

# 2020 網際網路程式設計全國大賽

## 國中組初賽

- 本次比賽共 6 題，含本封面共 14 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「相對或絕對誤差不超過  $\epsilon$  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。

# 2020 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. 方陣

Problem ID: square

小 Y 很喜歡方陣，為此他蒐集了各式各樣的美麗的  $3 \times 3$  方陣，每天都拿出幾個來欣賞。一個方陣如果被小 Y 稱做美麗的，代表它符合以下幾種條件：

- 所有數字皆介於 1~9 之間，數字可以重複。
- 每行、每列、以及兩條對角線上的數字的總和皆要相同。

這天，小 Y 拿出他的一個方陣，卻發現上面缺了一個數字！

「肯定是小 P 搞的鬼！」

小 Y 雖然很生氣，但為了看到美麗的方陣，他必須先把這個方陣修復完成，可是搞不好小 P 故意放了一個不美麗的方陣進來混淆小 Y。你可以告訴他這個方陣是否能被修復成美麗的方陣嗎？

### Input

輸入的共有三行，每行三個字元，代表小 P 搗蛋後， $3 \times 3$  方陣的樣子。該方陣會有恰好一個字元 X，代表這個方陣缺少數字的位置。

- 所有字元皆為 1~9 之間的數字或 X
- X 只會出現恰好一次

### Output

若存在一個數字替換 X 後使得方陣符合小 Y 的要求，則輸出 "Yes"（不含引號），否則輸出 "No"（不含引號）。

<b>Sample Input 1</b>	<b>Sample Output 1</b>
816 35X 492	Yes

<b>Sample Input 2</b>	<b>Sample Output 2</b>
111 1X1 111	Yes

<b>Sample Input 3</b>	<b>Sample Output 3</b>
123 456 7X9	No

## B. 機器人障礙賽

Problem ID: robot

身為 NPSC 的參賽者，懂得操控機器人是一件非常正常的事情，然而想要在看不見機器人的情況下操控機器人就是另一門學問了。

你要操控的機器人現在正在  $n \times m$  的方格中的其中一格，然而你不知道你的機器人目前在哪一格（不會在障礙物上）。

透過某些手段，你獲得了這個方格的樣貌：某些格子是空的，某些格子上有著障礙物。

你可以透過程式給你的機器人一個只含有 "LRUD" 四種字元（不含引號）的行動序列字串，你的機器人將會照著那個字串中的字元依序分別往左、右、上、下走一格。且因為你的機器人很先進，所以如果你叫你的機器人走向有著障礙物的格子，你的機器人將會忽略那個字元。

有趣的是，在這  $n \times m$  的範圍外，有著無限寬廣且沒有障礙物的格子，我們暫且稱之為範圍外的格子。相對的，這  $n \times m$  範圍內的方格我們就叫他範圍內的格子。也就是說，萬一你的機器人走出了範圍內的格子之後，他還是會跟著行動序列字串繼續移動。且在範圍內的空格子中，保證每個格子都可以透過一系列的空格子（包含範圍外的格子）到達，不會有任何格子被障礙物包圍。

請輸出一個移動序列，使得在你不知道起點的情況下，也能走過所有範圍內的格子至少一次。

### Input

輸入的第一行有兩個正整數，分別為  $n$  及  $m$ 。

接下來的  $n$  行，每行有  $m$  個字元，代表地圖本身。如果第  $i$  行的第  $j$  的字元是  $.$ ，代表該格是空地；否則就會是  $\#$ ，代表該格是障礙物。

- $n, m \leq 4$
- $s_{i,j}$  只可能是  $.$ （不含引號）或是  $\#$ （不含引號）
- 保證不會有某個空地被障礙物包起來，也就是保證每個空地都能互相走到
- 範圍內空格的數量大於 1

## Output

請輸出一個字串，字串長度不超過  $5 \times 10^6$ ，且字串只包含 "LRUD" 四種字元（不含引號），代表你給機器人的行動序列。

如果有多組字串滿足條件，輸出任意一組即可。

保證在這個長度限制以內一定找得到一個可以達成目標的字串。

Sample Input 1	Sample Output 1
2 2 .. ..	RDLLUURRRDDLLLLLUUUU
Sample Input 2	Sample Output 2
1 4 .#.#	URRDDLLLLU

