A - Inside or Outside

Time Limit: 2 sec / Memory Limit: 1024 MiB

Score : 700 points

Problem Statement

There is an integer sequence $x = (x_1, \ldots, x_N)$, which is initialized with $x_1 = \cdots = x_N = 0$.

You will perform M operations on this integer sequence. In the i-th operation, you are given an integer pair (L_i, R_i) such that $1 \leq L_i \leq R_i \leq N$, and you must perform **exactly one** of the following three operations:

- Operation 0: Do nothing. This operation incurs a cost of 0.
- Operation 1: For each integer j with $1 \leq j \leq N$, if $L_i \leq j \leq R_i$ **holds**, set $x_j = 1$. This operation incurs a cost of 1.
- Operation 2: For each integer j with $1 \leq j \leq N$, if $L_i \leq j \leq R_i$ **does not hold**, set $x_j = 1$. This operation incurs a cost of 1.

Your goal is to make $x_1 = \cdots = x_N = 1$ hold at the end. Determine whether this goal can be achieved. If it can be achieved, present one way to achieve it where the total cost of the operations is minimized.

Constraints

- $1 \leq N \leq 1000000$
- $1 \leq M \leq 200000$
- $1 \leq L_i \leq R_i \leq N$
- All input values are integers.

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \quad M \\
L_1 \quad R_1 \\
\vdots \\
L_M \quad R_M
\end{array}
$$

Output

If the goal is not achievable, print `-1`.

If the goal is achievable, print one way to achieve it where the total cost of the operations is minimized, in the following format, where K is the minimum total cost of the operations, and op_i is the type of operation ($0, 1$, or 2) chosen for the i-th operation.

$$K$$
$$\mathrm{op}_1 \quad \cdots \quad \mathrm{op}_M$$

If there are multiple ways that minimize the total cost, printing any one of them is accepted.

Sample Input 1

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

Sample Output 1

```
2
2 0 1 0
```

In the sample output, x changes as follows:

- Initially, $x = (0, 0, 0, 0, 0)$.
- In the 1st operation, Operation 2 is performed. x_1 and x_5 become 1, so $x = (1, 0, 0, 0, 1)$.
- In the 2nd operation, Operation 0 is performed. x remains $(1, 0, 0, 0, 1)$.
- In the 3rd operation, Operation 1 is performed. x_1, x_2, x_3, x_4 become 1, so $x = (1, 1, 1, 1, 1)$.
- In the 4th operation, Operation 0 is performed. x remains $(1, 1, 1, 1, 1)$.

Sample Input 2

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

Sample Output 2

```
1
0 1 0 0
```

Sample Input 3

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

Sample Output 3

```
2
1 1
```

Sample Input 4

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

Sample Output 4

```
-1
```

B - L Partition

Time Limit: 2 sec / Memory Limit: 1024 MiB

Score : 800 points

Problem Statement

There is an $N \times N$ grid. Let (i, j) denote the cell at the i-th row from the top and the j-th column from the left.

For $K = 1, 2, \ldots, N$, a **level K L-shape** is a set of $2K - 1$ cells that satisfies at least one of the following four conditions:

- All cells reachable from a certain cell (i, j) by moving down or right between 0 and $K - 1$ cells, inclusive (where $1 \leq i \leq N - K + 1, 1 \leq j \leq N - K + 1$).
- All cells reachable from a certain cell (i, j) by moving down or left between 0 and $K - 1$ cells, inclusive (where $1 \leq i \leq N - K + 1, K \leq j \leq N$).
- All cells reachable from a certain cell (i, j) by moving up or right between 0 and $K - 1$ cells, inclusive (where $K \leq i \leq N, 1 \leq j \leq N - K + 1$).
- All cells reachable from a certain cell (i, j) by moving up or left between 0 and $K - 1$ cells, inclusive (where $K \leq i \leq N, K \leq j \leq N$).

You are given a cell (a, b) and Q queries $k_1, \ldots, k_Q$.

For each i, print the number, modulo 998244353, of ways to partition the entire grid into exactly one level 1 L-shape, one level 2 L-shape, $\ldots$, and one level N L-shape so that cell (a, b) is contained in the level k_i L-shape.

Constraints

- $1 \leq N \leq 10^7$
- $1 \leq a \leq N$
- $1 \leq b \leq N$
- $1 \leq Q \leq \min\{N, 200000\}$
- $1 \leq k_1 < \cdots < k_Q \leq N$
- All input values are integers.

Input

The input is given from Standard Input in the following format:

$$N \quad a \quad b$$
$$Q$$
$$k_1 \quad \cdots \quad k_Q$$

Output

Print Q lines.

The i-th line should contain the number, modulo 998244353, of ways to partition the grid into exactly one level 1 L-shape, one level 2 L-shape, $\ldots$, and one level N L-shape so that cell (a, b) is contained in the level k_i L-shape.

Sample Input 1

```
3 1 2
1
2
```

Sample Output 1

```
6
```

The six ways shown in the following figure are the solutions. In the figure, an integer k in a cell means that the cell belongs to the level k L-shape.

