Problem A. Depth of Interval

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

You are given a positive integer N and a permutation $P = (P_1, P_2, \ldots, P_N)$ of $(1, 2, \ldots, N)$.

For an integer pair (L, R), we define the value $f(L, R)$ recursively as follows.

- If $1 \leq L < R \leq N$: Let integers a and b be such that, among $P_L, P_{L+1}, \ldots, P_R$, the smallest element and the second smallest element are P_a and P_b, respectively. Then we define

$$f(L, R) = f(\min(a, b) + 1, \max(a, b) - 1) + 1.$$

- Otherwise, we define $f(L, R) = 0$.

For each $k = 1, 2, \ldots, N$, find the number of integer pairs (L, R) such that $f(L, R) = k$.

Input

The input is given in the following format:

N
$P_1\ P_2\ \cdots\ P_N$

- All input values are integers.
- $2 \leq N \leq 3 \times 10^5$
- $(P_1, P_2, \ldots, P_N)$ is a permutation of $(1, 2, \ldots, N)$

Output

Print N lines. For each $k = 1, 2, \ldots, N$, print the number of integer pairs (L, R) satisfying $f(L, R) = k$ on the k-th line.

Examples

standard input	standard output
7 2 6 5 1 4 7 3	14 7 0 0 0 0 0
5 1 2 3 4 5	10 0 0 0 0
9 8 6 2 4 9 7 3 5 1	25 8 3 0 0 0 0 0 0

Note

In the first example, the value $f(1,7)$ is computed as follows. Among $P_1, P_2, \ldots, P_7$, the smallest element and the second smallest element are P_4 and P_1, respectively. Thus, $f(1,7) = f(2,3) + 1$.

Next, among P_2, P_3, the smallest element and the second smallest element are P_3 and P_2, respectively, so $f(2,3) = f(3,2) + 1$.

Since $f(3,2) = 0$, we have $f(2,3) = 1$, and therefore $f(1,7) = 2$.

Problem B. Increasing Swaps

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

You are given a permutation $P = (P_1, P_2, \ldots, P_N)$ of $(1, 2, \ldots, N)$.

For a sequence of positive integers $T = (T_1, T_2, \ldots, T_{N-1})$ of length $N - 1$, the value $f(T)$ is defined as the return value of the following procedure.

Procedure $f(T)$

1. $t \leftarrow 0$.
2. While P is not sorted in ascending order:
   (a) $t \leftarrow t + 1$.
   (b) Let $S = (s_1, s_2, \ldots, s_k)$ be the sequence of all indices $i \in \{1, \ldots, N-1\}$ such that $T_i \leq t$, sorted in increasing order ($s_1 < s_2 < \cdots < s_k$).
   (c) For $j = 1$ to k:
      - Let $i \leftarrow s_j$. Then swap P_i and P_{i+1}.
   (d) If $t \geq 10^{100}$, return 10^{100}.
3. Return t.

Find the minimum possible value of $f(T)$ over all sequences T of positive integers. It is guaranteed that there exists a sequence T such that $f(T) < 10^{100}$.

Input

The input is given in the following format:

N
$P_1\ P_2\ \cdots\ P_N$

- All input values are integers.
- $2 \leq N \leq 5000$.
- $1 \leq P_i \leq N$.
- $P_i \neq P_j$ for $i \neq j$.

Output

Output the answer.

Examples

standard input	standard output
4 4 2 1 3	2
20 15 13 7 3 4 8 16 12 2 5 1 17 11 18 9 19 20 10 6 14	39

Note

In the first example, let $T = (2, 1, 2)$. At $t = 1$, with $S = (2)$, P becomes $(4, 1, 2, 3)$. At $t = 2$, with $S = (1, 2, 3)$, sequential swaps change P as $(4, 1, 2, 3) \to (1, 4, 2, 3) \to (1, 2, 4, 3) \to (1, 2, 3, 4)$. Thus $f(T) = 2$. Since no T yields $f(T) < 2$, the answer is 2.

Problem C. Sum of Three Inversions

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes

You are given integers N, X, Y, K, and M. Find the number of triples (A, B, C) of integer sequences of length N that satisfy all of the following conditions, modulo M.

- For each $i = 1, 2, \ldots, N$, the tuple (A_i, B_i, C_i) is a permutation of $(1, 2, 3)$.
- The sequence A contains exactly X occurrences of 1 and Y occurrences of 2.
- The sum of the number of inversions of A, B, and C is equal to K.

Here, the **number of inversions** of a sequence a is defined as the number of pairs of integers (i, j) satisfying $1 \leq i < j \leq |a|$ and $a_i > a_j$.

