Problem A. Splits

Input file:	`input.txt` or standard input
Output file:	`output.txt` or standard output
Time limit:	1 second
Memory limit:	512 megabytes

Given two integers n and k.

Split of n is a sequence of positive integer numbers $a_1, a_2, \ldots, a_m$, that meets the condition $a_1 + a_2 + \ldots + a_m = n$.

You should find any split of n that elements a_i have exactly k different values.

Input

The first line contains one integers n ($1 \le n \le 100\,000$), the number, which you should split.

The second line contains one integers k ($1 \le k \le n$), the required number of different values in a split.

Output

If a required split doesn't exist you should print -1.

Otherwise print number m ($1 \le m \le n$) of the elements in your split.

In the next line print the sequence of m positive integer numbers $a_1\ a_2\ \ldots\ a_m$ ($1 \le a_i \le n$), which sum equals to n and it has exactly k different values.

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

Examples

input	output
14 3	6 3 3 1 5 1 1
10 1	1 10
5 4	-1

Note

In the first example we split the number 14 into six values, which are equal 1, 3 or 5. Notice that required split isn't unique, for example, split $[1,\ 1,\ 2,\ 2,\ 4,\ 4]$ is also correct.

Let's have a look on all the nondecreasing splits of number 5:

- $[1,\ 1,\ 1,\ 1,\ 1], [5]$ have only one different value.
- $[1,\ 1,\ 1,\ 2], [1,\ 1,\ 3], [1,\ 2,\ 2], [1,\ 4], [2,\ 3]$ have two different values.

Thus there is no split of the number 5 that have four different values. So the answer for the third test is -1.

Scoring

There are 50 tests in this problem, **including tests from the examples**, each test costs 2 points.

Solutions that correctly works with $n \le 5$ will be scored at least 20 points.

Solutions that correctly works with $n \le 20$ will be scored at least 60 points.

Solutions that correctly works with $20 < n$, $k = 1$ will be scored at least 10 points.

Solutions that correctly works with $20 < n$, $k \leq 2$ will be scored at least 20 points.

Problem B. Non-triangles

Input file:	`input.txt` or standard input
Output file:	`output.txt` or standard output
Time limit:	2 seconds
Memory limit:	512 megabytes

Consider an undirected graph G, with a positive integer weight assigned to each vertex. A *non-triangle* in the graph G is a triple of **distinct** vertices u, v, w such that at least one of the edges (u, v), (v, w), or (u, w) is not present in the graph. The *cost* of a non-triangle is defined as the total weight of its vertices. The cost of the graph is defined as the largest cost among its non-triangles, or 0 if there are no non-triangles in the graph.

You are given an undirected graph with n vertices, and a sequence of q queries of two types:

- `1 u v`. Add the edge (u, v) $(1 \le u < v \le n)$ to the graph. It is guaranteed that the graph does not contain the edge (u, v) immediately prior to this query.
- `2 u v`. Remove the edge (u, v) $(1 \le u < v \le n)$ from the graph. It is guaranteed that the graph contains the edge (u, v) immediately prior to this query.

Print the cost of the graph after each query.

Input

The first line contains three integers n, m, q $(3 \le n \le 200\,000, 0 \le m \le 200\,000, 1 \le q \le 200\,000)$ — the number of vertices in the graph, the number of edges in the graph prior to all queries, and the number of queries, respectively.

The second line contains n integers c_i $(1 \le c_i \le 10^8)$. The i-th of these integers describes the weight of the vertex i.

The following m lines describe edges of the graph prior to all queries. The i-th of these lines contains two integers u_i and v_i $(1 \le u_i < v_i \le n)$, describing an edge between vertices u_i and v_i. It is guaranteed that each pair (u, v) is listed at most once.

The following q lines describe queries according to the format shown above.

Output

Print q integers. The i-th of these integers should be equal to the cost of the graph after processing first i queries.

