2023 網際網路程式設計全國大賽
高中組初賽

- 本次比賽共 7 題，含本封面共 16 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**）。

  輸出和題目指定的輸出格式必須完全一致，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 `TIMELIMIT` 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 ϵ 皆視為正確」，ϵ 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	蛋餅愛數樹	5 s	1024 MB
B	巫醫巫醫出題	1 s	1024 MB
C	砲打皮皮 5	1 s	1024 MB
D	全國系統程式設計大賽	3 s	1024 MB
E	巫醫巫醫拿錢錢	1 s	1024 MB
F	買漫畫	2 s	1024 MB
G	菜月昴與逆序數對	1 s	1024 MB

2023 網際網路程式設計全國大賽
輸入輸出範例

C 程式範例：

```c
#include <stdio.h>
int main()
{
    int cases;
    scanf("%d", &cases);
    for (int i = 0; i < cases; ++i)
    {
        long long a, b;
        scanf("%lld %lld", &a, &b);
        printf("%lld\n", a + b);
    }
    return 0;
}
```

C++ 程式範例：

```cpp
#include <iostream>
int main()
{
    int cases;
    std::cin >> cases;
    for (int i = 0; i < cases; ++i)
    {
        long long a, b;
        std::cin >> a >> b;
        std::cout << a + b << std::endl;
    }
    return 0;
}
```

A. 蛋餅愛數樹

Problem ID: tree-coloring

在一個充滿奇幻色彩的世界裡，有一種特別的彩色樹。這棵樹的每個節點都是有顏色的，但是顏色在這棵樹剛長成時不會顯現出來，而是要在時間的催化下才會各自慢慢顯現。在這個世界中，有一位名叫蛋餅的小精靈，他對這棵彩色樹充滿了好奇心。每天，他都會在樹林中漫步，觀察彩色樹的顏色變化，並試著計算出從某一個節點到樹上每個節點的路徑上，各自會經過多少種不同的顏色。現在，蛋餅需要你的幫助來完成他的數樹任務。

每次詢問時，蛋餅會指定一個節點 x，並且想知道從這個節點到樹上每一個節點所經過的不同顏色數量之總和。也就是說，假設從節點 a 走到節點 b 會經過的不同顏色數量為 $s(a,b)$，蛋餅想要詢問的就是 $\sum_{i=1}^{N} s(x,i)$。而在每次詢問之間可能會有節點顯現出他的原始顏色（注意，每個節點只有一種原始顏色，也只會被顯現一次）。

Input

輸入第一行有三個正整數 N, K, Q，代表有 N 個節點（每個節點一開始都沒有顏色）、接下來會被塗上的顏色有 K 種，以及總共有 Q 筆操作。

接下來的 $N-1$ 行每行包含兩個正整數 a_i, b_i，代表節點 a_i 和節點 b_i 中間有一條邊。

接下來的 Q 行，每行首先會有一個字串 op_i 代表會發生的事件：

- 若 op_i 為 `add`，接下來會有兩個正整數 x_i, c_i，代表節點 x_i 在此刻以後會顯現出他的原始顏色 c_i。
- 若 op_i 為 `query`，接下來會有一個正整數 x_i，代表蛋餅想算出此時節點 x_i 的答案。

- $1 \le N, K, Q \le 2 \cdot 10^5$
- $1 \le a_i, b_i \le N$
- $op_i \in \{\texttt{add}, \texttt{query}\}$
- $1 \le x_i \le N$
- $1 \le c_i \le K$

保證輸入會是一棵樹。

Output

對於每個 op_i 為 query 的詢問，請依序輸出該筆詢問要求的答案。

Sample Input 1

```
5 5 10
1 2
1 3
1 4
1 5
add 5 4
query 1
add 1 1
query 2
add 2 2
query 3
add 3 3
query 4
add 4 4
query 5
```

Sample Output 1

```
1
5
6
7
11
```

Sample Input 2

```
5 5 10
4 2
1 3
1 4
5 1
add 2 1
add 1 4
add 5 2
add 3 2
add 4 2
query 1
query 2
query 3
query 4
query 5
```

Sample Output 2

```
10
12
10
9
10
```

B. 巫醫巫醫出題

Problem ID: reorder

巫醫巫醫又抓到假解了！為了不讓錯誤的演算法通過巫醫巫醫熱騰騰剛新出的題目，巫醫巫醫決定加強測試資料，使得在賽中不會有假解通過。巫醫巫醫準備了 M 筆不同的測試資料，並且每筆測試資料的時間限制都一樣是 T 秒。