Input

The input is given in the following format:

N X Y K M

- All input values are integers.
- $2 \leq N \leq 50$.
- $0 \leq X, Y \leq N$.
- $X + Y \leq N$.
- $0 \leq K \leq \frac{3}{2}N(N-1)$.
- $10^8 \leq M \leq 10^9$.

Output

Print the answer.

Examples

standard input	standard output
3 1 1 4 998244353	24
4 0 0 18 123456789	0
50 10 20 1000 1000000000	805988728

Note

In the first example, there are 24 triples (A, B, C) that satisfy the conditions. For example, if we take $A = (1, 2, 3)$, $B = (3, 3, 2)$, and $C = (2, 1, 1)$, then (A, B, C) satisfies all the conditions.

Problem D. Grid Path Tree

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes

You are given positive integers N and M.

For each $k = 1, 2, \ldots, N + M - 1$, let ans_k denote the answer to the following problem.

A pair of sequences (a, b), where $a = (a_1, a_2, \ldots, a_{N+M-1})$ and $b = (b_1, b_2, \ldots, b_{N+M-1})$, each of length $N + M - 1$, is called a **good pair of sequences** if all of the following conditions are satisfied:

- $(a_1, b_1) = (1, N + 1)$
- For $i = 2, 3, \ldots, N + M - 1$, one of the following holds:
  - $(a_i, b_i) = (a_{i-1} + 1, b_{i-1})$
  - $(a_i, b_i) = (a_{i-1}, b_{i-1} + 1)$
- $(a_{N+M-1}, b_{N+M-1}) = (N, N + M)$

For a **good pair of sequences** (a, b), define a tree $T(a, b)$ as follows:

- It is a tree with $N + M$ vertices labeled from 1 to $N + M$.
- For each $i = 1, 2, \ldots, N + M - 1$, there is an edge connecting vertex a_i and vertex b_i.

For a tree, let $\mathrm{dist}(i, j)$ be the number of edges on the simple path between vertices i and j. The **score** of the tree is defined as the number of integer pairs (i, j) such that $1 \le i < j \le N + M$ and $\mathrm{dist}(i, j) = k$.
Compute the sum of the **scores** of $T(a, b)$ over all **good pairs of sequences** (a, b), modulo 998244353.

Compute all values $\mathrm{ans}_1, \mathrm{ans}_2, \ldots, \mathrm{ans}_{N+M-1}$, and output $\sum_{k=1}^{N+M-1} (\mathrm{ans}_k \oplus k)$, where $\oplus$ denotes the bitwise exclusive OR (XOR).

Input

The input is given in the following format:

N M

- $1 \le N \le 5 \times 10^6$
- $1 \le M \le 5 \times 10^6$
- All input values are integers.

Output

Output the answer.

Examples

standard input	standard output
2 2	14
24 167	21925979855
4297614 4167924	4162418864110099

Note

In the first example, $(\mathrm{ans}_1, \mathrm{ans}_2, \mathrm{ans}_3) = (6, 4, 2)$. Therefore, the value to be output is

$$(6 \oplus 1) + (4 \oplus 2) + (2 \oplus 3) = 7 + 6 + 1 = 14.$$

Problem E. Max Twice Subsequences

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

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

Answer Q queries. In the i-th query, integers L_i, R_i are given. Let $B = (A_{L_i}, A_{L_i+1}, \ldots, A_{R_i})$. For this sequence B, solve the following problem and output the answer.

You are given a sequence of positive integers $B = (B_1, B_2, \ldots, B_{|B|})$. A positive integer sequence $C = (C_1, C_2, \ldots, C_k)$ is called a **good sequence** if it appears at least twice as a non-empty subsequence of B. Formally, this means that there exist a positive integer k and two sequences of indices $(i_1, i_2, \ldots, i_k)$ and $(j_1, j_2, \ldots, j_k)$ such that all of the following conditions hold:

- $1 \le i_1 < i_2 < \cdots < i_k \le |B|$
- $1 \le j_1 < j_2 < \cdots < j_k \le |B|$
- For all $p = 1, 2, \ldots, k$, we have $B_{i_p} = B_{j_p} = C_p$
- There exists at least one p $(1 \le p \le k)$ such that $i_p \ne j_p$

Determine whether a good sequence exists. If it exists, output the **rolling hash** of the lexicographically largest good sequence. Otherwise, output -1.

The **rolling hash** of a positive integer sequence $a = (a_1, a_2, \ldots, a_n)$ is defined as $\left(\sum_{i=1}^{n} a_i 3^{i-1}\right) \bmod 998244353$.

