Problem A. Ancestor

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

NIO is playing a game about trees.

The game has two trees A, B each with N vertices. The vertices in each tree are numbered from 1 to N and the i-th vertex has the weight v_i. The root of each tree is vertex 1. Given K key numbers $x_1, \ldots, x_k$, find the number of solutions that remove exactly one number so that the weight of the lowest common ancestors of the vertices in A with the remaining number is greater than the weight of the lowest common ancestors of the vertices in B with the remaining number.

Input

The first line has two positive integers $N, K (2 \le K \le N \le 10^5)$.

The second line has K unique positive integers $x_1, \ldots, x_K (x_i \le N)$.

The third line has N positive integers $a_i (a_i \le 10^9)$ represents the weight of vertices in A.

The fourth line has $N - 1$ positive integers $\{pa_i\}$, indicating that the number of the father of vertices $i + 1$ in tree A is pa_i.

The fifth line has n positive integers $b_i (b_i \le 10^9)$ represents the weight of vertices in B.

The sixth line has $N - 1$ positive integers $\{pb_i\}$, indicating that the number of the father of vertices $i + 1$ in tree B is pb_i.

Output

One integer indicating the answer.

Examples

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

Note

The lowest common ancestor (LCA) (also called least common ancestor) of two nodes v and w in a tree or directed acyclic graph (DAG) T is the lowest (i.e. deepest) node that has both v and w as descendants, where we define each node to be a descendant of itself (so if v has a direct connection from w, w is the lowest common ancestor).(From Wiki.)

Problem B. Boss

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

NIO is a big boss. He will now dispatch his N employees to K cities. And he must send exactly e_i employees to the i-th city. NIO has no redundant employees which means,

$$N = \sum_{i=1}^{K} e_i$$

The cost of sending the i-th employee to the j-th city is c_{ij}. NIO wants to know the minimum cost of the dispatch.

Input

The first line contains two integers $N(1 \le N \le 10^5)$ and $K(1 \le K \le 10)$ — the number of employees and the number of the cities.

The second line contains K integers $e_i(1 \le e_i \le n)$ — the number of employees for i-th city.

The following N lines each contain K integers $c_{ij}(0 \le c_{i_j} \le 10^5)$ — the cost of sending the i-th employee to the j-th city.

Output

One integer represent the minimum cost.

Examples

standard input	standard output
5 2 1 4 1 1 4 3 4 5 2 1 3 2	11
10 8 1 1 1 2 2 1 1 1 6 10 5 6 5 4 9 2 7 8 8 6 1 4 9 5 7 4 6 2 5 6 9 1 8 4 5 3 10 5 10 1 8 1 7 3 5 9 10 1 6 10 6 4 10 6 8 7 10 8 9 4 8 9 5 1 10 4 5 3 8 7 10 1 7 3 2 8 1 2 2 10 6 10 2 5 9 1 6 1	26

Problem C. Concatenation

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	1024 megabytes

NIO was the king of the OIN Kingdom.

He had N children and wanted to teach them how to count. In the OIN Kingdom, pental is used in counting, so his children can only use 0, 1, 2, 3, and 4 to represent a number.

One day, NIO asked his children to write down their favorite numbers. After that, he came out with a problem: if he concatenated these numbers into a big number, what is the smallest number he could get? However, this problem was too hard for him to solve, so can you help him?

Input

The first line contains an integer $N(1 \le N \le 2 * 10^6)$, denoting the number of NIO's children.

Then follows N lines, each line contains a string s_i denotes the favorite number of the ith child. The string is composed of 0, 1, 2, 3, and 4, but may have leading zeros because NIO's children hadn't fully understood how to count.

$\sum_{i=1}^{N} |s_i| \le 2 * 10^7$

Output

One integer denotes the smallest number NIO can get.

Example

standard input	standard output
5 121 1 12 00 101	00101112112

Note

If you have designed an algorithm whose time complexity is $O(|S| \log |S|)$ or so, please think twice before submitting it. Any algorithm other than linear complexity is NOT supposed to pass this problem. But of course, you can have a try. If you do so, we wish you good luck.

