

# 2022 網際網路程式設計全國大賽

## 國中組決賽

- 本次比賽共 7 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。
- 任何題目內提到的「一行」，皆代表以換行字元「`\n`」結尾的字串。
- 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。

Problem	Problem Name	Time Limit	Memory Limit
A	數字換換換	1 s	1024 MB
B	領取香蕉	1 s	1024 MB
C	吃餅乾	1 s	1024 MB
D	賽馬播報	1 s	1024 MB
E	大富翁	1 s	1024 MB
F	拼拼圖	1 s	1024 MB
G	看星星的鄭教授	1 s	1024 MB

# 2022 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. 數字換換換

Problem ID: numbers

身為一個專業的遊戲設計師，小波最近正致力於開發一款名叫「數字換換換」的遊戲。這款遊戲的規則相當簡單，一開始遊戲中會有  $N$  個排成一列的整數，且**每一項都介於 0 到 2 之間**，而遊戲的目標是要讓所有數字都變成 0。為了達成這個目標，玩家可以進行許多次操作，每次操作可以挑選兩個相鄰的數字，並將它們進行以下改變：

- 若這兩個數字都是  $X$ ，則可以選擇將其中一個數字變成  $Y$ ，另一個變成  $Z$ ，其中  $X, Y, Z$  為三個互不相同的數字且  $0 \leq X, Y, Z \leq 2$ 。
- 若這兩個數字不同，則將它們都變成這兩個數字裡另一種沒有出現過的數字。

舉例來說，若現在對兩個分別為 0, 0 的數字進行操作，則經過操作後可以變成 1, 2 或是 2, 1。而若對兩個分別為 0, 1 的數字進行操作，則經過操作後兩數都會變成 2。

然而小波現在遇到了一個問題，他發現遊戲中不是所有的初始數字都可以完成目標，也就是將所有數字都變成 0。像是範例測資二的資料，可以證明無論如何絕對無法達成目標。

因此忙碌的小波找上了你，他給你很多的遊戲初始數字，想請你幫他判斷能不能達成目標。

### Input

首先第一行包含一個正整數  $N$ ，代表遊戲中有多少個數字排成一列。

接著第二行包含  $N$  個整數  $a_1, a_2, \dots, a_N$ ， $a_i$  代表從左邊數過來第  $i$  個數字是什麼。

- $2 \leq N \leq 2 \times 10^5$
- $0 \leq a_i \leq 2$

### Output

請輸出一行 Yes 或 No，代表這個初始數字能不能達成目標，也就是經過一連串的操作將數字全部變成 0。

<b>Sample Input 1</b>	<b>Sample Output 1</b>
4 0 1 2 0	Yes

<b>Sample Input 2</b>	<b>Sample Output 2</b>
2 0 1	No

<b>Sample Input 3</b>	<b>Sample Output 3</b>
6 0 0 0 0 0 0	Yes

## B. 領取香蕉

Problem ID: banana

香蕉，作為一種方便食用又能快速補充能量的食物，和巧克力一樣受到全國爬山大賽 (National Pa Shan Contest，簡稱 NPSC) 選手的歡迎。一般來說 NPSC 主辦單位會在比賽場地的各處設置領取香蕉的地點，讓選手們隨時隨地都可以吃到香蕉。然而，今年因為人力不足，比賽場地裡只有一個領取香蕉的地點，而且毫不意外地永遠都大排長龍。

舊乙作為一個香蕉愛好者，比賽中沒有香蕉吃是絕對不行的。因此，他事先仔細研究了香蕉領取處：香蕉領取處前方是一個巨大的有  $N$  個橫列與  $M$  個直行的棋盤，排隊時，棋盤的每個格子上永遠都會有恰一個人，並且每個格子  $(i, j)$  的地面上都寫了一個數字  $a_{i,j}$ ，代表格子  $(i, j)$  上的人就是隊伍中的第  $a_{i,j}$  個人。格子  $(i, j)$  指的是位於第  $i$  個橫列與第  $j$  個直行的格子。