Input

The input is given in the following format:

N Q
A_1 A_2 $\ldots$ A_N
L_1 R_1
L_2 R_2
$\vdots$
L_Q R_Q

- All input values are integers.
- $1 \le N, Q \le 3 \times 10^5$
- $1 \le A_i \le N$ $(1 \le i \le N)$
- $1 \le L_i \le R_i \le N$ $(1 \le i \le Q)$

Output

Print Q lines. For each $i = 1, 2, \ldots, Q$, output the answer for the sequence $B = (A_{L_i}, A_{L_i+1}, \ldots, A_{R_i})$.

Examples

standard input	standard output
5 4 3 2 1 2 3 1 5 1 3 2 4 2 5	36 -1 2 11
5 1 3 2 1 4 1 1 5	18
10 10 1 4 5 2 5 3 4 2 5 4 1 10 1 9 2 10 1 8 2 9 3 10 1 7 2 8 3 9 4 10	56 20 56 35 20 56 17 35 20 17

Note

In the first sample, the behavior is as follows.

- For the first query, $B = (3, 2, 1, 2, 3)$. Among all non-empty subsequences that appear at least twice, the lexicographically largest one is $(3, 2, 3)$. Two valid index sequences that produce this subsequence are $(1, 2, 5)$ and $(1, 4, 5)$.
- For the second query, $B = (3, 2, 1)$. There is no non-empty subsequence that appears at least twice.
- For the third query, $B = (2, 1, 2)$. Among all non-empty subsequences that appear at least twice, the lexicographically largest one is (2).
- For the fourth query, $B = (2, 1, 2, 3)$. Among all non-empty subsequences that appear at least twice, the lexicographically largest one is $(2, 3)$.

Problem F. Decimal Pyramid

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes

You are given a string S of length N consisting of the digits `1`, `2`, ..., `9`.

Consider a triangular pyramid made up of a total of $\frac{N(N+1)}{2}$ blocks.

The pyramid is divided into N layers, numbered $1, 2, \ldots, N$ from top to bottom. Layer i ($1 \le i \le N$) contains i blocks arranged in a single horizontal row from left to right. Each block has a string written on it. Let $C_{i,j}$ denote the string written on the j-th block from the left ($1 \le j \le i$) in layer i.

The strings $C_{i,j}$ satisfy the following conditions:

- If $i = N$, then $C_{i,j}$ is the string of length 1 consisting of the j-th character of S.
- If $1 \le i < N$, then $C_{i,j}$ is the concatenation of $C_{i+1,j}$ and $C_{i+1,j+1}$, in this order.

Interpret $C_{1,1}$ as a decimal integer, and compute its value modulo 998244353.

Input

The input is given in the following format:

N
S

- N is an integer.
- $1 \le N \le 2 \times 10^5$.
- S is a string of length N consisting of the digits `1`, `2`, ..., `9`.

Output

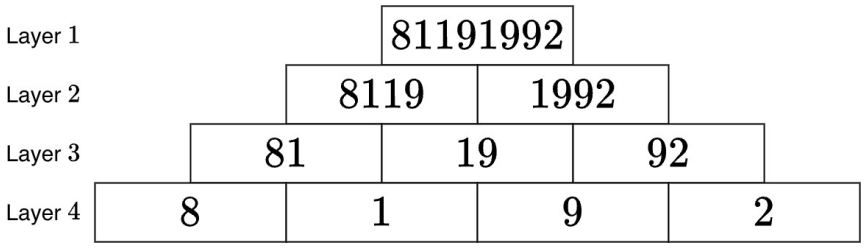
Output the answer.

Examples

standard input	standard output
4 8192	81191992
1 5	5
14 11123455678999	913063116

Note

In the first example, S = `8192`. By constructing the pyramid according to the rules, we obtain $C_{1,1}$ = `81191992`.



In the second example, $S = 5$. The pyramid consists of a single block, and $C_{1,1} = 5$.

Problem G. Don't Make Zero

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

This is an **interactive problem** (a problem where your program and the judge system interact via input and output).

An integer sequence is called a **non-zero sequence** if it satisfies **both** of the following conditions:

- Any two elements are distinct.
- The sum of any non-empty (not necessarily contiguous) subsequence is not equal to 0.

For example, (5), $(1, -2, 3)$, and $(-3, 7, 5, -6)$ are non-zero sequences, while (0), $(1, -3, 1)$, $(2, 3, -2)$, and $(1, 2, 3, -4)$ are not.

You are given positive integers N and X.

Output, one by one from the beginning, the elements of a non-zero sequence $(A_1, A_2, \ldots, A_X)$ of length X $(= 2\lfloor\sqrt{N}\rfloor - 1)$, consisting of integers between $-N$ and N (inclusive). However, the sign of each element is specified immediately before it is output.