然而，如果生太多測試資料就會讓評測系統火力全開，承受不了選手提交的所有程式碼。在評測系統上，每個選手會依序測試每筆測試資料，這個順序是可以由裁判修改的。在依序測試的途中，如果有遇到一筆測試資料沒有通過，那麼接下來的所有測試資料都會被跳過不評測，以節省評測所需要的時間。

因為巫醫巫醫稽查選手的能力特別強，他已經調查並預測出了今年所有 N 支隊伍寫出來的程式碼在每一筆測試資料是否會答對。在巫醫巫醫的預測是正確的前提下，巫醫巫醫想要安排測試資料的順序，使得所需要消耗的評測時間盡量少，但巫醫巫醫在忙著出題，因此想請你幫忙回答這個問題。請注意你可以任意決定測試資料被測試的順序，但這個順序是每個參賽者共用的。

Input

輸入第一行有三個以空格分開的正整數 N, M, T，分別代表參賽隊伍個數，測試資料的數量以及一筆測試資料的時間限制。接下來 N 行的第 i 行會有一個長度 M 且只包含 O 和 X 兩種字元的字串，這個字串的第 j 個字元如果是 X，代表第 i 個參賽隊伍無法通過第 j 筆測試資料，否則該字元會是 O，代表第 i 個參賽隊伍可以通過第 j 筆測試資料。

- $1 \leq N \leq 188$
- $1 \leq M \leq 18$
- $1 \leq T \leq 30$

Output

輸出一行，表示在最佳的測試資料順序下，如果所有參賽者提交的程式碼在每筆沒有被跳過的測試資料都執行滿 T 秒的時間限制的話，總共會消耗評測系統幾秒的時間。

Sample Input 1	Sample Output 1
5 5 3 0000X 0X000 000X0 0X000 000XX	27

Sample Input 2	Sample Output 2
9 9 3 000X000XX 0X0X00000 0X000X000 0XX0X0000 000X0X000 00000X0X0 000000000 XX000000X X00X00000	66

Note

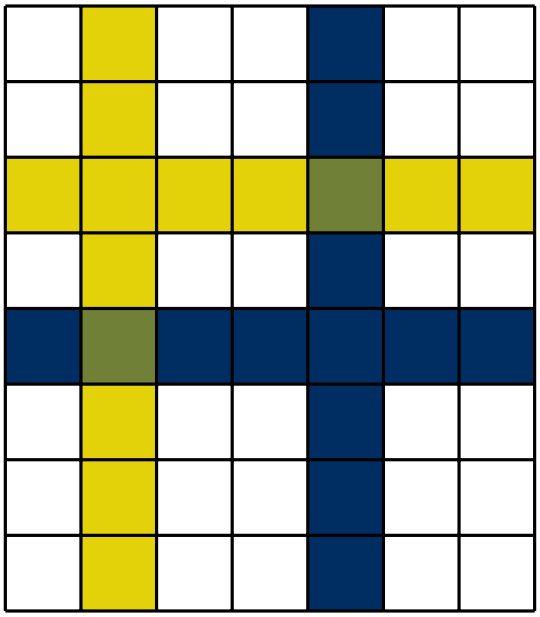
範例一當中，其中一種最好的測試資料順序是第五筆、第二筆、第四筆、第一筆、第三筆。五個參賽者依序分別會在執行 1 筆、2 筆、3 筆、2 筆、1 筆測試資料後被跳過評測，因此總共會花 $1 \times 3 + 2 \times 3 + 3 \times 3 + 2 \times 3 + 1 \times 3 = 27$ 秒。

C. 砲打皮皮 5

Problem ID: cross

在一場充滿挑戰的冒險中，你面臨著一個由皮皮們形成的巨大能量網格。這個網格內包含 $N \times M$ 個格子，每個格子裡都有一隻皮皮。一開始，所有皮皮都處於休眠狀態。每次，當你啟動一隻皮皮時，它會與其上下左右所有同行或是同列的皮皮進行能量交流，使得自己以及整個大十字上皮皮的狀態都發生變化（即從休眠狀態變為活躍狀態，或從活躍狀態變回休眠狀態）。

以下圖的 8×7 巨大能量網格為例：當我們啟動位於 $(3, 2)$ 的皮皮時，所有被黃色螢光十字標記的皮皮狀態都會發生變化（即全部變為活躍狀態），接下來當我們啟動位於 $(5, 5)$ 的皮皮時，所有被藍色螢光十字標記的皮皮狀態都會發生變化，其中，位於 $(3, 5)$ 和 $(5, 2)$ 的兩隻皮皮因為狀態被改變了兩次，會回到原本的休眠狀態。