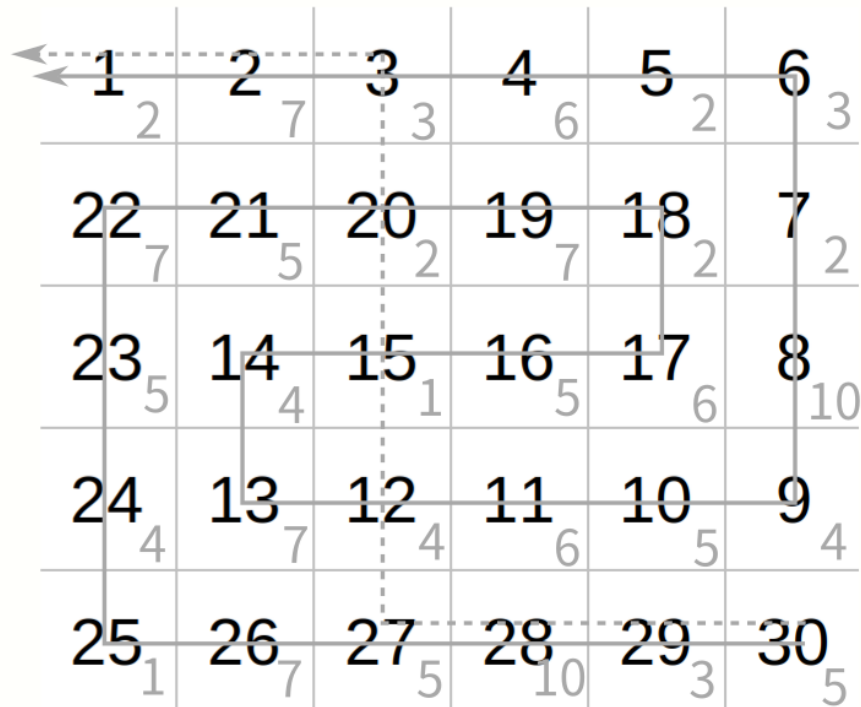
數字差 1 的格子一定是相鄰（共用一條邊）的。香蕉的發放很有效率，每過一秒鐘，在寫著數字 1 的格子上的人就會領到香蕉並離開隊伍，而其他人會移動到下一個格子去，也就是說本來在寫著數字  $x$  的格子上的人，會移動到寫著  $x - 1$  的格子。與此同時，會有一個新的想要香蕉的人來排隊，他會排在寫著數字  $N \times M$  的格子上。寫著數字 1 和數字  $N \times M$  的格子一定會在這個棋盤的邊界，才不會有人需要穿越長長的隊伍。

舊乙現在成功排入了這個隊伍，他目前在寫著數字  $N \times M$  的格子上。照理來說，他要等待  $N \times M$  秒才能拿到香蕉，但他現在迫切地需要香蕉。因此，他決定使出他的絕招——插隊！

假設舊乙現在在格子  $(x, y)$  上，那麼對於和  $(x, y)$  相鄰的格子  $(i, j)$ ，只要滿足  $|a_{x,y} - a_{i,j}| \neq 1$ ，也就是那個格子既不是前一個格子，也不是下一個格子，那麼舊乙就可以偷偷地插隊到格子  $(i, j)$  上，同樣要花費 1 秒鐘。聽起來很美好，但也不是完全沒有代價，這麼做的時候舊乙要承受  $c_{i,j}$  的風險值，如果他在整個排隊的過程中，承受的風險值超過  $B$ ，那麼他就會因為插隊被主辦單位取消比賽資格。

要是被取消資格就得不償失了（他甚至連香蕉也拿不到），請你幫他計算在不會被取消資格的前提下，他至少要花幾秒才能領到香蕉。

舉例來說，當  $N = 5, M = 6, B = 11$  時，下圖是香蕉領取處前的棋盤示意圖。大黑色數字是每個格子上寫的數字，右下角的小灰色數字則是插隊到那個格子需要承受的風險值。灰色實線路線是在完全不插隊的情況下，領取香蕉的正常路線，虛線路線則是一種總風險值為 10 的舊乙可能的移動路線。



## Input

輸入的第一行有三個整數  $N, M, B$ ，表示棋盤的大小與舊乙最大能承受的總風險值。

接下來有  $N$  行，其中第  $i$  行包含  $M$  個正整數  $a_{i,1}, a_{i,2}, \dots, a_{i,M}$ ， $a_{i,j}$  表示格子  $(i, j)$  上寫著的數字。

再來有  $N$  行，其中第  $i$  行包含  $M$  個正整數  $c_{i,1}, c_{i,2}, \dots, c_{i,M}$ ， $c_{i,j}$  表示插隊到格子  $(i, j)$  時舊乙需要承受的風險值。