You are given R test cases; interact with the judge for each of them.

Input

- $1 \le R \le 10^4$
- $1 \le N \le 10^4$
- $X = 2\lfloor\sqrt{N}\rfloor - 1$
- The sum of N over all test cases does not exceed 10^4

Interaction Protocol

First, the number of test cases R is given in the following format:

R

Then the following interaction is repeated R times.

For each test case, positive integers N and X are given in the following format:

N X

After that, for each $i = 1, 2, \ldots, X$ in this order, perform the following interaction.

First, the sign op_i of A_i is given in the following format:

op_i

op_i is one of `+` or `-`, with the following meanings:

- If $\mathrm{op}_i =$ `+`, then A_i must be a positive integer.
- If $\mathrm{op}_i =$ `-`, then A_i must be a negative integer.

Then, output an integer A_i between $-N$ and N (inclusive) with the specified sign, on one line (you do not need to print a sign for positive integers):

A_i

Note

After each output, your program must flush standard output; Otherwise, you will receive Time Limit Exceeded.

Sample Interaction

input	output	explanation
`2`		The number of test cases R is given.
`4 3`		N, X for the first test case are given.
`-`		Since $\mathrm{op}_1 =$ `-`, A_1 must be a negative integer.
	`-4`	You output $A_1 = -4$.
`-`		Since $\mathrm{op}_2 =$ `-`, A_2 must be a negative integer.
	`-1`	You output $A_2 = -1$.
`+`		Since $\mathrm{op}_3 =$ `+`, A_3 must be a positive integer.
	`2`	You output $A_3 = 2$. Since $(A_1, A_2, A_3) = (-4, -1, 2)$ is a non-zero sequence, this test case is considered correct.
`3 1`		N, X for the second test case are given.
`+`		Since $\mathrm{op}_1 =$ `+`, A_1 must be a positive integer.
	`3`	You output $A_1 = 3$. Since $(A_1) = (3)$ is a non-zero sequence, this test case is considered correct.

Problem H. Akari Counting

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

You are given integers H, W, A, B, C, D.

There is a grid with H rows and W columns. The cell in the i-th row from the top and the j-th column from the left is called cell (i, j).

Each cell is colored either white or black. Cell (i, j) is black if and only if $A \leq i \leq B$ and $C \leq j \leq D$; otherwise, it is white.

You place lights on some of the white cells of this grid. A light placed on a white cell (i, j) **illuminates** all white cells that satisfy both of the following conditions:

- They are in the same row or the same column as cell (i, j).
- There is no black cell between cell (i, j) and that cell.

A placement of lights is said to be **valid** if it satisfies the following two conditions:

- Every white cell is illuminated by at least one light.
- No cell that has a light is illuminated by any other light.

Find the number of valid placements of lights, modulo 998244353.

Input

The input is given in the following format:

H W A B C D

- All input values are integers.
- $1 \leq A \leq B \leq H \leq 5 \times 10^5$
- $1 \leq C \leq D \leq W \leq 5 \times 10^5$
- $(A, B) \neq (1, H)$
- $(C, D) \neq (1, W)$

Output

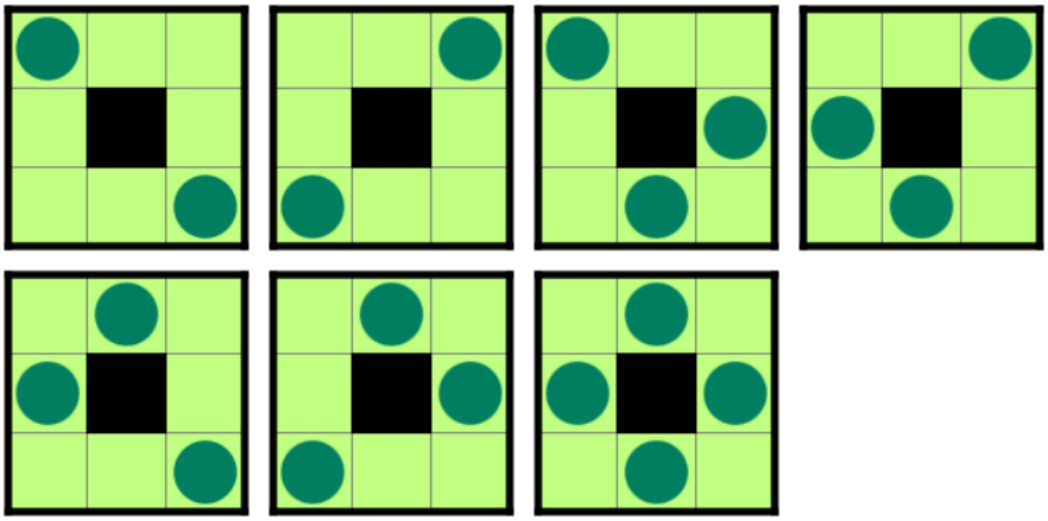
Output the answer.

