A - Disjoint Set Union

Time Limit: 5 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

You are given an undirected graph with N vertices and 0 edges. Process Q queries of the following types.

- `0 u v`: Add an edge (u, v).
- `1 u v`: Print 1 if u and v are in the same connected component, 0 otherwise.

Constraints

- $1 \leq N \leq 200,000$
- $1 \leq Q \leq 200,000$
- $0 \leq u_i, v_i < N$

Input

Input is given from Standard Input in the following format:

$$\begin{array}{lll} N & Q & \\ t_1 & u_1 & v_1 \\ t_2 & u_2 & v_2 \\ \vdots & & \\ t_Q & u_Q & v_Q \end{array}$$

For each query of the latter type, print the answer.

Sample Input 1

```
4 7
1 0 1
0 0 1
0 2 3
1 0 1
1 1 2
0 0 2
1 1 3
```

Sample Output 1

```
0
1
0
1
```

Source Name

Based on Library Checker (Unionfind) (APACHE LICENSE, V2.0) (https://github.com/yosupo06/library-checker-problems)

B - Fenwick Tree

Time Limit: 5 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

You are given an array $a_0, a_1, ..., a_{N-1}$ of length N. Process Q queries of the following types.

- `0 p x`: $a_p \gets a_p + x$
- `1 l r`: Print $\sum_{i=l}^{r-1} a_i$.

Constraints

- $1 \leq N, Q \leq 500,000$
- $0 \leq a_i, x \leq 10^9$
- $0 \leq p < N$
- $0 \leq l_i < r_i \leq N$
- All values in Input are integer.

Input

Input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \quad Q \\
a_0 \quad a_1 \quad \ldots \quad a_{N-1} \\
\mathrm{Query}_0 \\
\mathrm{Query}_1 \\
: \\
\mathrm{Query}_{Q-1}
\end{array}
$$

Output

For each query of the latter type, print the answer.

Sample Input 1

```
5 5
1 2 3 4 5
1 0 5
1 2 4
0 3 10
1 0 5
1 0 3
```

Sample Output 1

```
15
7
25
6
```

Source Name

Based on Library Checker (Point Add Range Sum) (APACHE LICENSE, V2.0) (https://github.com/yosupo06/library-checker-problems)

C - Floor Sum

Time Limit: 5 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

In this problem, you should process T testcases.

For each testcase, you are given four integers N, M, A, B.

Calculate $\sum_{i=0}^{N-1} floor((A \times i + B)/M)$.

Constraints

- $1 \leq T \leq 100,000$
- $1 \leq N, M \leq 10^9$
- $0 \leq A, B < M$

Input

Input is given from Standard Input in the following format:

$$\begin{array}{l} T \\ N_0 \quad M_0 \quad A_0 \quad B_0 \\ N_1 \quad M_1 \quad A_1 \quad B_1 \\ : \\ N_{T-1} \quad M_{T-1} \quad A_{T-1} \quad B_{T-1} \end{array}$$

Output

Print the answer for each testcase.

Sample Input 1

```
5
4 10 6 3
6 5 4 3
1 1 0 0
31415 92653 58979 32384
1000000000 1000000000 999999999 999999999
```

Sample Output 1

```
3
13
0
314095480
499999999500000000
```

Source Name

Based on Library Checker (Sum of Floor of Linear) (APACHE LICENSE, V2.0) (https://github.com/yosupo06/library-checker-problems)

D - Maxflow

Time Limit: 5 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

You are given a grid of N rows and M columns. The square at the i-th row and j-th column will be denoted as (i, j). Some of the squares contain an object. All the remaining squares are empty. The state of the grid is represented by strings $S_1, S_2, \cdots, S_N$. The square (i, j) contains an object if $S_{i,j} =$ # and is empty if $S_{i,j} = .$.

Consider placing 1×2 tiles on the grid. Tiles can be placed vertically or horizontally to cover two adjacent empty squares. Tiles must not stick out of the grid, and no two different tiles may intersect. Tiles cannot occupy the square with an object.

Calculate the maximum number of tiles that can be placed and any configulation that acheives the maximum.

Constraints

- $1 \leq N \leq 100$
- $1 \leq M \leq 100$
- S_i is a string with length M consists of # and $.$.