- $1 \leq N, M \leq 50$
- $1 \leq a_{i,j} \leq N \times M$
- 所有  $a_{i,j}$  兩兩不同
- 滿足  $a_{i,j} = 1$  或  $N \times M$  的格子  $(i, j)$  一定在邊界上
- 對於滿足  $|a_{x_1, y_1} - a_{x_2, y_2}| = 1$  的  $x_1, y_1, x_2, y_2$ ， $|x_1 - x_2| + |y_1 - y_2| = 1$
- $0 \leq B \leq 1000$

- $1 \leq c_{i,j} \leq 1000$

## Output

輸出一行，包含一個整數，表示舊乙在不會被取消資格的前提下，至少要花幾秒才能拿到香蕉。

### Sample Input 1

```
4 4 5
12 11 2 1
13 10 3 4
14 9 8 5
15 16 7 6
5 5 4 4
3 5 1 5
2 5 2 4
4 4 3 2
```

### Sample Output 1

```
8
```

### Sample Input 2

```
5 6 11
1 2 3 4 5 6
22 21 20 19 18 7
23 14 15 16 17 8
24 13 12 11 10 9
25 26 27 28 29 30
2 7 3 6 2 3
7 5 2 7 2 2
5 4 1 5 6 10
4 7 4 6 5 4
1 7 5 10 3 5
```

### Sample Output 2

```
10
```

*This page is intentionally left blank.*



## C. 吃餅乾

Problem ID: biscuit

小堇跟小薺是感情很好的一對姐妹，為了慶祝她們的四歲生日，她們決定到小橘開設的「黑色薔薇」烘焙坊購買一些手工餅乾。她們一共購買了  $N$  塊餅乾，並將這些餅乾排成一列，將左邊數來的第  $i$  塊餅乾編號為  $i$ 。為了和平的決定要怎麼分配餅乾，小堇跟小薺決定進行以下的遊戲。

身為姐姐的小堇，她會先選擇一塊餅乾  $a_1$ ，接下來小薺會選擇一塊餅乾  $a_2$ ，並且把編號介於  $a_1$  和  $a_2$  之間的所有餅乾吃掉；接下來小堇會再選擇一塊還沒被吃掉的餅乾  $a_3$ ，並把編號介於  $a_2$  和  $a_3$  之間所有還沒被吃掉的餅乾吃掉；接下來小薺會再選擇一塊還沒被吃掉的餅乾  $a_4$ ，並把編號介於  $a_3$  和  $a_4$  之間所有還沒被吃掉的吃掉，依此類推，直到  $N$  塊餅乾都被吃掉為止。

請注意，選擇第一塊餅乾的小堇並不會吃掉那一塊餅乾。從第二塊餅乾開始，兩人才能將自己選擇的位置與前一個人選擇的位置之間的餅乾吃掉。我們稱編號  $i$  介於編號  $x$  和  $y$  之間，若  $i$  滿足  $\min(x, y) \leq i \leq \max(x, y)$ ，其中  $\min(x, y)$  表示  $x, y$  之間較小的數字， $\max(x, y)$  表示  $x, y$  之間較大的數字。請注意， $x$  和  $y$  兩個數字也被視為在區間之內。

為了確認這樣的遊戲是不是公平的，小堇跟小薺決定請你寫一支程式判斷，在姐妹兩人都使用最佳策略選擇每一塊餅乾的情況下，誰能吃到比較多的餅乾？

### Input

輸入僅一行，包含一正整數  $N$ 。

- $1 \leq N \leq 100$

### Output

輸出一行，包含一個字串。若小堇能吃到比較多的餅乾，請輸出 Sumire；若小薺能吃到比較多的餅乾，請輸出 Nazuna；若兩人吃到的餅乾一樣多，請輸出 Draw。

#### Sample Input 1

2

#### Sample Output 1

Nazuna

**Sample Input 2**

**Sample Output 2**

3	Nazuna
---	--------

## D. 賽馬播報

Problem ID: horserace

在 AT7 結束了程式競賽之旅後，沉迷於知名賽馬培育手機遊戲的他，決定投入賽馬運動，立志成為最強的練馬師，訓練出最強的馬匹！而身為專業競賽轉播員的愍乙己，播報賽馬當然也是難不倒他，他也將繼續支持 AT7，在 AT7 的練馬師之路上不斷給予最真誠的祝福。

在今天的賽馬比賽中，一共有  $N$  匹馬正在環狀的賽道上奔馳，所有的馬分別以 1 到  $N$  的正整數編號，且任兩匹馬的編號皆相異。在比賽開始時，所有馬匹會依照賽前決定的起跑順序  $a_1, a_2, \dots, a_N$  依序從跑道上的起點出發，以順時針的方向繞行環狀賽道。在編號  $a_N$  的馬出發之前，這段時間被視為「熱身時間」，此時不會有馬匹已經跑完一整圈回到起跑點，也不允許較晚出發的馬匹超越前方馬匹的狀況發生。