你的目標是找出一種方法，通過一系列的啟動操作，使得整個能量網格中的所有皮皮都轉變為活躍狀態（注意，你可以啟動同一隻皮皮不只一次）。

注意，由於能量有限，你最多只能進行 32768 次的啟動操作。

皮皮，啟動！

Input

輸入包含以空格隔開的兩個正整數 N、M，分別代表巨大能量網格的長和寬。

- $1 \leq N, M \leq 100$

Output

若無法將整個能量網格中的所有皮皮都轉變為活躍狀態，請直接輸出「-1」（不含引號）。

若存在一系列的啟動操作能將整個能量網格中的所有皮皮都轉變為活躍狀態，請在第一行輸出你所需要的啟動操作數量 K。

接下來的 K 行每行請輸出兩個數字 i, j，代表你要啟動位於第 i 行第 j 列的皮皮。

如果有多種可以將整個能量網格中的所有皮皮都轉變為活躍狀態的方式，輸出任何一種皆會被視為正確。

- $1 \leq K \leq 32768$
- $1 \leq i \leq N$
- $1 \leq j \leq M$

Sample Input 1	Sample Output 1
2 2	4 2 1 2 2 1 1 1 2

Note

在這個範例中，只要將每個皮皮都啟動一次即可成功將所有皮皮都轉變為活躍狀態。

D. 全國系統程式設計大賽

Problem ID: nspc

眾所皆知全國系統程式設計大賽 (National Systems Programming Contest，簡稱 NSPC) 是題目品質優良的比賽，為了確保不同程度的選手都能有良好的比賽體驗，裁判長一定會好好把關題目的難度。為此，裁判長制定了一套題目難度標準，難度分為 N 個等級，每道題目都會是某一個難度等級。

今年的 NSPC 共有 M 名裁判，每一名裁判能夠出的題目都落在一個難度等級的區間裡。第 i 名裁判可以出的難度等級範圍是 $[s_i, t_i]$，特別的是，每個裁判肯定都是傾向出簡單題或傾向出難題，因此 $s_i = 1$ 或者 $t_i = N$ 至少有一個成立（也有可能 $s_i = 1$ 而且 $t_i = N$）。只要是落在這個區間裡的難度等級，這個裁判都一定出得了符合這個難度等級要求的題目，但難度等級不在這個區間裡的題目，這個裁判無論如何都不會出。

接下來到出題截止時間前有 Q 天，每天都會發生一個事件，事件是以下兩種的其中之一：

1. 有一個裁判改變了他的出題想法，因此他能出的難度等級範圍改變了。注意一名裁判改變出題想法後，他可能會從傾向出簡單題變成傾向出難題，也有可能從傾向出難題變成傾向出簡單題，當然也有可能不變。
2. 裁判長提出了一個出題計劃，每個出題計劃可以以兩個數字 (l, r) 表示，代表他希望這場比賽包含難度等級 $l, l+1, \ldots, r$ 的題目各恰一題。為了避免有人過勞，每一名裁判最多只能出一題，裁判長想知道這個出題計劃有沒有可能達成。

Input

第一行有兩個整數 N, M，代表裁判長制定了幾個難度等級以及裁判的人數。

接下來有 M 行，其中第 i 行包含兩個整數 s_i, t_i，代表第 i 名裁判可以出題的難度等級區間。

下一行有一個整數 Q，代表接下來到出題截止時間前還有 Q 天。

接下來有 Q 行，其中第 i 行代表第 i 天發生的事件為何，有以下兩種可能：

- $1\ x_i\ s'_i\ t'_i$：第 x_i 名裁判改變了他的出題想法，因此他能出題的難度等級範圍變更成了 $[s'_i, t'_i]$。

- $2\ l_i\ r_i$：裁判長提出了一個出題計劃 (l_i, r_i)，希望這場比賽包含難度等級 $l_i, l_i+1, \ldots, r_i$ 的題目各恰一題。

- $1 \le N, M, Q \le 5 \times 10^5$
- 對於每個 $1 \le i \le M$，$1 \le s_i \le t_i \le N$、$s_i = 1$ 或 $t_i = N$
- 若第 i 天的事件是第 1 種事件，則：
  - $1 \le x_i \le M$
  - $1 \le s'_i \le t'_i \le N$
  - $s'_i = 1$ 或 $t'_i = N$
- 若第 i 天的事件是第 2 種事件，則 $1 \le l_i \le r_i \le N$