Examples

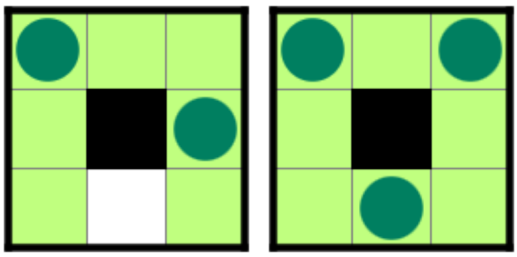
standard input	standard output
3 3 2 2 2 2	7
2 3 1 1 1 2	3
500000 500000 100000 200000 100000 250000	360665510

Note

In the first example, there are exactly 7 valid placements of lights. Cells with lights and white cells illuminated by lights are shown in green in the figures.



Some placements do not satisfy the conditions: in the left case, there exists at least one white cell that is not illuminated by any light; in the right case, there is a white cell with a light that is illuminated by another light. Both cases are invalid.



Problem I. Subgrid Connected Components

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2.5 seconds
Memory limit:	384 megabytes

Please note that the memory limit for this problem is 384 MiB.

You are given a grid with $2N + 1$ rows and $2N + 1$ columns, where N is a positive integer. Let the cell in the i-th row from the top and the j-th column from the left ($1 \leq i, j \leq 2N + 1$) be denoted by (i, j). Each cell (i, j) contains a character $S_{i,j}$, which satisfies the following properties:

- If i is odd and j is odd, then $S_{i,j} =$ `o`
- If i is odd and j is even, then $S_{i,j} =$ `-` or `.`
- If i is even and j is odd, then $S_{i,j} =$ `|` or `.`
- If i is even and j is even, then $S_{i,j} =$ `.`

You are given Q independent queries. In the i-th query ($1 \leq i \leq Q$), odd integers U_i, D_i, L_i, R_i ($1 \leq U_i \leq D_i \leq 2N + 1,\ 1 \leq L_i \leq R_i \leq 2N + 1$) are given. For each query, answer the following question.

> In the subgrid $[U_i, D_i] \times [L_i, R_i]$, treat the character `o` as vertices and the characters `-` and `|` as edges. How many connected components does the resulting undirected graph have?

More precisely, answer the following.

> Consider an undirected graph G with $((D_i - U_i + 2)/2) \times ((R_i - L_i + 2)/2)$ vertices. Each vertex corresponds to a pair (x, y), where x is an odd integer with $U_i \leq x \leq D_i$ and y is an odd integer with $L_i \leq y \leq R_i$.
> The undirected graph G has the following edges, and no other edges:
>
> - If odd integers x, y satisfy $U_i \leq x \leq D_i$, $L_i \leq y \leq R_i - 2$, and $S_{x,y+1} =$ `-`, then there is an undirected edge between vertices (x, y) and $(x, y + 2)$.
> - If odd integers x, y satisfy $U_i \leq x \leq D_i - 2$, $L_i \leq y \leq R_i$, and $S_{x+1,y} =$ `|`, then there is an undirected edge between vertices (x, y) and $(x + 2, y)$.
>
> Find the number of connected components of the undirected graph G.

Input

The input is given in the following format:

N
$S_{1,1}S_{1,2} \ldots S_{1,2N+1}$
$S_{2,1}S_{2,2} \ldots S_{2,2N+1}$
$\vdots$
$S_{2N+1,1}S_{2N+1,2} \ldots S_{2N+1,2N+1}$
Q
$U_1\ D_1\ L_1\ R_1$
$U_2\ D_2\ L_2\ R_2$
$\vdots$
$U_Q\ D_Q\ L_Q\ R_Q$

- N is an integer satisfying $1 \leq N \leq 2000$.
- If i is odd and j is odd, then $S_{i,j} =$ `o`
- If i is odd and j is even, then $S_{i,j} =$ `-` or `.`
- If i is even and j is odd, then $S_{i,j} =$ `|` or `.`
- If i is even and j is even, then $S_{i,j} =$ `.`
- Q is an integer satisfying $1 \leq Q \leq 7000$.
- $1 \leq U_i \leq D_i \leq 2N + 1$
- $1 \leq L_i \leq R_i \leq 2N + 1$
- U_i, D_i, L_i, R_i are odd integers.

