Problem A. Angle Beats

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

You have a rectangular board consisting of $n \times m$ squares. Each square contains a character which is either "*", "+", or ".".

A tromino is a figure formed by a square of the board, called the center, and two other squares, each sharing an edge with the center. A tromino is L-shaped if these two squares have a common vertex, and I-shaped otherwise.

You can draw some disjoint trominoes on the board. The center of an I-shaped tromino must contain "+". The center of an L-shaped tromino must contain either "*" or "+". All non-center squares of all trominoes must contain ".".

Your goal is to draw the maximum number of non-intersecting polyominoes.

Input

The first line contains two integers n and m: the number of rows and columns of the board ($2 \le n, m \le 100$).

Each of the next n lines contains m characters, and each character is either "*", "+", or ".". Together, these lines describe the board.

Output

Print n lines, each containing m characters: the board with trominoes on it. If a square belongs to some tromino, output a lowercase English letter, and if not, output the character contained in this square. Squares of the same tromino must contain the same letter. Squares which share an edge and belong to different trominoes must contain different letters.

If there are several possible answers, print any one of them.

Examples

standard input	standard output
2 2 *. ..	aa a.
3 3 ... .*. ...	.a. aa. ...
5 5 +*..+ ..++. .+.++ .**.+ .+*.+	++baa .bb+a ccc++ .**.+ .+*.+
11 13 ....+++....+++.. ....+..+..... ....+..+..... ....+..+..... .+++....+++.. ....+......+. ....+......+. ....+......+. .+++....+++..	.abc....abc.. aabcc..aabcc. ..baaacccb... ...dddeee.... .abfffgggbc.. aab+a..aabcc. ..baa...abaaa ...ccc....ddd .abddd..abccc aabee..aabee. ..be.....be..

Problem B. Best Subsequence

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	256 mebibytes

You have an array $w_1, w_2, \ldots, w_n$ of length n.

You need to choose a subsequence of k elements. Let their indices be $1 \le i_1 < i_2 < \ldots < i_k \le n$.

Your goal is to find the minimum possible value of

$$\max\left((w_{i_1} + w_{i_2}), (w_{i_2} + w_{i_3}), \ldots, (w_{i_{k-1}} + w_{i_k}), (w_{i_k} + w_{i_1})\right)$$

among all such subsequences.

Input

The first line of input contains two integers n and k: the number of elements in the array w and the length of subsequence ($3 \le k \le n \le 200\,000$).

The second line contains n space-separated integers $w_1, w_2, \ldots, w_n$ ($1 \le w_i \le 10^9$).

Output

Print one integer: the minimum possible value of

$$\max\left((w_{i_1} + w_{i_2}), (w_{i_2} + w_{i_3}), \ldots, (w_{i_{k-1}} + w_{i_k}), (w_{i_k} + w_{i_1})\right)$$

among all subsequences of length k.

Example

standard input	standard output
5 3 17 18 17 30 35	35

Problem C. Cool Pairs

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

You have two permutations of n elements, $p_1, p_2, \ldots, p_n$ and $q_1, q_2, \ldots, q_n$, and one integer k.

You need to find two integer arrays, a and b, with the following properties:

- The elements of the arrays must be integers such that $-n \le a_i, b_i \le n$.
- The permutations induce the following order: $a_{p_1} \le a_{p_2} \le \ldots \le a_{p_n}$ and $b_{q_1} \le b_{q_2} \le \ldots \le b_{q_n}$.
- A pair (i, j) is **cool** if $i < j$ and $a_i + b_j < 0$. The number of cool pairs must be exactly k.

Input

The first line of the input contains two integers n and k: the number of elements and the required number of cool pairs ($1 \le n \le 300\,000$, $0 \le k \le \frac{n \cdot (n-1)}{2}$).

The second line contains n space-separated integers: the permutation $p_1, p_2, \ldots, p_n$.

The third line contains n space-separated integers: the permutation $q_1, q_2, \ldots, q_n$.

It is guaranteed that each integer from 1 to n appears exactly once in each permutation.

Output

If there is no such pair of integer arrays that the number of cool pairs is equal to k, print "`No`" on a single line.

Otherwise, print "`Yes`" on the first line, and print the arrays a and b on the next two lines. Separate array elements by spaces.

Example

standard input	standard output
5 3 3 5 1 2 4 1 2 3 4 5	Yes 2 3 -1 5 1 -5 -3 -2 -2 0

Problem D. Dates

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	256 mebibytes

Ilya loves dating! He is planning his dates for the next t days (these days are numbered from 1 to t). For each day x, he knows that he can plan at most a_x dates.

Ilya has n known girls numbered from 1 to n. Girl i is willing to go on a date with him at most once, on any day in $[l_i, r_i]$, and Ilya will get p_i pleasure for dating this girl. For each $i < n$, it is true that $l_i \le l_{i+1}$ and $r_i \le r_{i+1}$.

Ilya's goal is to plan dates with some girls to maximize total pleasure. Help him! Find the maximum total pleasure he can get if he will properly choose the girls and the days of dates with them.

