Codeforces Round #240 (Div. 1)

A. Mashmokh and Numbers

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

It's holiday. Mashmokh and his boss, Bimokh, are playing a game invented by Mashmokh.

In this game Mashmokh writes sequence of n distinct integers on the board. Then Bimokh makes several (possibly zero) moves. On the first move he removes the first and the second integer from from the board, on the second move he removes the first and the second integer of the remaining sequence from the board, and so on. Bimokh stops when the board contains less than two numbers. When Bimokh removes numbers x and y from the board, he gets $gcd(x, y)$ points. At the beginning of the game Bimokh has zero points.

Mashmokh wants to win in the game. For this reason he wants his boss to get exactly k points in total. But the guy doesn't know how choose the initial sequence in the right way.

Please, help him. Find n distinct integers $a_1, a_2, ..., a_n$ such that his boss will score exactly k points. Also Mashmokh can't memorize too huge numbers. Therefore each of these integers must be at most 10^9.

Input
The first line of input contains two space-separated integers n, k ($1 \le n \le 10^5$; $0 \le k \le 10^8$).

Output
If such sequence doesn't exist output -1 otherwise output n distinct space-separated integers $a_1, a_2, ..., a_n$ ($1 \le a_i \le 10^9$).

Examples

input
5 2
output
1 2 3 4 5

input
5 3
output
2 4 3 7 1

input
7 2
output
-1

Note
$gcd(x, y)$ is greatest common divisor of x and y.

B. Mashmokh and ACM

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Mashmokh's boss, Bimokh, didn't like Mashmokh. So he fired him. Mashmokh decided to go to university and participate in ACM instead of finding a new job. He wants to become a member of Bamokh's team. In order to join he was given some programming tasks and one week to solve them. Mashmokh is not a very experienced programmer. Actually he is not a programmer at all. So he wasn't able to solve them. That's why he asked you to help him with these tasks. One of these tasks is the following.

A sequence of l integers $b_1, b_2, ..., b_l$ $(1 \le b_1 \le b_2 \le ... \le b_l \le n)$ is called *good* if each number divides (without a remainder) by the next number in the sequence. More formally $b_i \mid b_{i+1}$ for all i $(1 \le i \le l - 1)$.

Given n and k find the number of good sequences of length k. As the answer can be rather large print it modulo 1000000007 $(10^9 + 7)$.

Input

The first line of input contains two space-separated integers n, k $(1 \le n, k \le 2000)$.

Output

Output a single integer — the number of good sequences of length k modulo 1000000007 $(10^9 + 7)$.

Examples

input
3 2
output
5

input
6 4
output
39

input
2 1
output
2

Note

In the first sample the good sequences are: $[1, 1], [2, 2], [3, 3], [1, 2], [1, 3]$.

C. Mashmokh and Reverse Operation

time limit per test: 4 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Mashmokh's boss, Bimokh, didn't like Mashmokh. So he fired him. Mashmokh decided to go to university and participate in ACM instead of finding a new job. He wants to become a member of Bamokh's team. In order to join he was given some programming tasks and one week to solve them. Mashmokh is not a very experienced programmer. Actually he is not a programmer at all. So he wasn't able to solve them. That's why he asked you to help him with these tasks. One of these tasks is the following.

You have an array a of length 2^n and m queries on it. The i-th query is described by an integer q_i. In order to perform the i-th query you must:

- split the array into $2^{n - q_i}$ parts, where each part is a subarray consisting of 2^{q_i} numbers; the j-th subarray $(1 \le j \le 2^{n - q_i})$ should contain the elements $a[(j - 1) \cdot 2^{q_i} + 1], a[(j - 1) \cdot 2^{q_i} + 2], \ldots, a[(j - 1) \cdot 2^{q_i} + 2^{q_i}]$;
- reverse each of the subarrays;
- join them into a single array in the same order (this array becomes new array a);
- output the number of inversions in the new a.

Given initial array a and all the queries. Answer all the queries. Please, note that the changes from some query is saved for further queries.

Input

The first line of input contains a single integer n $(0 \le n \le 20)$.

The second line of input contains 2^n space-separated integers $a[1], a[2], \ldots, a[2^n]$ $(1 \le a[i] \le 10^9)$, the initial array.

The third line of input contains a single integer m $(1 \le m \le 10^6)$.

The fourth line of input contains m space-separated integers $q_1, q_2, \ldots, q_m$ $(0 \le q_i \le n)$, the queries.

Note: since the size of the input and output could be very large, don't use slow output techniques in your language. For example, do not use input and output streams (cin, cout) in C++.

Output

Output m lines. In the i-th line print the answer (the number of inversions) for the i-th query.

Examples

input
2 2 1 4 3 4 1 2 0 2

output
0 6 6 0

input
1 1 2 3 0 1 1

output
0 1 0

Note

If we reverse an array $x[1], x[2], \ldots, x[n]$ it becomes new array $y[1], y[2], \ldots, y[n]$, where $y[i] = x[n - i + 1]$ for each i.