Sample Input 2

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

Sample Output 2

```
4
32
128
```

Sample Input 3

```
100 50 50
4
1 10 50 100
```

Sample Output 3

```
934228871
758172260
444239843
0
```

C - Basic Grid Problem with Updates

Time Limit: 5 sec / Memory Limit: 1024 MiB

Score : 800 points

Problem Statement

There is an $H \times W$ grid. Let (h, w) denote the cell at the h-th row from the top and the w-th column from the left. A non-negative integer $A_{h,w}$ is written in cell (h, w).

Takahashi starts at cell (sh, sw) and will perform Q changes to the grid. The i-th change is given by a character d_i (d_i is one of `L`, `R`, `U`, `D`) and a non-negative integer a_i, meaning Takahashi will do the following:

- Move one cell in the direction d_i. That is, if d_i is `L`, move left; if `R`, move right; if `U`, move up; if `D`, move down by one cell. Then, let the destination cell be (h, w), and set $A_{h,w}$ to a_i.

It is guaranteed that in each change, he can move one cell in direction d_i.

After each change, print the answer to the following problem:

> A sequence of cells $P = ((h_1, w_1), \ldots, (h_M, w_M))$ is said to be a **path** if and only if it satisfies all of the following conditions:
>
> - $(h_1, w_1) = (1, 1)$, $(h_M, w_M) = (H, W)$, and $M = H + W - 1$.
> - For every i with $1 \leq i \leq M - 1$, either $(h_{i+1}, w_{i+1}) = (h_i + 1, w_i)$ or $(h_{i+1}, w_{i+1}) = (h_i, w_i + 1)$.
>
> There are $\binom{H+W-2}{H-1}$ paths. For a path $P = ((h_1, w_1), \ldots, (h_M, w_M))$, define $f(P) = \prod_{1 \leq i \leq M} A_{h_i, w_i}$. Print the sum, modulo 998244353, of $f(P)$ over all paths P.

Constraints

- $2 \leq H, W \leq 200000$
- $HW \leq 200000$
- $0 \leq A_{h,w} < 998244353$
- $1 \leq Q \leq 200000$
- $1 \leq sh \leq H, 1 \leq sw \leq W$
- $0 \leq a_i < 998244353$
- $H, W, A_{h,w}, Q, sh, sw$, and a_i are integers.
- Each d_i is `L`, `R`, `U`, or `D`.
- In each change, Takahashi can move one cell in the direction d_i.

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{l}
H \quad W \\
A_{1,1} \quad \cdots \quad A_{1,W} \\
\vdots \\
A_{H,1} \quad \cdots \quad A_{H,W} \\
Q \quad sh \quad sw \\
d_1 \quad a_1 \\
\vdots \\
d_Q \quad a_Q
\end{array}
$$

Output

Print Q lines.

The i-th line should contain the sum, modulo 998244353, of $f(P)$ over all paths P after performing the i-th change to the grid.

Sample Input 1

```
2 3
1 2 3
4 5 6
3 2 2
U 7
R 8
L 9
```

Sample Output 1

```
456
666
822
```

- Initially, Takahashi is at $(2, 2)$.
- Move up, then set $A_{1,2}$ to 7. The value of $f(P)$ for each path is:
  - $P = ((1, 1), (1, 2), (1, 3), (2, 3))$: $f(P) = 1 \times 7 \times 3 \times 6 = 126$.
  - $P = ((1, 1), (1, 2), (2, 2), (2, 3))$: $f(P) = 1 \times 7 \times 5 \times 6 = 210$.
  - $P = ((1, 1), (2, 1), (2, 2), (2, 3))$: $f(P) = 1 \times 4 \times 5 \times 6 = 120$.
- Move right, then set $A_{1,3}$ to 8. The value of $f(P)$ for each path is:
  - $P = ((1, 1), (1, 2), (1, 3), (2, 3))$: $f(P) = 1 \times 7 \times 8 \times 6 = 336$.
  - $P = ((1, 1), (1, 2), (2, 2), (2, 3))$: $f(P) = 1 \times 7 \times 5 \times 6 = 210$.
  - $P = ((1, 1), (2, 1), (2, 2), (2, 3))$: $f(P) = 1 \times 4 \times 5 \times 6 = 120$.
- Move left, then set $A_{1,2}$ to 9. The value of $f(P)$ for each path is:
  - $P = ((1, 1), (1, 2), (1, 3), (2, 3))$: $f(P) = 1 \times 9 \times 8 \times 6 = 432$.
  - $P = ((1, 1), (1, 2), (2, 2), (2, 3))$: $f(P) = 1 \times 9 \times 5 \times 6 = 270$.
  - $P = ((1, 1), (2, 1), (2, 2), (2, 3))$: $f(P) = 1 \times 4 \times 5 \times 6 = 120$.

Sample Input 2

```
5 4
147015809 294958521 852121867 499798308
790350368 404692331 645419803 290531806
275766153 896286651 239187926 945049742
340760022 236352314 926236110 223464913
287023679 590772036 340282357 521075891
6 3 1
U 344644511
R 45812235
D 260083498
R 781118585
L 156297846
L 411901560
```

Sample Output 2

```
299123226
548055393
810247224
876210800
773990840
506814544
```

D - Matrix Pow Sum

Time Limit: 2 sec / Memory Limit: 1024 MiB

Score : 900 points

Problem Statement

You are given a prime number p and an $N \times N$ matrix $A = (A_{i,j})$ ($1 \leq i, j \leq N$). Each element of A is an integer between 0 and $p - 1$, inclusive.

