Problem A. Array

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	512 mebibytes

bobo has an array $a[1], a[2], \dots, a[n]$.

He subsequently presents q questions of the following 4 types:

- *1* l_i r_i c_i - Update $a[k]$ with $a[k] + c_i$ for all $l_i \le k \le r_i$;
- *2* l_i r_i c_i - Update $a[k]$ with $\min\{a[k], c_i\}$ for all $l_i \le k \le r_i$;
- *3* l_i r_i c_i - Update $a[k]$ with $\max\{a[k], c_i\}$ for all $l_i \le k \le r_i$;
- *4* l_i r_i - Ask for $\min\{a[k] : l_i \le k \le r_i\}$ and $\max\{a[k] : l_i \le k \le r_i\}$.

Input

The first line contains 2 integers n, q $(1 \le n, q \le 200000)$.

The second line contains n integers $a_1, a_2, \dots, a_n$ which denotes the initial values of the array $(|a_i| \le 10^9)$.

Each of the following q lines contains an integer t_i which denotes the type of i-th question. If $t_i = 1, 2, 3$, 3 integers l_i, r_i, c_i follows. If $t_i = 4$, 2 integers l_i, r_i follows. $(1 \le t_i \le 4, 1 \le l_i \le r_i \le n)$

- If $t_i = 1$, $|c_i| \le 2000$;
- If $t_i = 2, 3$, $|c_i| \le 10^9$.

Output

For each question of type 4, two integers denote the minimum and the maximum.

Examples

standard input	standard output

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

Problem B. Chromatic Number

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

bobo has a **connected** graph G, and he wants to color each vertices with one of the c colors so that no two adjacent vertices share the same color.

Find the number of ways to color modulo $(10^9 + 7)$.

Input

The first line contains 3 integers n, m, c, which denote the number of vertices, edges, and colors, respectively $(1 \le n \le 10^5, n - 1 \le m \le n + 8, 1 \le c \le 10^9)$.

The vertices are conveniently numbered by $1, 2, \ldots, n$.

Each of the following m lines contains 2 integers a_i, b_i, which denotes an edge between vertices a_i and b_i $(1 \le a_i, b_i \le n, a_i \ne b_i)$.

Output

A single integer denotes the number of ways.

Examples

standard input	standard output

3 3 3
1 2
2 3
3 1
	6

4 3 1000000000
1 2
2 3
3 4
	3584

Problem C. Nearest friend

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

In the country there are n houses connected by m bidirected roads. Distance between two houses is the length of shortest path between them.

There are k bobo living in the houses. For each bobo, find another bobo living nearest to him.

Input

The first line contains 3 integers n, m, k ($2 \le n \le 200000, n-1 \le m \le 200000, 2 \le k \le n$).

The houses are conveniently labeled by $1, 2, \dots, n$.

Each of the following m lines contains 3 integers a_i, b_i, c_i, which denotes a road between houses a_i and b_i with length c_i ($1 \le a_i, b_i \le n, 1 \le c_i \le 10000$).

The last line contains k integers $v_1, v_2, \dots, v_k$, where v_i denotes the house the i-th bobo lives in ($1 \le v_i \le n$).

It is guaranteed that every two houses can reach each other, and no two bobo live in the same house.

Output

For each bobo, a single integer denotes the house where the nearest bobo lives. If there are multiple such bobo, find the house with the smallest label.

Examples

standard input	standard output

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

3 3 3
1 2 1
2 3 1
3 1 1
3 2 1
	1
	1
	2

Problem D. Sequence Sorting

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	512 mebibytes

Given n sequences $s_1, s_2, \ldots, s_n$, sort them in lexicographic order.

Input

The first line contains an integer n ($1 \le n \le 1000000$).

Each of the following n lines contains an integer k_i which denotes the length of s_i, followed by k_i integers $s_{i,1}, s_{i,2}, \ldots, s_{i,k_i}$ which denotes the sequence s_i ($1 \le s_{i,1} \le s_{i,2} \le \cdots \le s_{i,k_i} \le 5000000$).

($k_i \ge 1, k_1 + k_2 + \cdots + k_n \le 5000000$)

Since the input is extremely large, it is recommended to use the following code snippet to read a 32-bit integer.

```
#include <cctype>

int get_int() {
   char ch = getchar();
   while (!isdigit(ch)) ch = getchar();
   int ret = 0;
   while (isdigit(ch)) {
      ret = ret * 10 + ch - '0';
      ch = getchar();
   }
   return ret;
}
```

Output

n integers $p_1, p_2, \ldots, p_n$, where p_i denotes the index of the i-th smallest sequence.

If two or more sequences are the same, sort them according to their indices.

Examples

standard input	standard output

4
2 1 3
1 12
3 1 3 4
2 1 3
	1
	4
	3
	2

Problem E. New Point

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

There are n distinct lattice points (whose coordinates are both integers) in a two-dimension plane, forming geometric patternings.

bobo is going to add a new lattice point so that the total number of right triangles whose legs are parallel to the coordinate axis is maximized.

Note that the new point should be chosen carefully to avoid coincidence.

Input

The first line contains an integer n ($1 \le n \le 200000$).

Each of the following n lines contains 2 integers x_i, y_i which denotes the point (x_i, y_i) ($|x_i|, |y_i| \le 10^9$).

Output

A single integer denotes the maximum number of triangles after adding the point.

