A - Is it rated?

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

AtCoder Regular Contest (ARC) currently has two divisions: Div. 1 and Div. 2. In ARC Div. 1, participants whose **rating** is between 1600 and 2999, inclusive, are **rated**. In ARC Div. 2, participants whose rating is between 1200 and 2399, inclusive, are rated.

You are given positive integers R and X.

Determine whether a person with rating R is rated in ARC Div. X.

Constraints

- $1 \leq R \leq 4229$
- $1 \leq X \leq 2$
- All input values are integers.

Input

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

$$R \quad X$$

Output

If a person with rating R is rated in ARC Div. X, output `Yes`; otherwise, output `No`.

Sample Input 1

```
2000 1
```

Sample Output 1

```
Yes
```

Because 2000 lies between 1600 and 2999, a person with rating 2000 is rated in ARC Div. 1.

Sample Input 2

```
1000 1
```

Sample Output 2

```
No
```

Because 1000 is less than 1600, a person with rating 1000 is not rated in ARC Div. 1.

Sample Input 3

```
1500 2
```

Sample Output 3

```
Yes
```

Because 1500 lies between 1200 and 2399, a person with rating 1500 is rated in ARC Div. 2.

Sample Input 4

```
2800 2
```

Sample Output 4

```
No
```

Because 2800 exceeds 2399, a person with rating 2800 is not rated in ARC Div. 2.

B - Not All

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 200 points

Problem Statement

You are given an integer sequence $A = (A_1, A_2, \ldots, A_N)$ of length N and a positive integer M.

Your goal is to make the following condition **false** by performing this operation between 0 and N times (inclusive): remove the last element of A.

- **Condition**: A contains every integer from 1 through M.

Find the minimum number of operations required.

Under the constraints of this problem, it can be proved that it is always possible to make the condition false by performing the operation between 0 and N times.

Constraints

- $1 \leq M \leq N \leq 100$
- $1 \leq A_i \leq M$
- All input values are integers.

Input

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

$$\begin{array}{l} N \quad M \\ A_1 \quad A_2 \quad \ldots \quad A_N \end{array}$$

Output

Output the minimum number of operations required to make the condition false.

Sample Input 1

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

Sample Output 1

```
2
```

Initially, $A = (3, 2, 3, 1, 2)$. Since A contains every integer from 1 through 3, the condition holds.

If you perform the operation once, $A = (3, 2, 3, 1)$. The condition still holds.

If you perform the operation once more, $A = (3, 2, 3)$. The integer 1 is missing, so the condition no longer holds.

Therefore, the minimum required number of operations is 2.

Sample Input 2

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

Sample Output 2

```
0
```

Since A initially lacks the integer 2, the condition is already false, so no operation is needed.

Sample Input 3

```
10 4
1 3 3 4 2 1 3 1 2 4
```

Sample Output 3

```
6
```

C - Sum of Product

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 300 points

Problem Statement

You are given a length-N integer sequence $A = (A_1, A_2, \ldots, A_N)$.

Compute the value of $\displaystyle\sum_{1 \le i < j \le N} A_i A_j$.

Constraints

- $2 \le N \le 3 \times 10^5$
- $1 \le A_i \le 10^4$
- All input values are integers.

Input

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

$$\begin{array}{l} N \\ A_1 \quad A_2 \quad \ldots \quad A_N \end{array}$$

Output

Output the answer.

Sample Input 1

```
3
4 2 3
```

Sample Output 1

```
26
```

We have $\displaystyle\sum_{1 \le i < j \le N} A_i A_j = A_1 A_2 + A_1 A_3 + A_2 A_3 = 4 \cdot 2 + 4 \cdot 3 + 2 \cdot 3 = 26$.

Sample Input 2

```
2
9 45
```

Sample Output 2

```
405
```

Sample Input 3

```
10
7781 8803 8630 9065 8831 9182 8593 7660 7548 8617
```

Sample Output 3

```
3227530139
```

D - Escape Route

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 400 points

Problem Statement

> One day, Takahashi visited a cinema and decided to draw arrows on every floor tile, each pointing toward the nearest emergency exit.

You are given a grid with H rows and W columns. Let (i, j) denote the cell at the i-th row from the top and the j-th column from the left.
Each cell is represented by one of the following characters $S_{i,j}$:

- `.` : corridor cell
- `#` : wall cell
- `E` : emergency exit

For any corridor cell (i, j), define $d(i, j)$, the distance to the nearest emergency exit, as follows:

- Starting from (i, j), repeatedly move to an adjacent **non-wall cell** in one of the four cardinal directions (up, down, left, right) until you reach an emergency exit. $d(i, j)$ is the minimum number of moves required.