Example

input	output
5 4 5 1 2 3 4 5 2 5 3 5 4 5 3 4 1 2 4 2 2 5 2 3 4 1 3 4 2 4 5	10 11 12 11 12

Note

Consider the sample case.

After the first query we can take vertices 2, 3, 5 (since there is no edge between 2 and 3), and the graph cost is $2 + 3 + 5 = 10$.

After the second query we can take vertices 2, 5, 4 (since there is no edge between 2 and 5), and the cost is $2 + 5 + 4 = 11$.

After the third query we can take vertices 3, 4, 5 (since there is no edge between 3 and 4), and the cost is $3 + 4 + 5 = 12$.

After the fourth query we can take vertices 2, 5, 4 (since there is no edge between 2 and 5), and the cost is $2 + 5 + 4 = 11$.

After the fifth query we can take vertices 4, 5, 3 (since there is no edge between 4 and 5), and the cost is $4 + 5 + 3 = 12$.

Scoring

Tests for this problem are divided into six groups. For each of the groups you earn points only if your solution passes all tests in this group and all tests in all of the **required** groups.

Group	Points	Additional constraints		Req. groups	Comment
		n	q		
0	0	–	–	–	Sample tests.
1	10	$n \le 10$	$q \le 100$	0	
2	10	$n \le 300$	$q \le 500$	$0 - 1$	
3	15	$n \le 2000$	$q \le 2000$	$0 - 2$	
4	20	$n \le 2000$	–	$0 - 3$	
5	20	–	–	–	No "add" queries.
6	25	–	–	$0 - 5$	

Problem C. Find the Path

Input file:	`input.txt` or standard input
Output file:	`output.txt` or standard output
Time limit:	3 seconds
Memory limit:	512 megabytes

You are given a tree (a connected undirected graph without cycles) with n vertices. For a pair of vertices u and v, let $f(u, v)$ denote the sequence of vertex indices along the unique path from the vetex u to the vertex v, ordered from u towards v. For example, for any edge (u, v) of the tree we have $f(u, v) = [u, v]$, and for any vertex u we have $f(u, u) = [u]$.

For the given tree, n^2 sequences $f(i, j)$ for all possible pairs $1 \le i, j \le n$ are considered and lexicographically ordered.

You are given q queries. Each query consists of a single integer k. For each query, find the pair of vertices u and v such that the path connecting them is at position k in the ordered list of sequences. Sequences in the list are numbered starting from 1.

Input

The first line contains two integers n and q ($1 \le n \le 100\,000, 1 \le q \le 300\,000$) — the number of vertices in the tree, and the number of queries, respectively.

The following $n-1$ lines contain pairs of integers u_i, v_i ($1 \le u_i, v_i \le n$), describing edges between vertices u_i and v_i. It is guaranteed that the given graph is a tree.

The following q lines describe queries. A query is described by a single integer k ($1 \le k \le n^2$) — the number of the path which endpoints have to be determined.

Output

For each query print the answer in a separate line. If $f(u, v)$ is at position k in the list of sequences, print u and v.

Example

input	output
3 4 1 2 2 3 1 5 2 9	1 1 2 1 1 2 3 1

Note

In the sample case, the sequences are as follows:

$$[1], [1, 2], [1, 2, 3],$$

$$[2], [2, 1], [2, 3],$$

$$[3], [3, 2], [3, 2, 1]$$

A sequence a of length n is lexicographically smaller than a sequence b of length m if and only if there exists an index $i \le \min(n, m)$ such that $a_j = b_j$ for all $j < i$ and $a_i < b_i$, or if $n < m$ and $a_i = b_i$ for all $i \le n$.

Note that k may exceed the limits of 32-bit integer type.

Scoring

Tests for this problem are divided into six groups. For each of the groups you earn points only if your solution passes all tests in this group and all tests in all of the **required** groups.

Let d_v denote degree of a vertex v — the number of vertices directly connected to v by an edge.