Input

The first line contains two integers n and t: the number of girls and the number of days ($1 \le n, t \le 300\,000$).

The next line contains t integers $a_1, a_2, \ldots, a_t$: here, a_i is the maximum number of dates Ilya can plan on day i ($0 \le a_i \le n$).

The next n lines contain the descriptions of the girls. The i-th of these lines contains three integers, l_i, r_i, and p_i: the borders of the i-th girl's days segment and the pleasure Ilya will get for dating her ($1 \le l_i \le r_i \le t$, $1 \le p_i \le 10^9$).

It is guaranteed that, for each $i < n$, it is true that $l_i \le l_{i+1}$ and $r_i \le r_{i+1}$.

Output

Print one integer: the maximum total pleasure Ilya can get if he will properly choose the girls and the days of dates with them.

Example

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

Problem E. Expected Value

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3.5 seconds
Memory limit:	256 mebibytes

You are given a connected plane graph on n vertices. At first, you are standing at vertex 1. Then, each second you are going to some vertex adjacent to the current one: the destination vertex is chosen uniformly at random among all adjacent vertices. Your task is to calculate the expected value of the first time you arrive at vertex n.

Input

The first line of input contains one integer n: the number of vertices in the given plane graph ($2 \le n \le 3000$).

The next n lines contain the description of the points where vertices are located. The i-th of them contains two integers x_i and y_i ($0 \le x_i, y_i \le 5000$). It is guaranteed that all given points are distinct.

The next line of input contains one integer m: the number of edges in the given plane graph ($n-1 \le m \le \frac{n \cdot (n-1)}{2}$).

The next m lines describe edges of the graph. The i-th of these lines contains two integers a_i and b_i ($1 \le a_i, b_i \le n$, $a_i \ne b_i$). It means that there is an edge between vertices a_i and b_i which is the segment between the corresponding points. It is guaranteed that no two of the given segments are intersecting (they can intersect only at the common ends), there are no multiple edges, and the graph is connected.

Output

It is guaranteed that, in the given tests, the required expected value can be represented as an irreducible fraction $\frac{P}{Q}$ where $P, Q > 0$. You need to print the value $(P \cdot Q^{-1})$ modulo $998\,244\,353$.

Examples

standard input	standard output
2 0 0 35 35 1 1 2	1
6 0 0 1 1 2 4 3 9 4 16 5 25 8 1 2 2 3 2 4 3 4 4 5 5 6 1 6 2 6	798595486

Problem F. Free Edges

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

You have an undirected graph. Initially, all edges are white. You can choose some edges and paint them black.

After that, while there is a vertex such that exactly one white edge comes out of it, this white edge also becomes black.

Your goal is to choose the minimum possible number of edges to paint black such that, after the process is finished, all edges will be black.

Input

The first line of the input contains two integers n and m: the number of vertices and the number of edges in your graph ($1 \le n, m \le 10^5$).

The next m lines contain description of the edges of the graph. Each of these lines contains two integers a_i and b_i, describing an edge between vertices a_i and b_i ($1 \le a_i, b_i \le n$, $a_i \ne b_i$).

It is guaranteed that there are no multiple edges.

Output

Print one integer: the minimum possible number of edges you need to paint black such that, after the end of the described process, all edges will be black.

Example

standard input	standard output
5 3 3 5 5 1 1 3	1

Problem G. Graph Counting

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	256 mebibytes

Consider undirected graphs on $2n$ vertices with no loops and no multiple edges. We will say that a graph G is **good** if there is no perfect matching in G, but for any edge not in G, if we add it to G, the resulting graph will have a perfect matching.

Your goal is to calculate the number of different good graphs on $2n$ vertices modulo 998 244 353.

Two graphs are different if they are non-isomorphic, meaning that one graph can not be transformed into another by relabeling the vertices.

Input

The first line of the input contains one integer n ($1 \le n \le 500\,000$). Recall that $2n$ is the number of vertices in graph.

Output

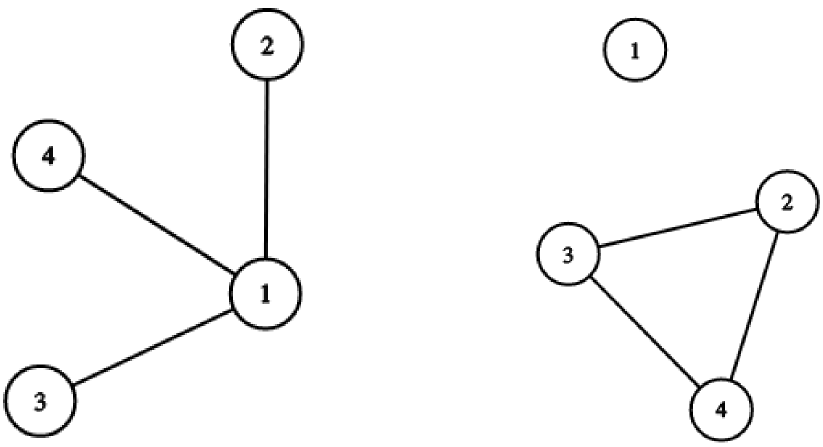
Print one integer: the number of different good graphs on $2n$ vertices modulo 998 244 353.