Output

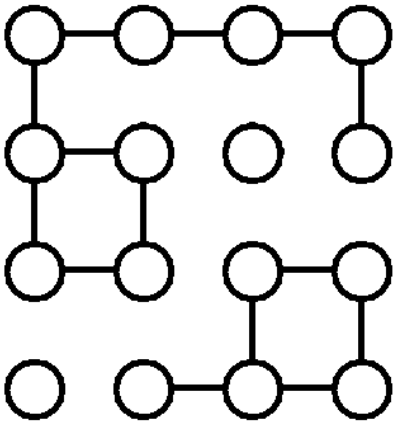
Output Q lines. For $i = 1, 2, \ldots, Q$, output the answer to the i-th query on the i-th line.

Example

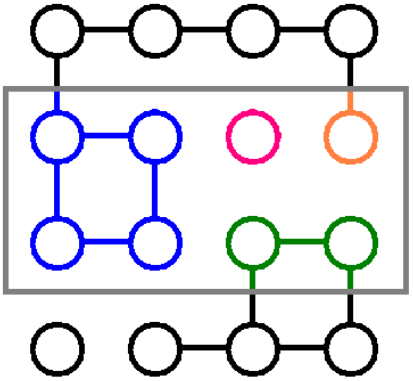
standard input	standard output
3 o-o-o-o |.....| o-o.o.o |.|.... o-o.o-o ....|.| o.o-o-o 12 3 5 1 7 1 1 1 1 1 3 1 3 1 3 1 7 1 1 1 1 1 1 1 7 1 7 1 1 1 7 1 7 3 5 3 5 3 5 3 7 3 7 3 7 5 7 3 5	4 1 1 2 1 1 2 4 3 4 4 2

Note

The given grid looks as follows:



The answer to the first query is 4, as shown in the following figure.



Problem J. Divide Polygon

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

You are given a positive integer N and a set of integers $S = \{S_1, S_2, \ldots, S_M\}$ of size M.

For each $k = 0, 1, \ldots, N-3$, answer the following question.

> Consider a regular N-gon whose vertices are labeled $1, 2, \ldots, N$. Draw exactly k diagonals so that no two diagonals intersect except possibly at their endpoints. As a result, the regular N-gon is divided into $k+1$ polygons. Let $e_1, e_2, \ldots, e_{k+1}$ be the numbers of sides of these resulting polygons.
> We say that a way of drawing k diagonals is a **good way** if it satisfies the following condition:
>
> - All of $e_1, e_2, \ldots, e_{k+1}$ are contained in the set S.
>
> Compute the number of **good ways** to draw k diagonals, modulo 998244353.

Input

The input is given in the following format:

> N M
> S_1 S_2 $\ldots$ S_M

- All input values are integers.
- $3 \leq N \leq 10^5$
- $1 \leq M \leq N-2$
- $3 \leq S_i \leq N$
- $S_i < S_{i+1}$

Output

Output $N-2$ lines. For $i = 1, 2, \ldots, N-2$, output on the i-th line the answer corresponding to $k = i-1$.

Examples

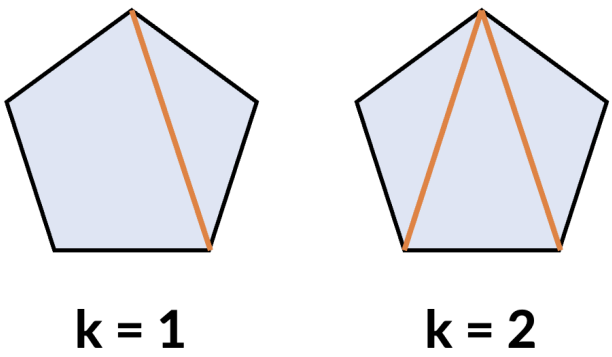
standard input	standard output
5 2 3 4	0 5 5
4 1 4	1 0
16 7 3 4 6 7 9 12 16	1 24 544 14280 120156 829464 3372120 10914816 24515700 40532624 52300160 42493880 17383860 2674440

Note

In the first example, when $k = 0$, we always have $e_1 = 5$. Since 5 is not contained in S, the answer is 0.

When $k = 1$, we always have $\{e_1, e_2\} = \{3, 4\}$, and both values are contained in S. There are 5 ways to draw one diagonal in a regular pentagon, so the answer is 5.

When $k = 2$, we always have $e_i = 3$ $(1 \le i \le 3)$. There are 5 ways to draw two non-intersecting diagonals in a regular pentagon, so the answer is 5.



Problem K. Two-Way Communication

Input file:	`standard input`
Output file:	`standard output`
Time limit:	0.5 seconds
Memory limit:	1024 megabytes

This is a communication problem.