Output

對於每個出題計劃，輸出 Yes 或 No 於一行，表示在裁判長提出這個出題計劃的當下，這個出題計劃是否有可能達成。

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

E. 巫醫巫醫拿錢錢

Problem ID: wiwi-rpg

巫醫巫醫最近迷上了一款名為方神的動作冒險 RPG，每日每夜都想盡辦法要在遊戲裡面賺錢。

終於，他來到了一個隱藏關卡，拿錢錢大挑戰！

在這個關卡中，有一個二維 $N \times M$ 的格子點地圖，地圖包含了格子點 $(0,0),(0,1),\dots(0,M-1),\dots(N-1,0),(N-1,1),\dots,(N-1,M-1)$。上面有兩個傳送門，分別在 $(S,0),(T,M-1)$。

關卡開始的時候，巫醫巫醫會被傳送到 $(S,0)$，而當巫醫巫醫走到 $(T,M-1)$ 的時候，他就能獲得曾經待過的所有格子點上面的金幣（包含 $(S,0),(T,M-1)$），並且離開關卡。

這個關卡有一些特殊限制：假設巫醫原本在座標 (r,c)，那巫醫巫醫下一步只能夠走到 $(r,c+1),(r+1,c),(r-1,c)$。也就是說，他只能往上下右移動，不能往左。不只這樣，巫醫巫醫只要走到超過地圖範圍的地方，或是已經走過的格子，就會立刻死掉。

巫醫巫醫想要知道，他有沒有辦法成功收集所有金幣並且離開遊戲，請你幫幫他。

Input

輸入第一行，有兩個正整數 N 和 M，意義如題目敘述。輸入第二行有兩個整數 S,T 意義如題目敘述。接下來有 N 行，其中第 i 行會有長度為 M 的 01 字串。令 $g_{i,j}$ 代表第 i 行的第 j 個字元，若 $g_{i,j}=1$ 則代表在座標 (i,j) 有一個金幣，$g_{i,j}=0$ 則代表沒有。

- $1 \le N \le 10^3$
- $2 \le M \le 10^3$
- $0 \le S,T < N$
- $g_{i,j} \in \{0,1\}$

Output

輸出一行，Yes 或是 No 代表巫醫巫醫能不能收集所有金幣並且離開關卡。

Sample Input 1	Sample Output 1
3 4 1 1 0011 0110 0110	Yes

Sample Input 2	Sample Output 2
3 4 1 1 1010 0110 0111	No

F. 買漫畫

Problem ID: comics

圓圓是個喜歡看漫畫的宅宅，她最喜歡的作者小坂老師的兩部作品《我推的方塊》跟《像素貓想讓人膜拜～天才們的競程裝弱戰～》都推出了漫畫！這讓圓圓迫不及待的想要去漫畫店購買。

《我推的方塊》一系列共有 N 集，而《像素貓想讓人膜拜～天才們的競程裝弱戰～》一系列共有 M 集。圓圓家附近一共有 K 間漫畫店，這些漫畫店為了避免惡性競爭，他們達成一個共識：**每一集漫畫只會在其中一間漫畫店中販賣**。而經過了事先調查，圓圓已經得知《我推的方塊》第 i 集會在第 a_i 間漫畫店販賣，而《像素貓想讓人膜拜～天才們的競程裝弱戰～》第 i 集會在第 b_i 間漫畫店販賣。

當圓圓去第 i 間漫畫店時，對於每一集在該間漫畫店有販賣的漫畫，她都可以選擇買或不買。因為圓圓還是個窮學生，每一集漫畫她最多只會買一本。又因為圓圓喜歡按照集數順序看漫畫，她所購買的漫畫必須要滿足以下條件：

- 若她買了《我推的方塊》第 i 集且 $i > 1$，則她必須要買《我推的方塊》第 $i - 1$ 集。
- 若她買了《像素貓想讓人膜拜～天才們的競程裝弱戰～》第 j 集且 $j > 1$，則她必須要買《像素貓想讓人膜拜～天才們的競程裝弱戰～》第 $j - 1$ 集。

然而圓圓是個宅宅，她最多只能去其中的 X 間漫畫店，否則她會因體力不足而暈倒。現在圓圓想要知道她最多可以買幾本漫畫，你能幫幫她嗎？

Input

輸入有三行，第一行有四個以空格分開的正整數 N, M, K, X。

第二行有 N 個以空格分開的正整數 $a_1, a_2, \ldots, a_N$。

第三行有 M 個以空格分開的正整數 $b_1, b_2, \ldots, b_M$。

- $1 \leq N, M \leq 5 \times 10^5$
- $1 \leq K \leq N + M$
- $1 \leq X \leq K$
- $1 \leq a_i, b_i \leq K$