Input

Input is given from Standard Input in the following format:

$$N \quad M$$
$$S_1$$
$$S_2$$
$$\vdots$$
$$S_N$$

Output

On the first line, print the maximum number of tiles that can be placed.

On the next N lines, print a configulation that achieves the maximum. Precisely, output the strings $t_1, t_2, \cdots, t_N$ constructed by the following way.

- t_i is initialized to S_i.
- For each (i, j), if there is a tile that occupies (i, j) and $(i + 1, j)$, change $t_{i,j}$:=`v`, $t_{i+1,j}$:=`^`.
- For each (i, j), if there is a tile that occupies (i, j) and $(i, j + 1)$, change $t_{i,j}$:=`>`, $t_{i,j+1}$:=`<`.

See samples for further information.

You may print any configulation that maximizes the number of tiles.

Sample Input 1

```
3 3
#..
..#
...
```

Sample Output 1

```
3
#><
vv#
^^.
```

The following output is also treated as a correct answer.

```
3
#><
v.#
^><
```

E - MinCostFlow

Time Limit: 5 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

You are given a grid of N rows and M columns. The square at the i-th row and j -th column will be denoted as (i, j). A nonnegative integer $A_{i,j}$ is written for each square (i, j).

You choose some of the squares so that each row and column contains at most K chosen squares. Under this constraint, calculate the maximum value of the sum of the integers written on the chosen squares. Additionally, calculate a way to choose squares that acheives the maximum.

Constraints

- $1 \leq N \leq 50$
- $1 \leq K \leq N$
- $0 \leq A_{i,j} \leq 10^9$
- All values in Input are integer.

Input

Input is given from Standard Input in the following format:

$$
\begin{array}{llll}
N & K & & \\
A_{1,1} & A_{1,2} & \cdots & A_{1,N} \\
A_{2,1} & A_{2,2} & \cdots & A_{2,N} \\
\vdots & & & \\
A_{N,1} & A_{N,2} & \cdots & A_{N,N}
\end{array}
$$

Output

On the first line, print the maximum value of the sum of the integers written on the chosen squares.

On the next N lines, print a way that achieves the maximum.

Precisely, output the strings $t_1, t_2, \cdots, t_N$, that satisfies $t_{i,j} =$`X` if you choose (i, j) and $t_{i,j} =$`.` otherwise.

You may print any way to choose squares that maximizes the sum.

Sample Input 1

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

Sample Output 1

```
19
X..
..X
.X.
```

Sample Input 2

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

Sample Output 2

```
50
XX.
XX.
..X
```

F - Convolution

Time Limit: 5 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

You are given two integer arrays $a_0, a_1, ..., a_{N-1}$ and $b_0, b_1, ..., b_{M-1}$. Calculate the array $c_0, c_1, ..., c_{(N-1)+(M-1)}$, defined by $c_i = \sum_{j=0}^{i} a_j b_{i-j} \bmod 998244353$.

Constraints

- $1 \leq N, M \leq 524288$
- $0 \leq a_i, b_i < 998244353$
- All values in Input are integer.

Input

Input is given from Standard Input in the following format:

$N \quad M$
$a_0 \quad a_1 \quad \ldots \quad a_{N-1}$
$b_0 \quad b_1 \quad \ldots \quad b_{M-1}$

Output

Print the answer in the following format:

$c_0 \quad c_1 \quad \ldots \quad c_{(N-1)+(M-1)}$

Sample Input 1

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

Sample Output 1

```
5 16 34 60 70 70 59 36
```

Sample Input 2

```
1 1
10000000
10000000
```

Sample Output 2

```
871938225
```

Source Name

Based on Library Checker (Convolution) (APACHE LICENSE, V2.0) (https://github.com/yosupo06/library-checker-problems)

G - SCC

Time Limit: 5 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

You are given a directed graph with N vertices and M edges, not necessarily simple. The i-th edge is oriented from the vertex a_i to the vertex b_i. Divide this graph into strongly connected components and print them in their topological order.