Problem D. Directed

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

NIO is playing a D&D-like game.

In this game, the map can be concerned as a connected graph with N vertices and $N-1$ edges. The vertices are numbered from 1 to n and vertex 1 is the destination of his journey. Initially, all edges can be passed in both directions. At the beginning of the game, K edges will be randomly selected to become directed edges and he can only approach the destination along the directed edge. In this way, no matter which edge is selected, he can still reach the destination.

But unfortunately, NIO is a road nut. He started at vertex s. At each step, he will randomly select an available edge connected with the current vertex and move to the adjacent vertex.

NIO wanted to know how many steps he expected to take to reach the destination.

Since the answer is too large, NIO only wants to know it modulo 998244353.

Input

The first line contains three integers N, K, and s $(1 \le N \le 10^6, 0 \le K \le N-1, 1 \le s \le N)$.

Each of the following $N-1$ lines contains two integers u_i and v_i $(1 \le u_i, v_i \le N, u_i \ne v_i)$, describing an undirected edge (u_i, v_i) in the graph

It is guaranteed that the graph is connected and there is at most one edge between any pair of vertices.

Output

A single non-negative integer, denoting the expectation, modulo 998244353.

Examples

standard input	standard output
3 1 3 1 2 2 3	3
5 2 4 1 2 2 3 3 4 2 5	332748124

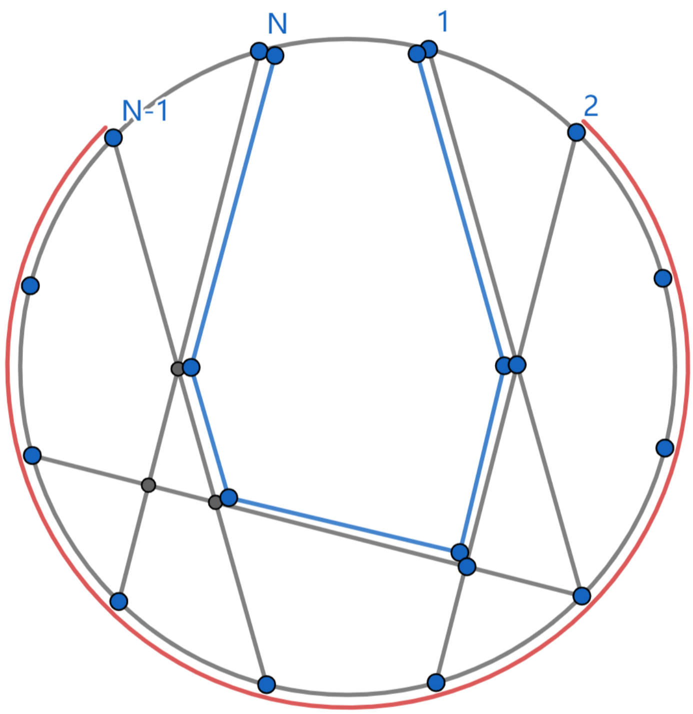
Problem E. Electrician

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

NIO would like to be an electrician.

He is now confronted with a difficult task. The device he is working on has a circular frame and there are N **equidistant** vertices on the frame. It can be simplified to a circle on a two-dimensional plane. In addition to the outermost circular frame, there are also M holders inside which can be simplified to segments connecting the vertices on the circle. Because these holders are produced together, they are all the same "length"L. "Length"L means that if a segment start at i-th vertex, it would end at $(i + L)$-th vertex and i must satisfy $(1 \le i \le N - L)$.

The task is to place two wires on the frame, one from vertex 1 to N and another from 2 to $N - 1$. The wires must be close to the segments or arcs on the frame and on where two segments intersect, a wire can also be connected from one segment to another.



For safety, two wires cannot intersect at any position, that is, they can not pass through the same vertex or intersections of segments, including vertices 1, 2, $N - 1$ and N. And on any segment and arc, the wire must follow the direction from the vertex with the small number to the vertex with the large number.

NIO wants to know how many schemes he has for placing wires.

Since the answer is too large, NIO only wants to know it modulo 998244353.

Input

The first line contains two integers N, M, L $(4 \le N \le 10^6, 0 \le M \le 10^4, 1 \le L < \frac{N}{2})$.