It is guaranteed that $d(i, j)$ is definable for every corridor cell in the given grid; that is, every corridor cell (i, j) has at least one emergency exit reachable by passing through only corridor cells.

Write exactly one arrow (up, down, left, or right) in every corridor cell so that the following condition holds:

- For every corridor cell (i, j), if you start at (i, j) and perform the following action $d(i, j)$ times, you reach an emergency exit:
  - Move one cell in the direction of the arrow written in your current cell. (You cannot move into a wall cell or outside the grid.)

Constraints

- $2 \leq H \leq 1000$
- $2 \leq W \leq 1000$
- Each $S_{i,j}$ is `.`, `#`, or `E`.
- Every corridor cell (i, j) has at least one reachable emergency exit.
- H and W are integers.

Input

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

$$
\begin{array}{l}
H \quad W \\
S_{1,1}S_{1,2}\ldots S_{1,W} \\
S_{2,1}S_{2,2}\ldots S_{2,W} \\
\vdots \\
S_{H,1}S_{H,2}\ldots S_{H,W}
\end{array}
$$

Output

Let $T_{i,j}$ be the state of cell (i, j) after writing the arrows. $T_{i,j}$ is one of the following:

- `^` : corridor cell with an upward arrow
- `v` : corridor cell with a downward arrow
- `<` : corridor cell with a leftward arrow
- `>` : corridor cell with a rightward arrow
- `#` : wall cell
- `E` : emergency exit

Output the grid in the following format:

$$\begin{array}{l} T_{1,1}T_{1,2}\dots T_{1,W} \\ T_{2,1}T_{2,2}\dots T_{2,W} \\ \vdots \\ T_{H,1}T_{H,2}\dots T_{H,W} \end{array}$$

If multiple solutions exist, any of them will be accepted.

Sample Input 1

```
3 4
...E
.#..
....
```

Sample Output 1

```
>>>E
^#>^
>>>^
```

Let us verify the condition for $(2, 3)$ in the sample output. The distance from $(2, 3)$ to the nearest emergency exit is 2, and following the arrows written in the sample output leads to an emergency exit in two moves.
The condition can be verified for every other corridor cell as well.

Sample Input 2

```
3 2
##
##
##
```

Sample Output 2

```
##
##
##
```

There may be cases with no corridor cells or emergency exits.

Sample Input 3

```
7 20
....................
..#..#..####..#E##..
..#..#..#..#..#.....
..E###..#..#..####..
.....#..#..E.....#..
.....#..####..####..
....................
```

Sample Output 3

```
>v<<<<<>>>>>>>>v<<<<
>v#^<#^^####v^#E##vv
>v#^<#v^#>v#vv#^<<<<
>>E###vv#>v#vv####^<
>>^<<#vv#>>E<<<<<#^<
>>^<<#vv####^<####^<
>>^<<<<<>>>>^<<<<<^<
```

E - Fruit Lineup

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 475 points

Problem Statement

You have A apples, B oranges, C bananas, and D grapes.
How many ways are there to arrange these $A+B+C+D$ fruits in a single row from left to right so that all of the following conditions hold? Find the count modulo 998244353.

- Every apple is placed to the left of every banana.
- Every apple is placed to the left of every grape.
- Every orange is placed to the left of every grape.

Here, the apples are indistinguishable; the same goes for the oranges, the bananas, and the grapes.

Constraints

- $1 \leq A \leq 10^6$
- $1 \leq B \leq 10^6$
- $1 \leq C \leq 10^6$
- $1 \leq D \leq 10^6$
- A, B, C, and D are integers.

Input

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

$$A \quad B \quad C \quad D$$

Output

Output the number, modulo 998244353, of valid arrangements.

Sample Input 1

```
1 1 1 1
```

Sample Output 1

```
5
```

There are five valid arrangements:

- apple, orange, banana, grape
- apple, orange, grape, banana
- apple, banana, orange, grape
- orange, apple, banana, grape
- orange, apple, grape, banana

Sample Input 2

```
1 2 4 8
```

Sample Output 2

```
2211
```

Sample Input 3

```
834150 21994 467364 994225
```

Sample Output 3

```
947921688
```

F - Chord Crossing

Time Limit: 3 sec / Memory Limit: 1024 MB

Score : 525 points

Problem Statement

On a circle, $2N$ points are placed at equal intervals and numbered $1, 2, \ldots, 2N$ in clockwise order starting from an arbitrary point.