Examples

standard input	standard output

4
0 1
1 0
0 -1
-1 0
	4

5
0 0
0 1
1 0
0 -1
-1 0
	9

Problem F. Planar Graph Connectivity

Input file:	*standard input*
Output file:	*standard output*
Time limit:	4 seconds
Memory limit:	512 mebibytes

bobo has a connected planar graph with n vertices.

He subsequently presents q questions of the following 2 types:

- *-* a_i b_i - Remove the edge between vertices a_i and b_i, and ask for the number of connected components.
- *?* a_i b_i - Ask if vertices a_i and b_i are connected.

Answer his questions.

Input

The first line contains 2 integers n, q $(1 \le n \le 100000, 1 \le q \le 200000)$.

Vertices are numbered by $1, 2, \ldots, n$ for convenience.

Each of the following n lines starts with an integer k_i which denotes the number of neighbors of vertex i, followed by k_i integers $v_{i,1}, v_{i,2}, \ldots, v_{i,k_i}$ which denote the neighbors, ordered in clockwise direction $(0 \le k_i \le n-1, 1 \le v_{i,j} \le n)$.

The following q lines denote the questions.

Note that the numbers (in the questions) are encoded. If the answer of the last question is last, then number x appears as $x \oplus \text{last}$. (Assume last $= 0$ at the beginning. "$\oplus$" denotes bitwise exclusive-or.)

Output

For the first type of questions, a single integer denotes the number of components.

For the second type of questions, "1" for connected and "0" for disconnected.

Examples

standard input	standard output
4 3 3 2 3 4 2 1 4 1 1 2 1 2 - 1 2 - 0 2 ? 3 1	1 2 0

Problem G. Popo Sort

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 megabytes

Popo, one of bobo's silly friends, is learning bogo sort. He immediately invents his own version, as shown in the following code.

```
function sort(s)
  success := False
  for i := 1, 2, ..., m do
    if s[a[i]] > s[b[i]] then
      swap s[a[i]] and s[b[i]]
      success := True
    end
  end
  if success then
    return sort(s)
  else
    return s
  end
end
```

Determine whether Popo's version is correct (can sort all possible sequences of length n into non-descending order).

Input

The first line contains 2 integers n, m ($1 \leq n \leq 1000000, 0 \leq m \leq 1000000$).

Each of the following m lines contains 2 integers a_i, b_i ($1 \leq a_i, b_i \leq n$).

Output

Print "Yes" if the version is correct, or "No" otherwise.

Examples

standard input	standard output

3 2
1 2
2 3
	Yes

2 2
1 2
2 1
	No

Problem H. Power of Three

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 megabytes

bobo has a binary matrix of size $n \times m$. Today bobo is going to remove some (maybe none) of rows to maximize the *boboness* of the matrix.

The *boboness* is defined as follows. If there are odd number of ones in the i-th column, $a_i \cdot 3^{b_i}$ points are added up to the *boboness*, initially 0.

Find the maximum of *boboness*.

Input

The first line contains 2 integers n, m $(1 \le n \le 200000, 1 \le m \le 70)$.

Each of the following n lines contains m integers which denotes the matrix.

Each of the last m lines contains 2 integers a_i, b_i $(a_i \in \{-1, 1\}, 1 \le b_i \le 35)$.

It is guaranteed that for all $i \ne j$, either $a_i \ne a_j$ or $b_i \ne b_j$.

Output

A single integer denotes the maximum of *boboness*.

Examples

standard input	standard output

2 4
1101
0010
-1 1
-1 2
1 1
1 2
	3

1 1
1
-1 1
	0

Problem I. Remove Obstacles

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

bobo is in a maze with 2 rows and n columns. The rows are numbered from top to bottom, while the columns are numbered from left to right. Some cells may contain obstacles.

bobo starts at cell $(1, 1)$ and is going to cell $(2, n)$. He can step upward, downward or rightward. However, he cannot step into cells with obstacles or go out of the maze. Meanwhile, he won't visit a cell more than once.

As a magician, bobo can remove at most k obstacles. He wonder the maximum number of cells he can visit.

Input

The first line contains 2 integers n, k $(1 \leq n \leq 1000000, 0 \leq k \leq 2000000)$.

The second and third line each contains n characters which denotes the maze, where "`#`" denotes obstacle cell and "`.`" denotes empty cell.

It is guaranteed that bobo can go from $(1, 1)$ to $(2, n)$ without removing any obstacles.

Output

A single integer denotes the maximum number of cells.

Examples

standard input	standard output
2 1 .# ..	3
3 1 .#. ...	6

Problem J. Salesmen

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

bobo lives in a country where personal rockets are big fashion. The country consists of n cities which are conveniently numbered by $1, 2, \ldots, n$.

Cities are connected by bidirectional roads, and there is exactly one path between any two cities.

There are m salesmen in bobo's country. The i-th salesman travels along the roads between cities a_i and b_i and sells c_i rockets.

Since the rockets are not very high-quality, people in the i-th city will buy at most w_i rockets.

Now bobo wants to know how many rockets can be sold in salesmen's best effort (i.e. the maximum number).

Input

The first line contains 2 integers n, m $(1 \le n, m \le 10000)$.

The second line contains n integers $w_1, w_2, \ldots, w_n$ $(0 \le w_i \le 100000)$.

Each of the following $(n-1)$ lines contains 2 integers u_i, v_i which denotes a road between cities u_i and v_i $(1 \le u_i, v_i \le n)$.

Each of the last m lines contains 3 integers a_i, b_i, c_i $(1 \le a_i, b_i \le n, 0 \le c_i \le 100000)$.

Output

A single integer denotes the maximum number of rockets can be sold.

Examples

standard input	standard output

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