Constraints

- $1 \leq N \leq 500,000$
- $1 \leq M \leq 500,000$
- $0 \leq a_i, b_i < N$

Input

Input is given from Standard Input in the following format:

$N \quad M$
$a_0 \quad b_0$
$a_1 \quad b_1$
$\vdots$
$a_{M-1} \quad b_{M-1}$

Output

Print $1 + K$ lines, where K is the number of strongly connected components. Print K on the first line. Print the information of each strongly connected component in next K lines in the following format, where l is the number of vertices in the strongly connected component and v_i is the index of the vertex in it.

$$l \quad v_0 \quad v_1 \quad \ldots \quad v_{l-1}$$

Here, for each edge (a_i, b_i), b_i should not appear in **earlier** line than a_i.

If there are multiple correct output, print any of them.

Sample Input 1

```
6 7
1 4
5 2
3 0
5 5
4 1
0 3
4 2
```

Sample Output 1

```
4
1 5
2 4 1
1 2
2 3 0
```

Source Name

Based on Library Checker (Strongly Connected Components) (APACHE LICENSE, V2.0) (https://github.com/yosupo06/library-checker-problems)

H - Two SAT

Time Limit: 5 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

Consider placing N flags on a line. Flags are numbered through 1 to N.

Flag i can be placed on the coordinate X_i or Y_i. For any two different flags, the distance between them should be at least D.

Decide whether it is possible to place all N flags. If it is possible, print such a configulation.

Constraints

- $1 \leq N \leq 1000$
- $0 \leq D \leq 10^9$
- $0 \leq X_i < Y_i \leq 10^9$
- All values in Input are integer.

Input

Input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \quad D \\
X_1 \quad Y_1 \\
X_2 \quad Y_2 \\
\vdots \\
X_N \quad Y_N
\end{array}
$$

Output

Print `No` if it is impossible to place N flags.

If it is possible, print `Yes` first. After that, print N lines. i-th line of them should contain the coodinate of flag i.

Sample Input 1

```
3 2
1 4
2 5
0 6
```

Sample Output 1

```
Yes
4
2
0
```

Sample Input 2

```
3 3
1 4
2 5
0 6
```

Sample Output 2

```
No
```

I - Number of Substrings

Time Limit: 5 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

You are given a string of length N. Calculate the number of distinct substrings of S

.

Constraints

- $1 \leq N \leq 500,000$
- S consists of lowercase English letters.

Input

Input is given from Standard Input in the following format:

S

Output

Print the answer.

Sample Input 1

```
abcbcba
```

Sample Output 1

```
21
```

Sample Input 2

```
mississippi
```

Sample Output 2

```
53
```

Sample Input 3

```
ababacaca
```

Sample Output 3

```
33
```

Sample Input 4

```
aaaaa
```

Sample Output 4

```
5
```

Source Name

Based on Library Checker (Number of Substrings) (APACHE LICENSE, V2.0) (https://github.com/yosupo06/library-checker-problems)

J - Segment Tree

Time Limit: 5 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

You are given an array $a_0, a_1, ..., a_{N-1}$ of length N. Process Q queries of the following types.

The type of i-th query is represented by T_i.

- $T_i = 1$: You are given two integers X_i, V_i. Replace the value of A_{X_i} with V_i.
- $T_i = 2$: You are given two integers L_i, R_i. Calculate the maximum value among $A_{L_i}, A_{L_i+1}, \cdots, A_{R_i}$.
- $T_i = 3$: You are given two integers X_i, V_i. Calculate the minimum j such that $X_i \leq j \leq N, V_i \leq A_j$. If there is no such j, answer $j = N + 1$ instead.

Constraints

- $1 \leq N \leq 2 \times 10^5$
- $0 \leq A_i \leq 10^9$
- $1 \leq Q \leq 2 \times 10^5$
- $1 \leq T_i \leq 3$
- $1 \leq X_i \leq N, 0 \leq V_i \leq 10^9$ ($T_i = 1, 3$)
- $1 \leq L_i \leq R_i \leq N$ ($T_i = 2$)
- All values in Input are integer.

Input

Input is given from Standard Input in the following format:

N Q
A_1 A_2 $\cdots$ A_N
`First query`
`Second query`
$\vdots$
Q`-th query`

Each query is given in the following format:

If $T_i = 1, 3$,

T_i X_i V_i

If $T_i = 2$,

T_i L_i R_i

Output

For each query with $T_i = 2, 3$, print the answer.