在賽馬比賽中，發生馬匹順序交換的「超車」狀況是最精彩的了！每次超車發生時，愍乙己便會大聲地播報：「編號  $x$  的馬超越了編號  $y$ ！」在賽道上，我們假設每匹馬的位置都是不一樣的，並且每匹馬都只會以順時針方向移動。因此當  $x$  超越  $y$  時， $y$  必須恰好是  $x$  順時針方向的下一匹馬，在超越過後， $x$  和  $y$  在賽道上的順序便會交換。

具體來說，從目前賽道上的任意一匹馬開始，以順時針方向依序記錄下  $N$  匹馬的編號為  $h_1, h_2, \dots, h_N$ 。對於所有  $1 \leq i < N$ ，編號為  $h_i$  的馬其順時針方向的下一匹馬為  $h_{i+1}$ ，在  $h_i$  超越  $h_{i+1}$  之後，所有馬匹的順序將變為  $h_1, h_2, \dots, h_{i+1}, h_i, \dots, h_N$ 。而對於編號  $h_N$  的馬，其順時針方向的下一匹馬為  $h_1$ ，在  $h_N$  超越  $h_1$  後，所有馬匹的順序將變為  $h_N, h_2, h_3, \dots, h_{N-2}, h_{N-1}, h_1$ 。

精采刺激的比賽正在進行，愍乙己依序記錄下了這次比賽發生的  $M$  次超越事件，卻忘記了比賽開始前每匹馬的起跑順序！他想請你寫一支程式回推出可能的起跑順序，或告訴他他的記錄有誤，沒有起跑順序可以依序讓這  $M$  次超越發生。

以範例測資 1 為例， $[1, 3, 2, 4], [2, 4, 1, 3], [3, 2, 4, 1]$  等都是可能的起跑順序；而  $[1, 2, 3, 4], [4, 3, 2, 1]$  等都不是可能的起跑順序。

### Input

輸入的第一行，包含兩個整數  $N, M$ ，表示馬的數量以及超越事件發生的次數。

接下來  $M$  行，每行包含兩個整數  $x, y$ ，表示依序發生的超越事件。

- $2 \leq N \leq 10^5$
- $0 \leq M \leq 10^5$
- $1 \leq x, y \leq N$
- $x \neq y$

## Output

若不存在可能的起跑順序，請輸出一行 No；否則，請先輸出一行 Yes，第二行請輸出  $N$  個整數  $a_i$ ，表示起跑的順序。

如果有多種可能的順序，你可以輸出任意一種。

請注意，同一行兩個整數間請以一個空白隔開，並且最後一個整數後不得有空白。

Sample Input 1	Sample Output 1
4 2 2 3 1 4	Yes 1 3 2 4
Sample Input 2	Sample Output 2
4 2 2 3 2 3	No
Sample Input 3	Sample Output 3
3 3 3 1 2 1 1 2	Yes 1 3 2

<b>Sample Input 4</b>	<b>Sample Output 4</b>
4 5 2 1 3 1 1 3 4 3 2 3	Yes 1 2 3 4

*This page is intentionally left blank.*

## E. 大富翁

Problem ID: monopoly

小波最喜歡玩單人遊戲，今天他打算玩一個名叫單人大富翁的遊戲。這個遊戲跟一般的大富翁不同，玩家一開始就會有  $10^{100}$  元，而且遊戲途中只會有領薪水、付過路費、暫停休息三種事件。

遊戲盤面總共有  $N$  個格子排成一個環，其中對於所有  $1 \leq i < N$ ，第  $i$  格的下一格是第  $i + 1$  格，特別的，第  $N$  格的下一格是第 1 格。玩家的棋子會從第 1 格開始。每個格子上面都有一個整數，代表停在這一格會發生的事件，假設第  $i$  格上面寫的整數是  $a_i$ ，當玩家停在第  $i$  格時：

- 如果  $a_i > 0$ ，則玩家會領  $a_i$  元的薪水。
- 如果  $a_i < 0$ ，則玩家必須付  $|a_i|$  元的過路費。
- 如果  $a_i = 0$ ，則什麼都不會發生。