Group	Points	Additional constraints			Req. groups	Comment
		n	q	k		
0	0	–	–	–	–	Sample tests.
1	11	$n \le 100$	$q \le 10\,000$	–	0	
2	15	$n \le 1000$	$q \le 10\,000$	–	0 – 1	
3	12	–	–	–	–	$d_v = 1$ for exactly two vertices.
4	12	–	–	–	–	$d_v = 1$ for exactly $n - 1$ vertices.
5	25	–	–	$k \le n$	–	
6	25	–	–	–	0 – 5	

Problem D. Points on the Plane

Input file:	`input.txt` or standard input
Output file:	`output.txt` or standard output
Time limit:	3 seconds
Memory limit:	512 megabytes

You are given n points on the plane and q queries of the following kind:

- `1` x_1 y_1 x_2 y_2 ($x_1 \neq x_2$ or $y_1 \neq y_2$). Consider the line going through points (x_1, y_1) and (x_2, y_2). There are two half-planes having this line as a border. Consider one of them, which contains point $(x_1 + y_2 - y_1, y_1 + x_1 - x_2)$. In other words, if you consider directed vector from (x_1, y_1) to (x_2, y_2), the half-plane in question will be to the right. You need to check whether there exists at least one given point lying in this half-plane. Please note, that half-planes contain their border, so the points lying on the borderline are accounted as lying inside.
- `2` x_1 y_1 x_2 y_2 ($x_1 \neq x_2$ or $y_1 \neq y_2$). Consider the square, whose two opposite vertices are points (x_1, y_1) and (x_2, y_2). Please note, that sides of this square are not necessarily parallel to coordinate axes. You need to check whether there exists at least one given point lying inside or on the border of the square.

Input

The first line contains two integers n and q ($1 \le n, q \le 100\,000$), the number of points and the number of queries.

The next n lines contain pairs of integers x_i, y_i ($1 \le x_i, y_i \le 10^8$), describing the given points.

The next q lines give queries in the format given in the legend. ($1 \le x_{i,1}, y_{i,1}, x_{i,2}, y_{i,2} \le 10^8$)

Output

For each query print "`Yes`" if the described object contains at least one given point, and "`No`" otherwise.

Scoring

Tests for this problem are divided into eight groups. For each of the groups you earn points only if your solution passes all tests in this group and all tests in all of the **required** groups.

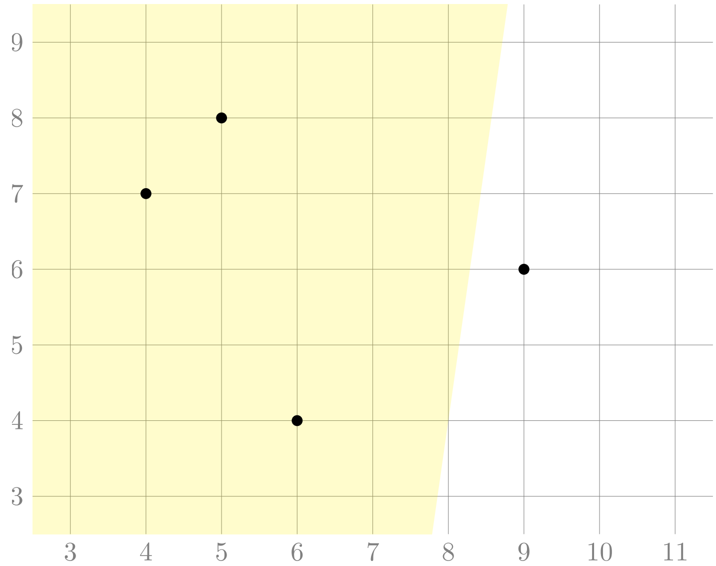
Group	Points	Additional constraints		Req. groups	Comment
		n	q		
0	0	–	–	–	Sample tests.
1	11	$n \le 1000$	$q \le 1000$	–	For all i holds $t_i = 1$.
2	12	–	–	1	For all i holds $t_i = 1$.
3	5	$n \le 1000$	$q \le 1000$	–	For all i holds $t_i = 2$ and $x_{i,1} - x_{i,2} = y_{i,1} - y_{i,2}$.
4	16	–	–	3	For all i holds $t_i = 2$ and $x_{i,1} - x_{i,2} = y_{i,1} - y_{i,2}$.
5	12	$n \le 1000$	$q \le 1000$	$0, 1, 3$	
6	17	$n \le 30\,000$	$q \le 30\,000$	$0, 1, 3, 5$	
7	9	$n \le 60\,000$	$q \le 60\,000$	$0, 1, 3, 5, 6$	
8	18	–	–	$0 - 7$	