Examples

standard input	standard output
2	2
353535	331835697

Note

Graphs for $2n = 4$:



Problem H. Hall's Theorem

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

Consider a bipartite graph with vertices grouped into two parts, left and right, and edges only between vertices from different parts. Let A be a subset of vertices from the left part. We define $N(A)$ as the set of vertices from the right part which are adjacent to at least one vertex in A.

A subset A of vertices from the left part is **critical** if $|N(A)| < |A|$.

Your task is to find a bipartite graph which has n vertices in the left part, n vertices in right part, and exactly k critical subsets.

Input

The first line contains two integers n and k: the number of vertices in each part of the bipartite graph and the required number of critical subsets ($1 \le n \le 20$, $0 \le k < 2^n$).

Output

On the first line, print one integer m: the number of edges in your bipartite graph.

The next m lines must describe the edges of your graph. Each of them must contain two integers a_i and b_i, describing the edge from a_i to b_i ($1 \le a_i, b_i \le n$).

The graph must contain no multiple edges. Additionally, it must have exactly k critical subsets.

If there are several possible solutions, print any one of them. It is guaranteed that the solution always exists under the given input constraints.

Example

standard input	standard output
3 5	2 1 1 2 1

Problem I. Interesting Graph

Input file:	*standard input*
Output file:	*standard output*
Time limit:	7 seconds
Memory limit:	256 mebibytes

You have an undirected graph with the following property:

For any subset A of 7 vertices of the graph, there are some two vertices $a, b \in A$ and some vertex $c \notin A$ such that all paths from a to b contain vertex c.

You need to find the number of ways to properly color this graph in $1, 2, \ldots, n$ colors modulo 998 244 353.

A graph is colored in k colors by assigning an integer color from 1 to k to every vertex. A coloring is proper if the endpoints of each edge in the graph have different colors.

Input

The first line of the input contains two integers n and m: the number of vertices and the number of edges in your graph ($1 \le n, m \le 10^5$).

The next m lines contain description of the edges of the graph. Each of these lines contains two integers a_i and b_i describing an edge between vertices a_i and b_i ($1 \le a_i, b_i \le n$, $a_i \ne b_i$). There are no multiple edges.

It is guaranteed that for any subset A of 7 vertices of the graph, there are some two vertices $a, b \in A$ and some vertex $c \notin A$ such that all paths from a to b contain vertex c.

Output

Print one line containing n space-separated integers. The i-th integer must be the number of ways to properly color this graph in i colors, taken modulo 998 244 353.

Example

standard input	standard output
5 3 3 5 5 1 1 3	0 0 54 384 1500

Problem J. Jealous Split

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

You have an array of non-negative integers $a_1, a_2, \ldots, a_n$.

You need to split it into k non-empty subsegments: $[1; b_1], [b_1 + 1; b_2], \ldots, [b_{k-1} + 1; n]$.

Let us denote the sum on i-th segment as s_i and the maximum on i-th segment as m_i. Your goal is to make $|s_i - s_{i+1}| \le \max(m_i, m_{i+1})$ for each $1 \le i \le k - 1$.

Input

The first line of the input contains two integers n and k: the size of the array and the required number of segments ($3 \le k \le n \le 100\,000$).

The next line contains n integers $a_1, a_2, \ldots, a_n$: the given array ($0 \le a_i \le 50\,000$).

Output

If splitting is possible, print "`Yes`" on the first line, and then print $k - 1$ space-separated integers $b_1, b_2, \ldots, b_{k-1}$ on the second line. The integers must satisfy $1 \le b_1 < b_2 < \ldots < b_{k-1} < n$. Additionally, the inequalities $|s_i - s_{i+1}| \le \max(m_i, m_{i+1})$ must hold for each $1 \le i \le k - 1$. If there are several possible solutions, print any one of them.

If splitting is impossible, print "`No`" on a single line.

Example

standard input	standard output
5 3 17 18 17 30 35	Yes 2 4

Problem K. Knowledge

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

You have a string s consisting of lowercase English letters "a" and "b".

You can make zero or more operations in any order. Here are the possible operations:

- Delete "aa" from any place of the string.
- Delete "bbb" from any place of the string.
- Delete "ababab" from any place of the string.
- Add "aa" to any place of the string.
- Add "bbb" to any place of the string.
- Add "ababab" to any place of the string.

Your goal is to calculate the number of strings of length x that can be obtained by such operations. As the answer can be very large, find it modulo 998 244 353.

Input

The first line of the input contains one integer n: the length of the string ($1 \le n \le 300\,000$).

The second line contains a string s of length n consisting of lowercase English letters "a" and "b".

The third line contains one integer x ($0 \le x \le 10^9$), the length of the string you need to obtain.

Output

Print one integer: the number of strings of length x that can be obtained from string s by making the operations described above, taken modulo 998 244 353.

Examples

standard input	standard output
6 ababab 3	1
3 bbb 2	1
5 babab 35	866826000