注意到因為一開始玩家就站在第一格，因此玩家會在進行任何操作前就觸發該格的事件。

遊戲總共會有  $T$  輪操作，每一輪操作的過程如下：

1. 玩家需擲一顆  $K$  面骰，骰子的  $K$  個面分別寫著 1 到  $K$  的正整數各一次。
2. 假設擲出來的點數是  $X$ ，則玩家會將他的棋子從現在所在的格子移到往下走  $X$  格後的格子並停住。
3. 玩家會觸發棋子所暫停的格子上的事件。

因為小波討厭機率學，所以他會在**遊戲開始前**就會先把  $K$  面骰動手腳，讓他每輪骰到的正整數都一樣。

假設最後小波所剩的錢為  $L$ ，小波想知道  $L - 10^{100}$  最大可以是多少，換句話說，他想知道在他好好地選擇對骰子動手腳的方式下，遊戲結束時他能淨賺的錢的最大可能值。小波不只機率學不好，數學也不好，因此他拜託你幫他算出這個值。

## Input

輸入的第一行包含三個正整數  $N, K, T$ ，依序代表遊戲盤面格子總數、骰子的面數以及遊戲的輪數。

接著第二行包含  $N$  個正整數  $a_1, a_2, \dots, a_N$ ，意義如題敘所述。

- $2 \leq N \leq 2000$
- $1 \leq K \leq N$
- $1 \leq T \leq 10^{12}$
- $-10^5 \leq a_i \leq 10^5$

## Output

輸出一行，這行有一個整數，代表在小波好好地選擇對骰子動手腳的方式下，遊戲結束時他所能淨賺的錢的最大可能值。

Sample Input 1	Sample Output 1
9 3 1000 5 1 5 -1 -4 8 7 -6 3	3667
Sample Input 2	Sample Output 2
9 3 1000000000000 5 1 5 -1 -4 8 7 -6 3	3666666666667
Sample Input 3	Sample Output 3
10 10 10000 -1 -2 -3 -4 -5 -6 -7 -8 -9 -10	-10001



## F. 拼拼圖

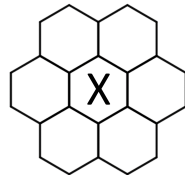
Problem ID: puzzle

(註：影印題本上的圖片是黑白的，但可於 Kattis 系統上看到彩色的圖片。)