Two programs, Alice and Bob, play the following cooperative game.

You are given integers A and B with $0 \le A, B < 2^{64}$. Initially, only Alice knows A, and only Bob knows B. Let $C := \min(A, B)$. The goal is for **both** Alice and Bob to be able to output C.
To exchange information, Alice and Bob may call the following functions:

- `send(x)`: Send a 1-bit value x to the other program. This call does not return until the other program calls `receive()`.
- `receive()`: Receive a 1-bit value from the other program and return it. This call does not return until the other program calls `send()`.

The total number of calls to `send()` made by Alice and Bob combined must not exceed 76. The same holds for `receive()`.
After communication is finished, each program must call the following function **exactly once**:

- `answer(x)`: Declare that the value of C is x. Immediately after calling this function, the program must terminate.

The game is successful if both Alice and Bob answer C correctly.

Write two programs, Alice and Bob, that always succeed in the game.

Interaction Protocol

This is an **interactive communication problem** (your programs and the judge communicate via standard input/output). Your program acting as Alice and your program acting as Bob will play the game through the judge.

Program Start

Your program first receives input in the following format:

Player
X

Here, *Player* is the string `Alice` or `Bob`, and X is an integer with $0 \le X < 2^{64}$.

- If *Player* is `Alice`, then $A = X$. From this point on, **your program must behave as Alice**.
- If *Player* is `Bob`, then $B = X$. From this point on, **your program must behave as Bob**.

After that, call `send`, `receive`, and finally `answer` as described below.

send

To call `send(x)`, output:

```
send x
```

Here, x must be 0 or 1.

This call does not return until the other program calls `receive()`. You will be judged as **Wrong Answer** if any of the following happens:

- the other program terminates while you are waiting,
- the other program also calls `send()` while you are waiting,
- the total number of calls to `send()` made by Alice and Bob exceeds 76.

receive

To call `receive()`, output:

```
receive
```

Then read the received bit x from standard input:

```
x
```

This call does not return until the other program calls `send()`. You will be judged as **Wrong Answer** if any of the following happens:

- the other program terminates while you are waiting,
- the other program also calls `receive()` while you are waiting,
- the total number of calls to `receive()` made by Alice and Bob exceeds 76.

If you are judged as Wrong Answer during the interaction, the input provided will be `-1`. **If you read `-1`, terminate your program immediately**, or you may receive Time Limit Exceeded.

answer

To call `answer(x)`, output:

```
answer x
```

Here, x must be equal to $\min(A, B)$. **After calling this function, terminate your program immediately.**

Note

- After each output, your program must flush standard output; Otherwise, you will receive Time Limit Exceeded.
- The judge program, your program acting as Alice, and your program acting as Bob run simultaneously. **The time and memory limits are measured as the sum of all of them**, so leave enough margin for both time and memory usage.
  - The judge program consumes up to about 50 ms and 5 MiB.
- You will handle integers in the range 0 to $2^{64} - 1$. Be careful about overflow.

Sample Interaction

Alice's Input	Alice's Output	Bob's Input	Bob's Output
Alice 31		Bob 25	
	send 0		receive
		0	
	receive		send 1
1			
	send 1		receive
		1	
	answer 25		answer 25

Problem L. Colorful Quadrilateral

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	1024 megabytes

You are given N distinct points on the 2D plane, numbered 1 through N. The coordinates of point i are (x_i, y_i).

Each point has a color. There are N colors in total, numbered 1 through N, and point i has color c_i.

Your task is to select four points with pairwise distinct colors, connect them in any order, and form a quadrilateral. The quadrilateral may have an interior angle of exactly 180°, but no pair of non-adjacent edges may intersect.

Determine whether such a quadrilateral can be formed. If it can, let S be the maximum possible area over all valid quadrilaterals, and output $2S$ as an integer. If it cannot be formed, output 0.

There are T independent test cases.

Input

The input is given in the following format:

T
case_1
case_2
$\vdots$
case_T

Here, each case_i is given in the following form:

N
$x_1\ y_1\ c_1$
$x_2\ y_2\ c_2$
$\vdots$
$x_N\ y_N\ c_N$

- All input values are integers.
- $1 \le T \le 10^4$
- $4 \le N \le 10^5$
- $|x_i|, |y_i| \le 10^8$
- $1 \le c_i \le N$
- If $i \ne j$, then $(x_i, y_i) \ne (x_j, y_j)$
- The total sum of N over all test cases does not exceed 10^5

Output

For each test case, output the answer on its own line.