Consider a matrix B obtained by replacing each zero in A with an integer between 1 and $p - 1$, inclusive. There are $(p - 1)^K$ such matrices B, where K is the number of zeros in A.

Find each element, modulo p, of the sum of B^p over all possible B.

Constraints

- $1 \leq N \leq 100$
- p is a prime such that $1 \leq p \leq 10^9$.
- $0 \leq A_{i,j} \leq p - 1$
- All input values are integers.

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \quad p \\
A_{1,1} \quad \cdots \quad A_{1,N} \\
\vdots \\
A_{N,1} \quad \cdots \quad A_{N,N}
\end{array}
$$

Output

Print N lines.

The i-th line should contain, in the order $j = 1, \ldots, N$, the (i, j) element of the sum, modulo p, of B^p over all possible B, separated by spaces.

Sample Input 1

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

Sample Output 1

```
0 2
1 2
```

B^p for all possible B are as follows:

- $\begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}^3 = \begin{pmatrix} 5 & 8 \\ 8 & 13 \end{pmatrix}$
- $\begin{pmatrix} 1 & 1 \\ 2 & 2 \end{pmatrix}^3 = \begin{pmatrix} 9 & 9 \\ 18 & 18 \end{pmatrix}$
- $\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}^3 = \begin{pmatrix} 14 & 13 \\ 13 & 14 \end{pmatrix}$
- $\begin{pmatrix} 2 & 1 \\ 2 & 2 \end{pmatrix}^3 = \begin{pmatrix} 20 & 14 \\ 28 & 20 \end{pmatrix}$

Print each element, modulo $p = 3$, of their sum $\begin{pmatrix} 48 & 44 \\ 67 & 65 \end{pmatrix}$.

Sample Input 2

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

Sample Output 2

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

B^p for all possible B are as follows:

- $\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}^2 = \begin{pmatrix} 3 & 3 & 3 \\ 3 & 3 & 3 \\ 3 & 3 & 3 \end{pmatrix}$

Print each element, modulo $p = 2$, of their sum $\begin{pmatrix} 3 & 3 & 3 \\ 3 & 3 & 3 \\ 3 & 3 & 3 \end{pmatrix}$.

Sample Input 3

```
4 13
0 1 2 0
3 4 0 5
0 6 0 7
8 9 0 0
```

Sample Output 3

```
8 0 6 5
11 1 8 5
8 0 4 12
8 0 1 9
```

E - Gaps of 1 or 2

Time Limit: 5 sec / Memory Limit: 1024 MiB

Score : 1100 points

Problem Statement

You are given a length-N sequence $A = (A_1, \ldots, A_N)$ of non-negative integers. Answer Q queries.

In the i-th query, you are given integers L_i and R_i such that $1 \leq L_i \leq R_i \leq N$. Solve the following problem for the subsequence $B = (A_{L_i}, \ldots, A_{R_i})$ of length $R_i - L_i + 1$.

> We repeat the following operation on B:
>
> - Choose integers i and j with $1 \leq i, j \leq |B|$ such that $B_i \geq 1, B_j \geq 1$, and $1 \leq j - i \leq 2$. Subtract 1 from B_i and B_j.
>
> Find the maximum number of times the operation can be performed.

Constraints

- $1 \leq N \leq 200000$
- $1 \leq Q \leq 200000$
- $0 \leq A_i \leq 10^9$
- $1 \leq L_i \leq R_i \leq N$
- All input values are integers.

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \quad Q \\
A_1 \quad \cdots \quad A_N \\
L_1 \quad R_1 \\
\vdots \\
L_Q \quad R_Q
\end{array}
$$

Output

Print Q lines.

The i-th line should contain the maximum number of times the operation can be performed for $B = (A_{L_i}, \ldots, A_{R_i})$.

Sample Input 1

```
6 1
1 1 4 0 3 2
1 6
```

Sample Output 1

```
5
```

In this example, we solve the problem for $B = (1, 1, 4, 0, 3, 2)$. We can perform five operations as follows:

- Choose $i = 1$ and $j = 3$. Then $B = (0, 1, 3, 0, 3, 2)$.
- Choose $i = 2$ and $j = 3$. Then $B = (0, 0, 2, 0, 3, 2)$.
- Choose $i = 3$ and $j = 5$. Then $B = (0, 0, 1, 0, 2, 2)$.
- Choose $i = 5$ and $j = 6$. Then $B = (0, 0, 1, 0, 1, 1)$.
- Choose $i = 5$ and $j = 6$. Then $B = (0, 0, 1, 0, 0, 0)$.

Sample Input 2

```
6 6
49 83 10 77 21 62
1 1
1 2
1 3
3 5
1 6
5 6
```

Sample Output 2

```
0
49
59
31
151
21
```