給定一個大小為  $N$  的正六邊形棋盤，請用指定形狀的拼圖把它填滿。

### 棋盤

首先我們定義大小為 2 的棋盤如下：

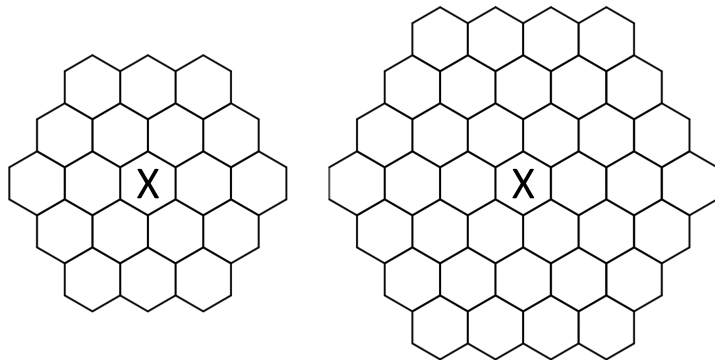


其中畫上 X 的格子是挖空的，上面不能放拼圖。

接著我們定義大小為  $N$  ( $N \geq 3$ ) 的棋盤，它可以用以下方法得出：

1. 放置一個大小為  $N - 1$  的棋盤
2. 將此棋盤的外圍多圍一圈正六邊形

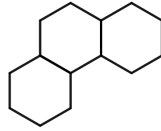
舉例來說，下圖顯示了大小分別為 3 與 4 的棋盤：



特別的是，大小為  $N$  的棋盤共有  $3N^2 - 3N$  個正六邊形（去掉正中間的格子）。

## 拼圖

拼圖的形狀如下，拼圖可以翻轉或旋轉，但不能改變拼圖的形狀。

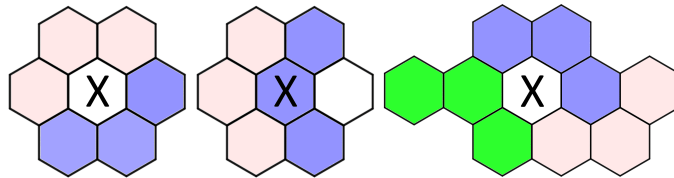


## 填滿

填滿一個棋盤需滿足以下條件：

- 棋盤的每一個正六邊形都被**恰**一塊拼圖覆蓋，特別注意正中間挖空的格子**不能**被任一塊拼圖覆蓋
- 沒有一塊拼圖超出棋盤範圍

以大小為 2 的棋盤為例，下圖說明了一些填滿與未填滿的例子：

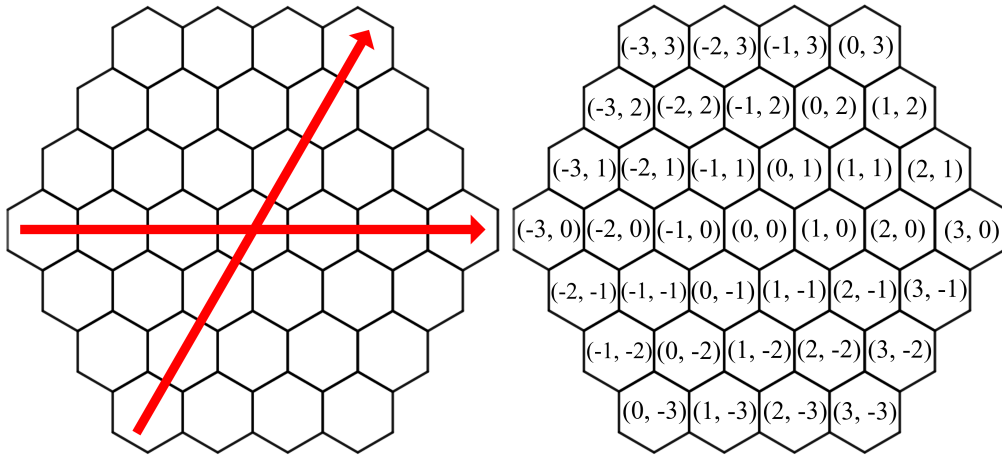


上圖中，最左邊是一個填滿的例子，其餘兩個都是未填滿的。中間的原因是正中間挖空的格子有被拼圖覆蓋，而右邊的原因是有兩塊拼圖超出了棋盤範圍。

由於一塊拼圖共包含 3 個正六邊形，因此可以得出一個結論：填滿一個大小為  $N$  的棋盤共需要  $N^2 - N$  塊拼圖。而且可以證明**任意大小的棋盤都能被這種形狀的拼圖填滿**。

## 座標

為了能夠表示出放置拼圖的位置，我們定義了正六邊形棋盤的座標系統。我們將棋盤的正中心設為原點  $(0, 0)$ ，將 3 點鐘方向定為  $x$  軸正向，1 點鐘方向定為  $y$  軸正向，如左下圖。而一個六邊形為一單位，這樣我們就可以推出每一個正六邊形格子的座標，如右下圖：