The following line contains M distinct integers $\{s_i\}$ $(1 \le s_i \le N - L)$, denoting that there is a segment connecting s_i and $(s_i + L)$.

Output

A single non-negative integer, denoting the answer, modulo 998244353.

Examples

standard input	standard output
6 3 1 1 2 3	4
12 5 4 1 2 5 7 8	42

Problem F. Fief

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

Duke NIO had 2 sons, Antileaf and Fstqwq, and they would inherit Duke NIO's fief. Duke NIO's fief was composed of n cities, and there were m bi-direction roads connecting these cities. For fairness, Duke NIO made the following rules for Antileaf and Fstqwq when inheriting his fief:

Firstly, both Antileaf and Fstqwq could choose a favorite city.

Then, according to the two favorite cities, Duke NIO would make a list. The content of the list is his n cities, and the first item of the list is Antileaf's favorite city while the last item of the list is Fstqwq's favorite city.

After that, Antileaf and Fstqwq could begin to inherit his fief. The number of cities each one would get should be determined by drawing a lot. For instance, Antileaf would get the first a cities in NIO's list(a is uniformly random between 1 and $n-1$) while Fstqwq would get the last $n-a$ cities.

After all the cities were assigned to his two sons, each son's fief should be a connected component.

However, he found that in some cases, he couldn't find a list that meet his requirement. He came out with some questions - if Antileaf chose city x as his favorite city and Fstqwq chose city y, could he always find a list such that Antileaf and Fstqwq's fief would be always connected regardless of the result of the lottery?

Input

The first line contains two integers n and m ($2 \le n \le 100000$, $0 \le m \le 200000$), denoting the number of cities and the number of roads. Then follows m lines, each line contains two integers u_i and v_i ($1 \le u_i,\ v_i \le n$, $u_i \ne v_i$), denotes the roads.

After that follows a line with a single integer q ($1 \le q \le 100000$), which denotes the number of NIO's queries. And then follows q lines, each line contains two integers x_i and y_i ($1 \le x_i,\ y_i \le n$, $x_i \ne y_i$), the meaning is described above.

Output

Output q lines. For the i-th line, if the answer of the i-th query is true, print "YES"(excluding quotes), otherwise print "NO"(excluding quotes).

Examples

standard input	standard output
4 3 1 2 2 3 4 3 3 1 3 4 1 3 4	NO YES NO
4 6 1 2 1 3 1 4 2 3 2 4 3 4 2 1 4 2 4	YES YES

Note

For the second query in the first example, the only valid list is $(4, 3, 2, 1)$.

For the first query in the second example, one valid list is $(1, 3, 2, 4)$.

For the second query in the second example, one valid list is $(2, 3, 1, 4)$.

Problem G. Geometry

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

NIO loves geometry.

One day he came out with a problem: given two convex polygons in a 2-dimension plane, each with a velocity, what is the moment when they collide? In this problem, it is considered to be a collision that two convex polygons share one point.

Input

The first line contains an integer n ($3 \le n \le 100000$), denoting the number of vertices of the first convex polygon. Then follows n lines, each line contains two integers x_i, y_i ($-10^9 \le x_i,\ y_i \le 10^9$), denotes the ith point of the convex polygon. The points are given in counter-clockwise order.

Then follows an integer m ($3 \le m \le 100000$), and then m lines, denoting the second polygon. The format and restrictions are the same as above.

The last line contains four integers $vx_1,\ vy_1,\ vx_2,\ vy_2$ ($-10^9 \le vx_1,\ vy_1,\ vx_2,\ vy_2 \le 10^9$). The velocity of the first polygon is (vx_1, vy_1) and the velocity of the second polygon is (vx_2, vy_2).

Output

Output a single real number t, indicating how many units of time they collided after the initial state. If the two convex polygons collide at the initial state, print 0. If they will never collide, print -1.

Your answer will be judged to be correct if the relative or absolute error with the jury's answer is less than or equal to 10^{-6}.