Sample Input 1

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

Sample Output 1

```
3
3
2
6
```

- First query: Print 3, which is the maximum of $(A_1, A_2, A_3, A_4, A_5) = (1, 2, 3, 2, 1)$.
- Second query: Since $3 > A_2$, $j = 2$ does not satisfy the condition．Since $3 \leq A_3$, print $j = 3$.
- Third query: Replace the value of A_3 with 1. It becomes $A = (1, 2, 1, 2, 1)$.
- Fourth query: Print 2, which is the maximum of $(A_2, A_3, A_4) = (2, 1, 2)$.
- Fifth query: Since there is no j that satisfies the condition, print $j = N + 1 = 6$.

K - Range Affine Range Sum

Time Limit: 5 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

You are given an array $a_0, a_1, ..., a_{N-1}$ of length N. Process Q queries of the following types.

- `0 l r b c`: For each $i = l, l+1, \dots, r-1$, set $a_i \gets b \times a_i + c$.
- `1 l r`: Print $\sum_{i=l}^{r-1} a_i \bmod 998244353$.

Constraints

- $1 \leq N, Q \leq 500000$
- $0 \leq a_i, c < 998244353$
- $1 \leq b < 998244353$
- $0 \leq l < r \leq N$
- All values in Input are integer.

Input

Input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \quad Q \\
a_0 \quad a_1 \quad \ldots \quad a_{N-1} \\
\textrm{Query}_0 \\
\textrm{Query}_1 \\
: \\
\textrm{Query}_{Q-1}
\end{array}
$$

Output

For each query of the latter type, print the answer.

Sample Input 1

```
5 7
1 2 3 4 5
1 0 5
0 2 4 100 101
1 0 3
0 1 3 102 103
1 2 5
0 2 5 104 105
1 0 5
```

Sample Output 1

```
15
404
41511
4317767
```

Source Name

Based on Library Checker (Range Affine Range Sum) (APACHE LICENSE, V2.0) (https://github.com/yosupo06/library-checker-problems)

L - Lazy Segment Tree

Time Limit: 5 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

You are given a binary array $A = (A_1, A_2, \cdots, A_N)$ of length N.

Process Q queries of the following types. The i-th query is represented by three integers T_i, L_i, R_i.

- $T_i = 1$: Replace the value of A_j with $1 - A_j$ for each $L_i \leq j \leq R_i$.
- $T_i = 2$: Calculate the inversion(*) of the array $A_{L_i}, A_{L_i+1}, \cdots, A_{R_i}$．

Note：The inversion of the array $x_1, x_2, \cdots, x_k$ is the number of the pair of integers i, j with $1 \leq i < j \leq k, x_i > x_j$.

Constraints

- $1 \leq N \leq 2 \times 10^5$
- $0 \leq A_i \leq 1$
- $1 \leq Q \leq 2 \times 10^5$
- $1 \leq T_i \leq 2$
- $1 \leq L_i \leq R_i \leq N$
- All values in Input are integer.

Input

Input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \quad Q \\
A_1 \quad A_2 \quad \cdots \quad A_N \\
T_1 \quad L_1 \quad R_1 \\
T_2 \quad L_2 \quad R_2 \\
\vdots \\
T_Q \quad L_Q \quad R_Q
\end{array}
$$

Output

For each query with $T_i = 2$, print the answer.

Sample Input 1

```
5 5
0 1 0 0 1
2 1 5
1 3 4
2 2 5
1 1 3
2 1 2
```

Sample Output 1

```
2
0
1
```

- First query: Print 2, which is the inversion of $(A_1, A_2, A_3, A_4, A_5) = (0, 1, 0, 0, 1)$.
- Second query: Replace the value of A_3 and A_4 with 1 and 1, respectively.
- Third query: Print 0, which is the inversion of $(A_2, A_3, A_4, A_5) = (1, 1, 1, 1)$.
- Fourth query: Replace the value of A_1, A_2 and A_4 with $1, 0$ and 0, respectively.
- Fifth query: Print 1, which is the inversion of $(A_1, A_2) = (1, 0)$.