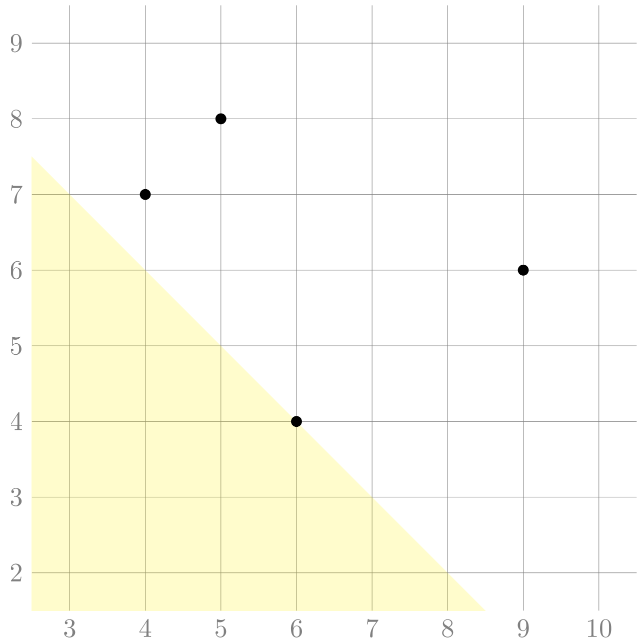
Examples

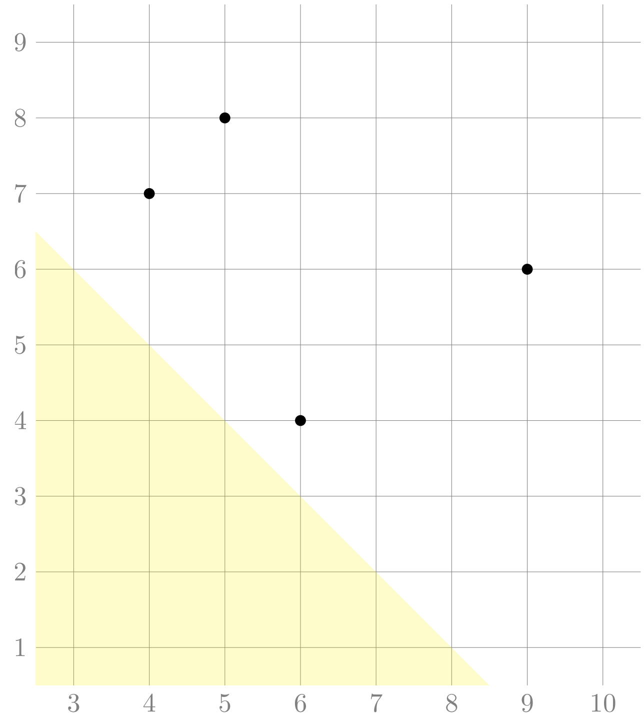
input	output
4 4 4 7 5 8 6 4 9 6 1 9 11 8 4 1 3 7 8 2 1 3 6 8 1 1 13 6 3 11	Yes Yes No No
4 8 4 7 5 8 6 4 9 6 1 9 11 8 4 1 3 7 8 2 1 3 6 8 1 1 13 6 3 11 2 6 4 5 8 2 6 6 7 7 2 7 5 9 11 2 5 3 6 6	Yes Yes No No Yes No Yes Yes