## C. NPSC

Problem ID: npsc

一年一度的 NPSC 又到來了！

話說為了方便分辨每年的 NPSC，我們常常會在 NPSC 的名字後面加上那年的年份，例如今年的 NPSC 我們會給他 "NPSC 2020"（不含引號）的標題，而今年的 NPSC 是第 22 屆。

不過主辦單位覺得每 20 屆的 NPSC 都會有種特殊的意義，這時候，主辦單位就會想要使用屆數，而非年份，來給今年的 NPSC 上標題。例如第一次的 20 屆就是前年的第 20 屆 NPSC，而對應的標題就是 "NPSC 20 years"（不含引號）而不是 "NPSC 2018"（不含引號）。而下一次有類似的標題就會是 18 年後的事情了。

你，作為 NPSC 參賽者的一員，覺得每次都要思考某一年的標題是什麼實在是太累了。於是你決定自己寫一個程式幫助自己，只要輸入想要詢問的年份，那個程式就會自動告訴你該年的標題是什麼。詳細的輸入輸出格式請參考以下段落。

### Input

輸入只有一個正整數  $Y$ ，代表年份。

- $2020 \leq Y \leq 2999$

### Output

請輸出一行，代表那年的標題。

如果該年的 NPSC 屆數是 20 的倍數，那麼請輸出 "NPSC x years"（不含引號），其中  $x$  代表屆數。

否則請輸出 "NPSC x"（不含引號），其中  $x$  是年份。

#### Sample Input 1

2020

#### Sample Output 1

NPSC 2020

<b>Sample Input 2</b>	<b>Sample Output 2</b>
2038	NPSC 40 years



## D. 好朋友

Problem ID: graph

請找出滿足以下條件的班級或宣稱這樣的班級不存在：

- 這個班級裡有  $N$  個男生和  $M$  個女生。
- 男生的座號從  $0$  到  $N - 1$ ，女生的座號則是從  $N$  到  $N + M - 1$ 。
- 每個學生在這個班級中都恰有  $x$  個男性朋友和  $y$  個女性朋友。

注意到朋友關係是互相的，即若  $A$  是  $B$  的好友，則  $B$  是  $A$  的好友。

### Input

輸入的第一行有一個正整數  $T$  代表測試資料有幾筆。

每筆測試資料佔一行，此行有用空白隔開的四個非負整數，分別代表一個測試資料的  $N$ ,  $M$ ,  $x$ ,  $y$ 。

- $1 \leq T \leq 500$
- $1 \leq N, M \leq 1000$
- $0 \leq x < N$
- $0 \leq y < M$
- 保證所有測試資料中  $N, M$  的總和不超過 1000

### Output

對於每筆測試資料，若不存在可能的班級，輸出一行 "-1"（不含引號），否則輸出一個整數  $E$  代表這個班級有幾對好朋友，並接著輸出  $E$  行，第  $i$  行有兩個非負整數  $u_i, v_i$ ，代表座號  $u_i$  和  $v_i$  的同學為好朋友。

<b>Sample Input 1</b>	<b>Sample Output 1</b>
3 2 2 1 1 2 3 1 0 1 3 0 0	4 0 1 2 3 0 2 1 3 -1 0

## E. 旅遊

Problem ID: travel

小 N 是一個喜好旅遊的人。他最近發現 NPSC 島十分適合旅遊，因此他決定規劃一個在 NPSC 島旅遊的行程。

經過詳細的調查之後，發現 NPSC 島總共有  $N$  個主要的城鎮，由 0 編號到  $N - 1$ ，並且每個城鎮都有至少一條通往其它城鎮的單向道路，有些城鎮甚至會有超過一條通往同一個城鎮的道路。為了得到最佳的旅遊體驗，小 N 把從每個城鎮出發的所有道路都排好順序。小 N 計畫了  $T$  天的行程，每天遊覽一個城鎮，並且要按照自己排的道路順序輪流選擇隔天的要前往的城鎮。

具體來說，如果小 N 某一天在編號  $x$  的城鎮，且在小 N 的排序之後由該城鎮出發的各個道路分別通往編號  $p_{x,1}, p_{x,2}, \dots, p_{x,M_x}$  的城鎮，那麼若今天是小 N 第一次在這個城鎮，隔天就前往編號  $p_{x,1}$  的城鎮；如果是第二次，就前往編號  $p_{x,2}$  的城鎮，以此類推；如果所有道路都輪過一次，那就從頭再開始輪流。也就是說，如果今天是小 N 第  $i$  次到編號  $N$  的城鎮，那麼他隔天就會前往編號  $p_{x,q+1}$  的城鎮，其中  $q$  是  $i - 1$  除以  $M_x$  的餘數。