There are M line segments numbered $1, 2, \ldots, M$ connecting these points. Segment i connects points A_i and B_i. Here, A_i and B_i are distinct **even numbers**. It is guaranteed that no two of these segments share a point.

Process Q queries. The j-th query is as follows:

- You are given two distinct **odd numbers** C_j and D_j. Among the M segments $1, 2, \ldots, M$, find how many share a point with the segment connecting points C_j and D_j.

Constraints

- $2 \leq N \leq 10^6$
- $1 \leq M \leq \min\left(\lfloor \frac{N}{2} \rfloor, 2 \times 10^5\right)$
- $1 \leq Q \leq 2 \times 10^5$
- $1 \leq A_i < B_i \leq 2N$
- $1 \leq C_j < D_j \leq 2N$
- A_i and B_i are even.
- C_j and D_j are odd.
- For any i_1 and i_2 $(i_1 \neq i_2)$, segments i_1 and i_2 do not share a point.
- All input values are integers.

Input

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

$$
\begin{array}{l}
N \quad M \\
A_1 \quad B_1 \\
A_2 \quad B_2 \\
\vdots \\
A_M \quad B_M \\
Q \\
C_1 \quad D_1 \\
C_2 \quad D_2 \\
\vdots \\
C_Q \quad D_Q
\end{array}
$$

Output

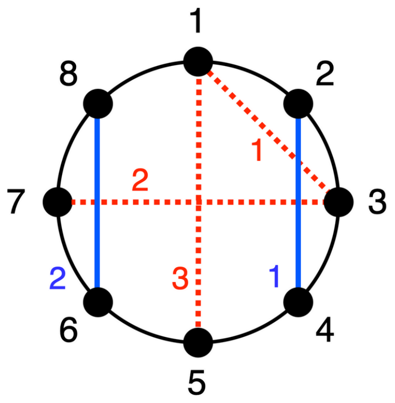
Output Q lines. The j-th line $(1 \leq j \leq Q)$ should contain the answer to the j-th query.

Sample Input 1

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

Sample Output 1

```
1
2
0
```



The figure above illustrates Sample Input 1. Black dots are the $2N\ (=8)$ points, blue solid lines are the initial $M\ (=2)$ segments, and red dashed lines are the $Q\ (=3)$ query segments.

- For the first query, the segment connecting points 1 and 3 intersects one initial segment: segment 1.
- For the second query, the segment connecting points 3 and 7 intersects two initial segments: segments 1 and 2.
- For the third query, the segment connecting points 1 and 5 intersects zero initial segments.

Sample Input 2

```
20 7
24 34
26 28
18 38
2 14
8 12
30 32
20 22
10
7 29
31 39
9 21
19 29
15 21
11 39
17 21
15 31
5 25
25 31
```

Sample Output 2

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

G - Range Shuffle Query

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 625 points

Problem Statement

You are given a length-N sequence $A = (A_1, A_2, \ldots, A_N)$.
Process Q queries.
Each query gives you integers L, R, X and asks you to solve the following.

> Let $B = (A_L, A_{L+1}, \ldots, A_R)$ be the sequence formed by the L-th through R-th elements of A.
> Perform the following procedure exactly once:
>
> - First, remove from B every element whose value is at least X.
> - Then, rearrange the remaining elements of B arbitrarily.
>
> How many distinct sequences B can result? Find the count modulo 998244353.

Constraints

- $1 \leq N \leq 2.5 \times 10^5$
- $1 \leq Q \leq 2.5 \times 10^5$
- $1 \leq A_i \leq N$
- $1 \leq L \leq R \leq N$
- $1 \leq X \leq N$
- All input values are integers.

Input

The input is given from Standard Input in the following format, where query_i denotes the i-th query:

$$
\begin{array}{l}
N \quad Q \\
A_1 \quad A_2 \quad \ldots \quad A_N \\
\mathrm{query}_1 \\
\mathrm{query}_2 \\
\vdots \\
\mathrm{query}_Q
\end{array}
$$

Each query is given in the following format:

$$L \quad R \quad X$$

Output

Output Q lines. The i-th line should contain the answer to the i-th query.

Sample Input 1

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

Sample Output 1

```
3
1
6
```

For the first query, there are three possible resulting sequences B: $(1, 1, 2)$, $(1, 2, 1)$, and $(2, 1, 1)$.
For the second query, there is one possible resulting sequence B: the empty sequence $()$.

Sample Input 2

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

Sample Output 2

```
1
120
6
3
1
2
```