$ 、寬為  $\frac{V-1}{2} - 1$ ，小矩形 Z 滿足左下角方格座標為  $(L + 1, B + \frac{V+1}{2})$ 、長為  $U - 1$ 、寬為  $\frac{V-1}{2} - 1$ 。

### 4. C 的定義

C 由一個大矩形 X，挖去一個小矩形 Y 得到，這兩個矩形的邊都平行於座標軸。設大矩形

X 的左下角方格座標為  $(L, B)$ ，長為  $U$  ( $U \geq 2$ )、寬為  $V$  ( $V \geq 3$ )，則小矩形 Y 滿足左下角方格座標為  $(L + 1, B + 1)$ ，長為  $U - 1$ 、寬為  $V - 2$ 。

現在，定義的部分瑤瑤都已經明白了，那麼讓我們開始畫畫吧！Let's draw！

## Input

測試資料第一行包含兩個正整數  $W, H$ ，代表這張神奇圖畫紙是長為  $W$ 、寬為  $H$  的矩形方格圖。

接下來有  $H$  行，每行有  $W$  個整數，輸入資料中的第  $j + 1$  行的第  $i$  個整數為格子座標  $(i, j)$  的數值  $A_{i,j}$  ( $1 \leq j \leq H, 1 \leq i \leq W$ )

- $1 \leq W \leq 200$
- $1 \leq H \leq 1000$
- $-200 \leq A_{i,j} \leq 200$

## Output

輸出共一行。若瑤瑤無論如何都無法在這張圖畫紙上畫出 N、P、S、C 四個字，則輸出 "Sad"。如果瑤瑤可以在這張圖畫紙上畫出 N、P、S、C 四個字，假設她所能畫出的 N、P、S、C 四字所覆蓋的格子數值總和最大為  $x$ ，若  $x$  為正整數，輸出 "Happy x"，反之則輸出 "Normal x"。輸出皆不含引號。

### Sample Input 1

```
16 5
42 4 22 81 197 25 20 41 194 93 61 134 123 131 97 192
127 110 47 169 21 70 16 179 132 1 120 15 178 83 123 96
179 193 164 154 85 195 116 117 169 88 105 55 61 174 77 152
48 178 153 7 173 109 65 118 116 64 65 10 53 97 136 117
199 94 116 55 168 41 7 66 13 101 113 21 172 96 149 108
```

### Sample Output 1

```
Happy 5759
```

**Sample Input 2**

```

16 5
-32 -146 -88 -109 -102 -157 -87 -86 -149 -5 -81 -30 -43 -1 -190 -89
-190 -35 -85 -192 -191 -133 -180 -184 -70 -164 -187 -121 -50 -81 -194 -114
-72 -101 -109 -108 -38 -37 -180 -196 -160 -63 -32 -172 -146 -116 -83 -4
-143 -103 -72 -21 -114 -44 -3 -63 -83 -28 -57 -60 -5 -16 -99 -60
-189 -28 -95 -178 -69 -110 -80 -86 -42 -39 -111 -149 -187 -66 -162 -46

```

**Sample Output 2**

```

Normal -2080

```

**Sample Input 3**

```

7 5
13 3 100 -75 144 40 4
47 -66 132 147 18 0 140
-16 -17 56 192 181 198 -15
18 157 -126 -109 -98 74 -161
-7 118 -197 161 118 -95 147

```

**Sample Output 3**

```

Sad

```

**Sample Input 4**

```

17 7
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 10 10 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 10 10 -1 10 10 -1 -1 -1 10 10 -1 -1 -1
10 10 -1 -1 10 10 -1 10 10 10 10 -1 -1 10 -1 10 10
10 10 10 10 10 10 -1 10 10 -1 10 -1 10 10 -1 10 -1
10 10 10 10 -1 10 -1 10 10 10 10 -1 10 -1 -1 10 -1
-1 -1 -1 -1 -1 10 -1 -1 -1 -1 -1 -1 10 10 -1 10 10

```

**Sample Output 4**

```

Happy 470

```