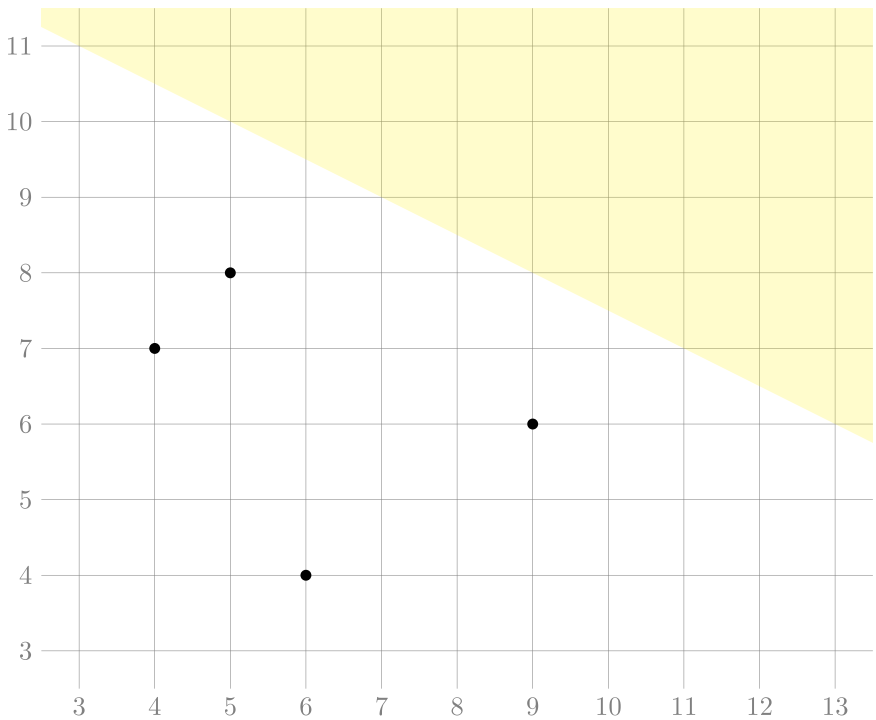
Note

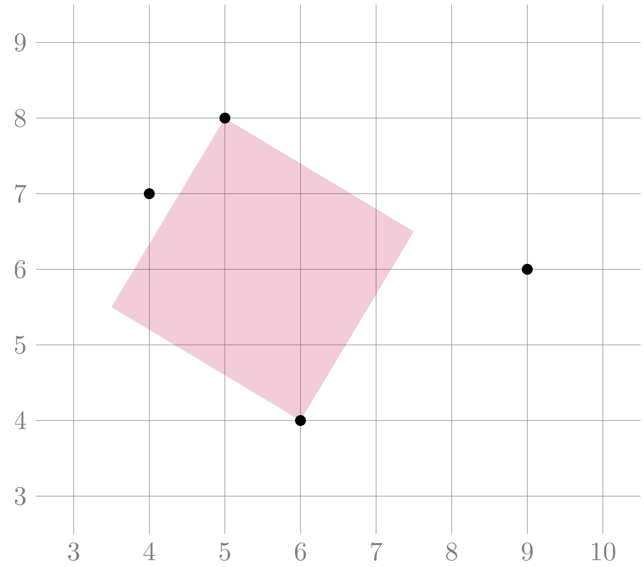
The following pictures show points and queries for all queries in the second example.