然而因為小 N 的行程實在是太長了，所以請你幫小 N 寫一個程式，方便他用自己排的道路順序和第一天所在的城鎮計算出這  $T$  天分別會在哪個城鎮。

### Input

第一行有三個以空白隔開的非負整數  $N, S, T$ ，代表島上的城鎮數量、小 N 第一天所在的城鎮編號與他總共要移動的次數。

接下來有  $N$  行，每行有數個以空白隔開的非負整數，第  $i$  行（從 0 開始編號）代表從第  $i$  個城鎮出發的所有道路。第一個數字  $M_i$  代表從該城鎮出發的道路總數，接下來  $M_i$  個數字依序代表每條道路連向哪一個城鎮，道路的順序即是按照小 N 規劃時決定的順序。

- $0 \leq S < N \leq 10^5$
- $T \leq 10^6$
- $1 \leq M_i$
- $M_0 + M_1 + \dots + M_{N-1} \leq 10^6$

## Output

請輸出  $T$  行，依序代表小  $N$  每天所在的城鎮編號（因此第一行一定是  $S$ ）。

Sample Input 1	Sample Output 1
4 0 9	0
3 1 3 2	1
1 0	0
2 0 0	3
1 0	0
	2
	0
	1
	0

Sample Input 2	Sample Output 2
4 1 8	1
2 2 1	0
2 0 3	2
2 1 0	1
2 2 1	3
	2
	0
	1

## F. 方陣, Again

Problem ID: square2

小 Y 很喜歡方陣，為此他蒐集了各式各樣的美麗的  $n \times n$  方陣，每天都拿出幾個來欣賞。一個方陣如果被小 Y 稱做美麗的，代表它符合以下幾種條件：

- 所有數字皆介於 1~9 之間，數字可以重複。
- 對這個方陣施展一個魔法：對於  $1 \leq i \leq n$ ，把這個方陣從上面數下來第  $i$  個橫列，變成從左邊數來第  $i$  個直行。施展完這個魔法後，新的方陣必須與舊的方陣一模一樣。

以一個  $3 \times 3$  的方陣為例，對以下左圖的方陣施展魔法後，會變成右圖的方陣：

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

從上圖可以看到，這並不是一個美麗的方陣。雖然他滿足第一個條件，但是在施展魔法後，兩個方陣並沒有一樣（只考慮數字，底色僅為示範用，不列入判斷方陣是否相同的考量）。

這天，小 Y 拿出他的一個方陣，卻發現上面缺了一些數字！

「肯定是小 P 搞的鬼！」

小 Y 雖然很生氣，但為了看到美麗的方陣，他必須先把這個方陣修復完成。修復方陣時，他需要在缺少數字的格子裡填入數字讓方陣變得美麗。但是小 Y 有一個小癖好，越小的數字他越喜歡，因此他希望修復完方陣之後，他所填入的**數字總和越小越好**。可是搞不好小 P 故意放了一個不美麗且無法修復的方陣進來混淆小 Y，這時小 Y 不能被小 P 給騙了。

### Input

輸入第一行是一個正整數  $n$ ，代表方陣的大小。接下來的  $n$  行，每行都有  $n$  個數字，用空白隔開，代表這個方陣的一個橫列（順序由上往下）。

- $1 \leq n \leq 2000$
- 小 Y 拿到的方陣中，所有數字都在 1 到  $2^{31} - 1$  這個範圍之中，若出現  $-1$  則代表那一格是缺少數字的

## Output

若這個方陣是可以修復的，請輸出一個整數，代表所需要填入的數字總和的最小值。若這個方陣無法修復，請輸出 "-1"（不含引號）。注意：若方陣本來就是美麗的而不需修復，則數字總和視為 0。

Sample Input 1	Sample Output 1
3 1 2 3 4 5 6 7 8 9	-1

Sample Input 2	Sample Output 2
4 -1 -1 2 3 1 4 9 8 2 -1 -1 -1 -1 8 2 -1	18

Sample Input 3	Sample Output 3
4 -1 2 3 10 -1 5 9 7 2 9 -1 6 -1 7 6 -1	-1