Example

standard input	standard output
3 0 0 1 1 1 0 3 0 0 -1 -1 0 -1 1 1 -1 -1	0

Problem H. Hacker

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

NIO is a hacker who likes to crack passwords for fun.

But one day he forgot his password. He quickly listed some vague clues. There is a lowercase string A of length n and k lowercase strings $B_1, \ldots, B_k$ of length m. Given a uniform weight $v_1, \ldots, v_m$ represents the weight of each position of any B_i.

To figure out his password, for all B_i, he wants to find its maximum interval weight sum, and the string formed by this interval is a substring of A(An empty interval represents an empty string, which is legal).

Input

The first line contains three positive integers $n, m, k(n, m, k \leq 10^5, m * k \leq 10^6)$.

The second line contains a string A of length n.

The third line contains m integers $v_i(-10^9 \leq v_i \leq 10^9)$.

The next k lines represent the string $\{B_i\}$.

Output

k lines for k answers.

Examples

standard input	standard output
5 5 5 uqusa -9 -7 8 -8 -3 saimh qusam qusaf ubgxj uquxw	0 8 8 0 8
10 8 5 aepeihmkkz 0 -3 2 -4 -5 2 -5 5 epesxnud epeihmkk peihmrgo epeihqvy eppjkuaz	2 5 2 2 5

Problem I. Ice Drinking

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	256 megabytes

One day NIO was very thirsty, so he went to the ice-drinking room and ordered n different drinks, each with a unique ingredient.

The owner thought NIO's request was too weird, so he asked NIO, "How can you eat like this?"However, NIO just responded with "none of your business". The ice-drinking room owner was very angry, so he decided to add the n ingredients to NIO's n drink at random while ensuring that each cup of drink will have exactly one ingredient.

NIO discovered what the ice-drinking owner was trying to do. Given a natural number k, he wants to know in advance what the expected value of x^k is, if we let x denote the number of cups with the correct ingredient.

Since Nio does not like fractions, he only wants to know the value of the result modulo 862118861.

Input

The only line of input contains two natural numbers n, k in order, where $1 \le n \le 10^{18}$, $0 \le k \le n+5\times10^3$.

Output

Output a natural number in one line, denoting the answer modulo 862118861.

Examples

standard input	standard output
3 4	14
4 2	2

Problem J. Jouney

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	1024 megabytes

NIO and Desprado2 are good friends and they lives in the same city. However, everytime NIO drive to visit desprado2, it takes him a long time to wait for red lights, and NIO is very distressed about this. There are n crossroads in their city. NIO is a very unlucky guy, everytime he go straight, turn left or turn around at a crossroad, he will encounter a red light and have to wait for it. Turning right at a crossroad don't need to wait for the red light.

NIO hates red lights, so he wants to know the minimal number of red lights he would encounter. Can you help him?

Input

The first line contains an integer n, $2 \le n \le 500000$, which denotes the number of crossroads in the city.

Then follows n lines, each line contains four **distinct** integers $c_{i,1}$, $c_{i,2}$, $c_{i,3}$, $c_{i,4}$ ($0 \le c_{i,j} \le n$), denotes the starting points of the four roads to the ith road. If $c_{i,j} = 0$, then these roads are from some other city and NIO will never go this way. The roads are in counter-clockwise order, that is, if NIO is at road $<c_{i,j}, i>$ and wants to go to road $<i, c_{i,j\%4+1}>$, he does not need to wait for the red light because he is turning right at crossroad i, otherwise he will have to wait for the red light. It is guaranteed that the map doesn't contain multiple edges and self-loops.

The last line contains four integers s_1, s_2, t_1, t_2 ($1 \le s_1,\ s_2,\ t_1,\ t_2 \le n$). NIO is at the road $<s_1, s_2>$ and he wants to go to road $<t_1, t_2>$ and visit Desprado2. It is guaranteed that both roads are in the map given above.

Please note that road $<a, b>$ is not equivalent to road $<b, a>$. If NIO is at road $<a, b>$ and he wants to go to road $<b, a>$, he should turn around at crossroad b and wait for a red light.

Output

Output one integer, the minimal number of red lights NIO would encounter.

If there is no way to visit Desprado2, print -1 instead.

Example

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