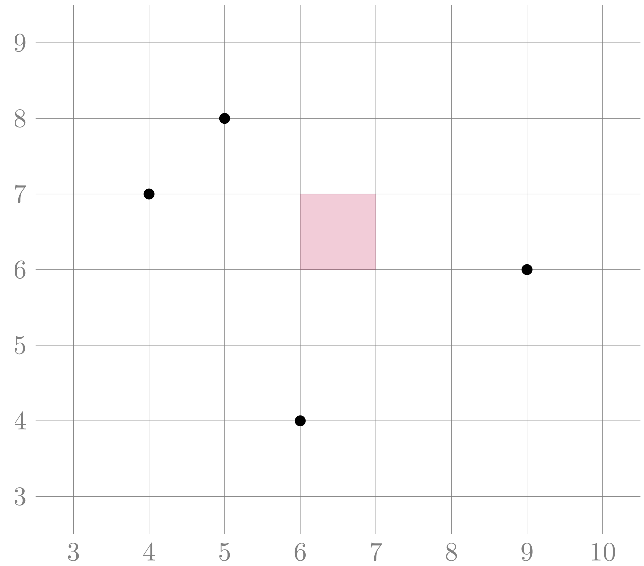


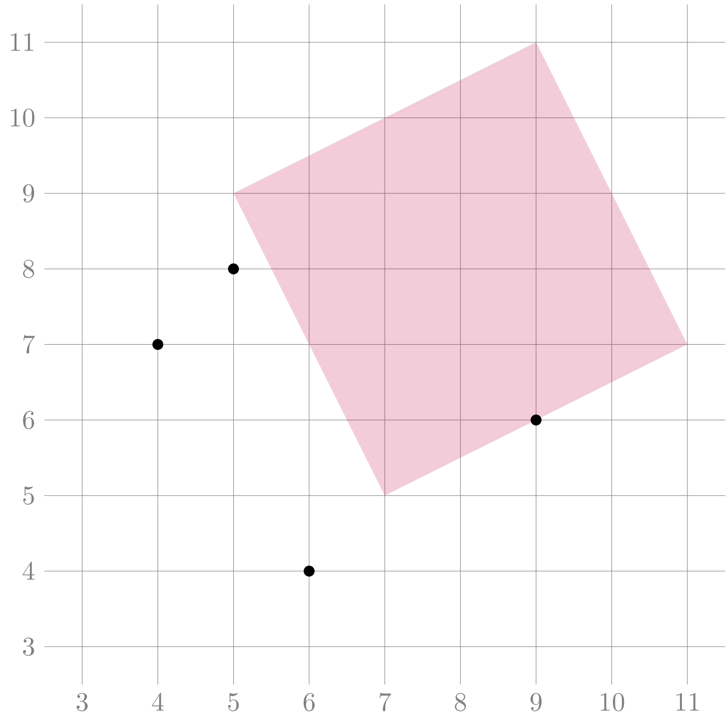


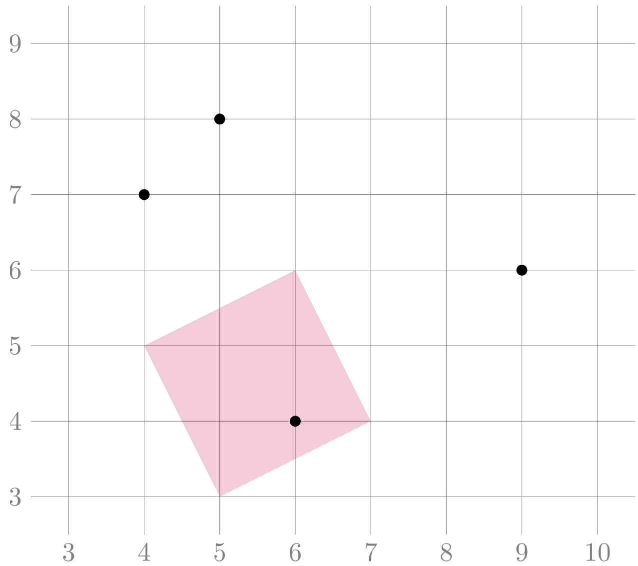

9
8
7
6
5
4
3
2
1
3
4
5
6
7
8
9
10


11
10
9
8
7
6
5
4
3
3
4
5
6
7
8
9
10
11
12
13


9
8
7
6
5
4
3
3 4 5 6 7 8 9 10


9
8
7
6
5
4
3
3 4 5 6 7 8 9 10


11
10
9
8
7
6
5
4
3
3 4 5 6 7 8 9 10 11


9
8
7
6
5
4
3
3
4
5
6
7
8
9
10