Example

standard input	standard output
4 6 2 4 1 5 4 2 6 2 3 5 1 1 2 1 2 1 3 4 4 1 1 1 3 5 2 7 2 3 4 3 4 5 0 0 1 0 1 2 1 0 2 0 2 3 2 0 3 4 0 0 1 0 100000000 2 100000000 0 3 100000000 100000000 4	15 17 0 20000000000000000

Note

In the first example, connecting points $1, 2, 3, 6$ in this order forms a valid quadrilateral with area $15/2$, which is the maximum among all valid choices. The quadrilateral formed by points $1, 2, 4, 5$ has area 9, but points 1 and 4 share the same color, so it is invalid.

In the second example, a valid quadrilateral exists, but no convex quadrilateral satisfies the requirements.

In the third example, no valid quadrilateral can be formed.

As shown in the fourth example, the answer can be very large.

Problem M. Many Approaches

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes

There is a park with N squares arranged in a row from left to right, numbered $0, 1, \ldots, N-1$ in this order.

Inside the park, there are N people, also numbered $0, 1, \ldots, N-1$. When you announce a sequence $X = (X_1, X_2, \ldots, X_{|X|})$ of non-negative integers between 0 and $N-1$, the people perform a **march** according to the following rules:

1. For each $i = 0, 1, \ldots, N-1$, person i moves to square i.
2. For each $j = 1, 2, \ldots, |X|$, in this order, do the following:
    - Every person not currently on square X_j moves exactly one square toward X_j.

You are given a sequence $A = (A_0, A_1, \ldots, A_{M-1})$ of length M, consisting of integers between 0 and $N-1$.

You must answer Q online queries. For each $i = 1, 2, \ldots, Q$, integers t'_i, L'_i, R'_i, P'_i are given. First, reconstruct t_i, L_i, R_i, P_i using the following procedure:

Let $\text{ans}_0 = 0$, and let ans_i denote the answer to the i-th query. Reconstruct t_i, L_i, R_i, P_i as follows:

- $t_i = ((t'_i + \text{ans}_{i-1}) \bmod 2)$
- $a = ((L'_i + \text{ans}_{i-1}) \bmod M)$
- $b = ((R'_i + \text{ans}_{i-1}) \bmod M)$
- $L_i = \min(a, b)$
- $R_i = \max(a, b)$
- $P_i = ((P'_i + \text{ans}_{i-1}) \bmod N)$

Here, for a non-negative integer a and a positive integer b, $(a \bmod b)$ denotes the remainder when a is divided by b, which is in the range 0 through $b-1$.

For each reconstructed (t_i, L_i, R_i, P_i), answer the following query:

- If $t_i = 0$: Let $X = (A_{L_i}, A_{L_i+1}, \ldots, A_{R_i})$. Simulate the march and output the final square where person P_i ends up.
- If $t_i = 1$: Let $X = (A_{L_i}, A_{L_i+1}, \ldots, A_{R_i})$. Simulate the march and output how many people end up on square P_i.

Input

The input is given in the following format:

$N\ M\ Q$
$A_0\ A_1\ \ldots\ A_{M-1}$
$t'_1\ L'_1\ R'_1\ P'_1$
$t'_2\ L'_2\ R'_2\ P'_2$
$\vdots$
$t'_Q\ L'_Q\ R'_Q\ P'_Q$

- All input values are integers.
- $1 \le N, M, Q \le 2 \times 10^5$
- $0 \le A_i \le N-1$ $(0 \le i \le M-1)$
- $0 \le t'_i, t_i \le 1$ $(1 \le i \le Q)$
- $0 \le L'_i, R'_i \le M-1$ $(1 \le i \le Q)$
- $0 \le L_i \le R_i \le M-1$ $(1 \le i \le Q)$
- $0 \le P'_i, P_i \le N-1$ $(1 \le i \le Q)$

Output

Output Q lines. For each $i = 1, 2, \ldots, Q$, output ans_i, the answer to the i-th query.

Examples

standard input	standard output
4 5 3 0 2 3 2 1 0 1 3 2 1 0 2 1 1 4 4 1	2 0 3
7 4 1 3 3 3 3 1 3 0 3	7

Note

In the first example, for $(t_i, L_i, R_i, P_i) = (0, 1, 3, 2)$ with $X = (2, 3, 2)$, person 2 moves as $2 \to 2 \to 3 \to 2$, so the answer is 2.

In the second example, for $(t_i, L_i, R_i, P_i) = (1, 2, 4, 3)$ with $X = (3, 2, 1)$, the number of people on square 3 at the end is 0.

In the third example, for $(t_i, L_i, R_i, P_i) = (1, 4, 4, 1)$ with $X = (1)$, the number of people on square 1 at the end is 3.