Output

輸出一行，表示圓圓最多可以買幾本漫畫。

Sample Input 1

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

Sample Output 1

```
7
```

Sample Input 2

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

Sample Output 2

```
18
```

Sample Input 3

```
20 20 18 8
8 9 1 17 18 2 11 17 15 12 9 16 8 13 17 15 16 9 8 17
15 16 2 16 16 8 2 5 5 5 10 12 11 10 12 3 2 13 8 4
```

Sample Output 3

```
16
```

Note

在第一筆範例測試資料中，圓圓可以選擇去第 $1, 2, 4$ 間漫畫店，這樣一來她可以買《我推的方塊》前 2 集與《像素貓想讓人膜拜～天才們的競程裝弱戰～》前 5 集，總共 7 本。

在第二筆範例測試資料中，她可以去每一間漫畫店所以可以全部購買。

G. 菜月昴與逆序數對

Problem ID: subaru

菜月昴又双叒叕遇到新的敵人了。沒錯，就是一道逆序數對問題！

我們回顧一下逆序數對：對於一個陣列 $a = [a_1, a_2, \ldots, a_k]$，$\text{inv}(a)$ 為滿足 $1 \le i < j \le k, a_i > a_j$ 的 (i, j) 個數，也就是逆序數對個數。特別的，一個空陣列的逆序數對個數定義為 0。

如果菜月昴遇到的問題只是單純給一個陣列求逆序數對的話，那完全不足為懼，但想當然這道逆序數對問題絕對沒那麼簡單。在經過數次交鋒後，菜月昴終於知道要解決這個問題所需要完成的事項。以下是他所解讀出來的問題敘述：

對於一個正整數陣列 p，我們稱他是一個「好陣列」若他的元素總和為 N。現在你的手上有一個長度為 N 的陣列 a，對於一個好陣列 $p = [p_1, p_2, \ldots, p_k]$，我們用以下步驟定義兩個陣列 b_p, c_p：

一開始 b_p, c_p 為空陣列。我們將 a 由左到右切成 k 段，第 i 段的大小為 p_i。接下來按照 $i = 1, 2, \ldots, k$ 的順序，將第 i 段中的第 $\lceil \frac{p_i}{2} \rceil$ 項拿走並放到 b_p 的最後面，剩下的 $p_i - 1$ 個數字按照原本的順序放到 c_p 的最後面，最終所得到的 b_p, c_p 長度分別為 $k, N - k$。注意到 a 陣列在做完以上所有步驟後會恢復原樣，也可以說 a 陣列本身並不會有任何變化。

對於一個好陣列 p，我們稱 a 的 p-合適程度為 $\text{inv}(b_p) - \text{inv}(c_p)$。

令 sum 為所有相異好陣列 p 對應的 a 的 p-合適程度之總和，請求出 sum 除以 998244353 的餘數。

菜月昴知道自己絕對沒那個能力解決這道問題，他知道這時候就要依靠身邊強大的人來幫助他。身為強大的人的你，請幫菜月昴解決這道問題。

註：$\lceil x \rceil$ 代表不小於 x 的最小整數。一個負整數 X 除以 998244353 的餘數為最小的非負整數 Y 使得 $Y - X$ 是 998244353 的倍數。

Input

輸入的第一行只有一個整數 N，代表陣列 a 的長度。

輸入的第二行有 N 個正整數 $a_1, a_2, \ldots, a_N$，代表陣列 a。

- $1 \leq N \leq 2 \times 10^5$
- 對於所有 $1 \leq i \leq N$，$1 \leq a_i \leq 10^6$

Output

輸出問題的答案，也就是 sum 除以 998244353 的餘數。

Sample Input 1	Sample Output 1
5 4 8 7 6 3	16

Sample Input 2	Sample Output 2
4 86419 7532 7122 45510	10

Note

在範例一中，$[1,1,1,1,1],[1,1,2,1],[2,1,2],[3,2],[5]$ 等陣列都是好陣列，而他們依序作為 p 的 b_p, c_p 以及 a 的 p-合適程度如下：

- $p=[1,1,1,1,1]$：$b_p=[4,8,7,6,3], c_p=[]$，a 的合適程度為 $7-0=7$。
- $p=[1,1,2,1]$：$b_p=[4,8,7,3], c_p=[6]$，a 的合適程度為 $4-0=4$。
- $p=[2,1,2]$：$b_p=[4,7,6], c_p=[8,3]$，a 的合適程度為 $1-1=0$。
- $p=[3,2]$：$b_p=[8,6], c_p=[4,7,3]$，a 的合適程度為 $1-2=-1$。
- $p=[5]$：$b_p=[7], c_p=[4,8,6,3]$，a 的合適程度為 $0-4=-4$。

注意到以上並未列舉出所有的好陣列。