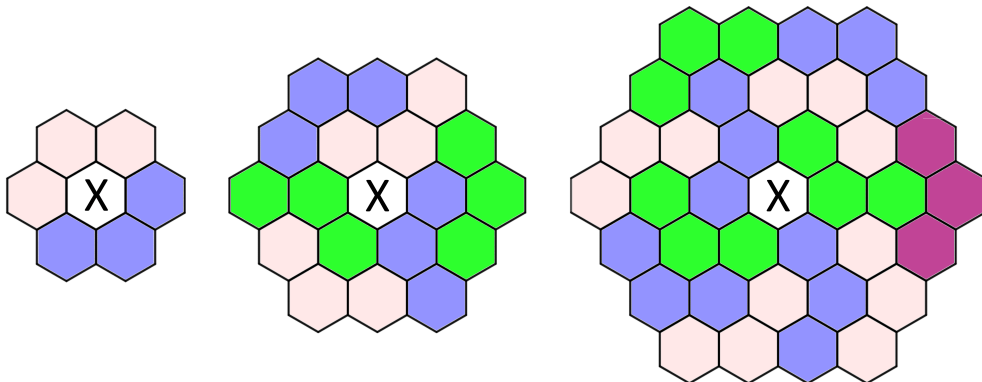
### Input

輸入第一行有一個正整數  $N$ ，代表六邊形棋盤的大小。

- $2 \leq N \leq 100$

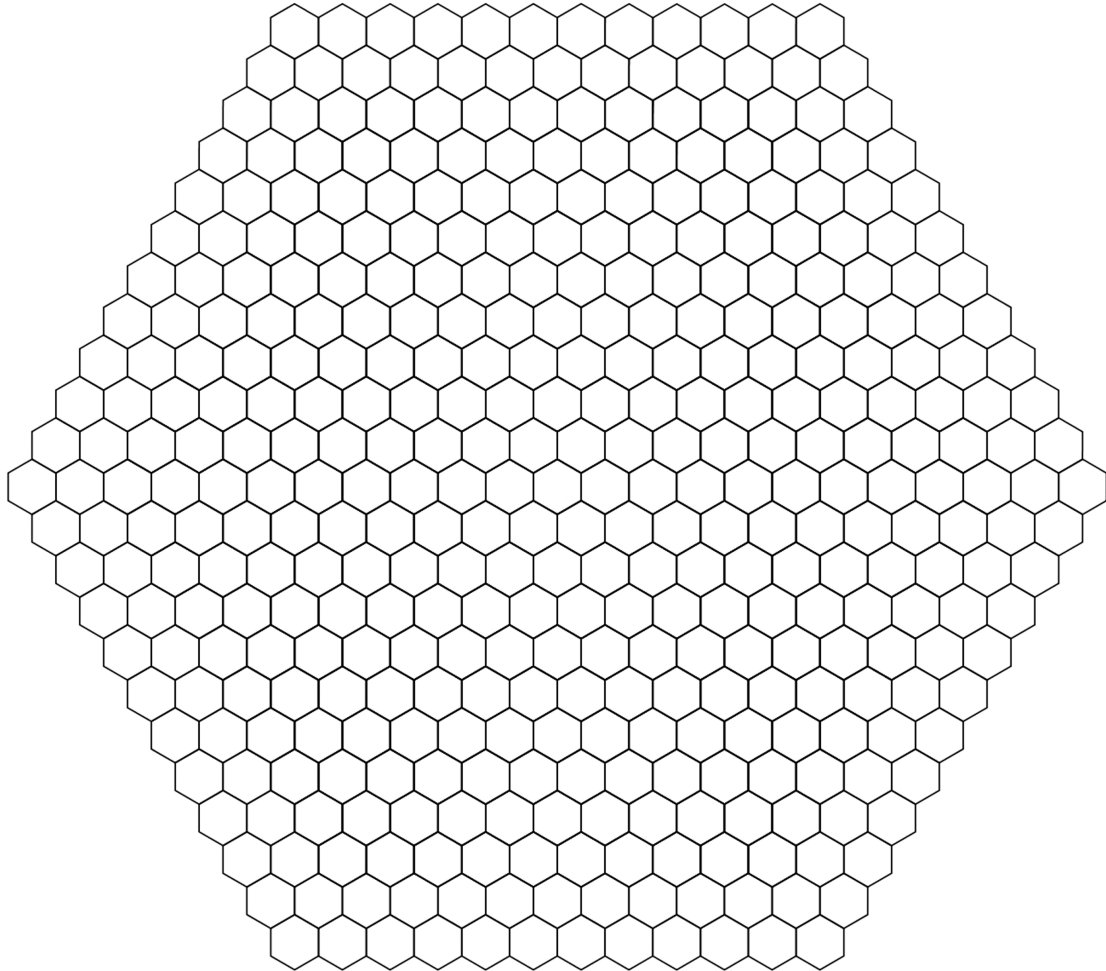
### Output

請輸出  $N^2 - N$  行，第  $i$  行有六個整數  $x_{i,1}, y_{i,1}, x_{i,2}, y_{i,2}, x_{i,3}, y_{i,3}$ ，代表第  $i$  塊拼圖會被放置在座標  $(x_{i,1}, y_{i,1}), (x_{i,2}, y_{i,2}), (x_{i,3}, y_{i,3})$  的位置。



## Hints

上圖顯示了棋盤大小分別為 2, 3, 4 的答案，同時你可以運用下方的巨大棋盤幫助你思考。



### Sample Input 1

3

### Sample Output 1

```
-1 2 -2 2 -2 1  
0 2 0 1 -1 1  
1 1 2 0 2 -1  
1 0 1 -1 2 -2  
-2 0 -1 0 0 -1  
-1 -1 0 -2 1 -2
```

## G. 看星星的鄭教授

Problem ID: stellar

「だって僕は星だから・Stellar Stellar !」、「星星點燈，照亮我的家門」、「馬鹿なわたしは歌うだけ・ぶちまけちゃおうか、星に」

從古至今有許多歌曲都在詠嘆天上繁星的浪漫。

不管是和朋友一起看星星，或是和喜歡的另一半一起看星星，知道哪個星星是哪個星座都會給你加不少的分。

為了簡化問題，我們假設星空是二維平面上一個  $N \times M$  的長方形棋盤方格， $N, M$  分別稱為星空的南北高度以及東西寬度。

- 在這個棋盤方格的  $N \times M$  格當中，每一格裡面最多只會有一顆星星。
- 我們將由這個棋盤方格由左至右、由上到下編號為  $(x, y)$ ，其中  $1 \leq x \leq M, 1 \leq y \leq N$ 。若在某一格  $(x, y)$  當中有一顆星星，則稱該星星的座標是  $(x, y)$ 。 $x$  和  $y$  分別被稱為該星星的  $x$  座標與  $y$  座標。 $x$  由左而右遞增， $y$  由上至下遞增。注意  $y$  座標遞增的方向。
- 一些星星會組成一個星座，並且所有星星都恰好屬於一個星座。星座以英文小寫和大寫字母作為編號。你不應該對此題中的星座形狀有任何多餘的假設。
- 在晚上的時候，我們能看到的所有星星是  $x$  座標介在  $L, R$ （包含）之間的星星。如果一個星座的所有星星都看得到的話，我們稱這個星座是能被完整看到的。
- 因為地球的公轉，每一天晚上看得到的星星集合都不一樣，因此能看到的星座也不一定。每過了一天，所有星星都會向棋盤方格的右邊移動一格。也就是說，每過一天，原本座標是  $(x, y)$  的星星會移動到  $(x + 1, y)$  去。特別地，原本  $x$  座標是  $M$  的那些星星，若他們原本座標為  $(M, y)$ ，過了一天之後則會移動到  $(1, y)$  去。

接下來鄭教授有  $Q$  個問題，想請問你在一年的第  $x_i$  天當中，能完整看到編號  $c_i$  的星座嗎？請注意，如果整個棋盤方格上沒有一個星星是  $c_i$  屬於這個星座，那麼請你當作看不到這個星座而輸出 No。

### Input

輸入的第一行有三個正整數  $N, M, Q$ ，分別表示星空的南北高度，星空的東西寬度，以及詢問的個數。輸入的第二行有兩個整數  $L, R$ ，代表在晚上的時候只能看到  $x$  座標處於  $L, R$  的星星。

接下來  $N$  行，每行都有一個長度為  $M$  的字串  $b_i$ ，其中  $b_i$  的第  $j$  個字元  $b_{i,j}$  代表在一年的第一天當中，座標  $(j, i)$  的星星是屬於哪一個星座。如果  $b_{i,j} = .$  代表在一年的第一天當中，沒有星星處在座標  $(j, i)$ 。

接下來  $Q$  行，每行都有一個正整數  $x_i$  以及一個字元  $c_i$ ，中間以空白隔開，代表鄭教授想請問你在一年的第  $x_i$  天當中，能完整看到編號  $c_i$  的星座嗎？

- $1 \leq N \leq 100$
- $1 \leq M \leq 10^5$
- $1 \leq Q \leq 100000$
- $1 \leq L \leq R \leq M$
- $b_{i,j}$  只會是「.」，小寫英文字母或是大寫英文字母
- $1 \leq x_i \leq M$
- $c_i$  只會是小寫英文字母或是大寫英文字母，即 a 到 z 和 A 到 Z

## Output

請輸出  $Q$  行，第  $i$  行輸出 Yes 或 No 代表第  $i$  個詢問是否可以完整看得到星座  $c_i$ 。

Sample Input 1	Sample Output 1
<pre> 10 25 10 5 20 .....0..0.... .CCCC.....SS. .....S... ....Z....Z.....S.. .....Z..Z.....S. .....X...X..SS.. .DD..F..... .D.D.FFFFFFF...X..... .D.D.F.....X..X..... .DD..F.....00...N.. 4 0 15 Z 25 0 4 X 4 N 20 Z 1 Z 5 Z 25 S 16 N                     </pre>	<pre> No No Yes No No No Yes Yes No Yes                     </pre>

Sample Input 2	Sample Output 2
7 26 20	No
2 23	No
b.....x...x....	Yes
..ddd..j..A.....o...u.	No
b...d.....A.A.z.z..o.o.b..	No
....d..j..AA.xz.z..ooo..u.	Yes
.d..d..j..A.A..z..o...o.u.	No
..dd...j.x.....b..u.	Yes
.x.....b..	No
16 a	Yes
26 M	No
21 o	Yes
6 n	Yes
2 g	Yes
9 u	Yes
7 s	No
17 d	No
14 A	No
9 j	Yes
26 u	No
21 x	
22 u	
3 z	
12 b	
20 d	
20 j	
13 A	
12 o	
3 b	