The number of inversions of an array $x[1], x[2], \ldots, x[n]$ is the number of pairs of indices i, j such that: $i < j$ and $x[i] > x[j]$.

D. Mashmokh and Water Tanks

time limit per test: 1 second
memory limit per test: 512 megabytes
input: standard input
output: standard output

Mashmokh is playing a new game. In the beginning he has k liters of water and p coins. Additionally he has a rooted tree (an undirected connected acyclic graph) that consists of m vertices. Each vertex of the tree contains a water tank that is empty in the beginning.

The game begins with the fact that Mashmokh chooses some (no more than k) of these tanks (except the root) and pours into each of them exactly 1 liter of water. Then the following process is performed until there is no water remained in tanks.

- The process consists of several steps.
- At the beginning of each step Mashmokh opens doors of all tanks. Then Mashmokh closes doors of some tanks (he is not allowed to close door of tank in the root) for the duration of this move. Let's denote the number of liters in some tank with closed door as w, Mashmokh pays w coins for the closing of that tank during this move.
- Let's denote by $x_1, x_2, \ldots, x_m$ as the list of vertices of the tree sorted (nondecreasing) by their depth. The vertices from this list should be considered one by one in the order. Firstly vertex x_1 (which is the root itself) is emptied. Then for each vertex x_i $(i > 1)$, if its door is closed then skip the vertex else move all the water from the tank of vertex x_i to the tank of its father (even if the tank of the father is closed).

Suppose l moves were made until the tree became empty. Let's denote the amount of water inside the tank of the root after the i-th move by w_i then Mashmokh will win $max(w_1, w_2, \ldots, w_l)$ dollars. Mashmokh wanted to know what is the maximum amount of dollars he can win by playing the above game. He asked you to find this value for him.

Input

The first line of the input contains three space-separated integers m, k, p $(2 \le m \le 10^5; 0 \le k, p \le 10^9)$.

Each of the following $m - 1$ lines contains two space-separated integers a_i, b_i $(1 \le a_i, b_i \le m; a_i \ne b_i)$ — the edges of the tree.

Consider that the vertices of the tree are numbered from 1 to m. The root of the tree has number 1.

Output

Output a single integer, the number Mashmokh asked you to find.

Examples

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

input
5 1000 1000 1 2 1 3 3 4 3 5
output
4

Note

The tree in the first sample is shown on the picture below. The black, red, blue colors correspond to vertices with 0, 1, 2 liters of water.

One way to achieve the maximum amount of money is to put 1 liter of water in each of vertices 3 and 4. The beginning state is shown on the picture below.

Then in the first move Mashmokh will pay one token to close the door of the third vertex tank. The tree after the first move is shown on the picture below.

After the second move there are 2 liters of water in the root as shown on the picture below.

E. Mashmokh's Designed Problem

time limit per test: 4 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

After a lot of trying, Mashmokh designed a problem and it's your job to solve it.

You have a tree T with n vertices. Each vertex has a unique index from 1 to n. The root of T has index 1. For each vertex of this tree v, you are given a list of its children in a specific order. You must perform three types of query on this tree:

1. find distance (the number of edges in the shortest path) between u and v;
2. given v and h, disconnect v from its father and connect it to its h-th ancestor; more formally, let's denote the path from v to the root by $x_1, x_2, \ldots, x_l$ $(h < l)$, so that $x_1 = v$ and x_l is root; disconnect v from its father (x_2) and connect it to x_{h+1}; vertex v must be added to the end of the child-list of vertex x_{h+1};
3. in the vertex sequence produced by calling function dfs(root) find the latest vertex that has distance k from the root.

The pseudo-code of function dfs(v):

```
// ls[v]: list of children of vertex v
// its i-th element is ls[v][i]
// its size is size(ls[v])
sequence result = empty sequence;
void dfs(vertex now)
{
   add now to end of result;
   for(int i = 1; i <= size(ls[v]); i = i + 1) //loop from i = 1 to i = size(ls[v])
      dfs(ls[v][i]);
}
```

Input

The first line of input contains two space-separated integers n, m $(2 \le n \le 10^5;\ 1 \le m \le 10^5)$, the number of vertices of T and number of queries to perform.

The i-th of the following n lines contains an integer l_i $(0 \le l_i \le n)$, number of i-th vertex's children. Then l_i space-separated integers follow, the j-th of them is the index of j-th child of i-th vertex. Note that the order of these vertices is important.

Each of the following m lines has one of the following format: "$1\ v\ u$", "$2\ v\ h$", or "$3\ k$". The first number in the line is the type of query to perform according to the problem statement. The next numbers are description of the query.

It's guaranteed that all the queries are correct. For example, in the second-type query h is at least 2 and at most distance of v from root. Also in the third-type query there is at least one vertex with distance k from the root at the time the query is given.

Output

For each query of the first or third type output one line containing the result of the query.

Examples

input
4 9 1 2 1 3 1 4 0 1 1 4 2 4 2 1 3 4 3 1 3 2 2 3 2 1 1 2 3 1 3 2

output
3 2 2 4 1 3 4

input
2 2 1 2 0 1 2 1 3 1
output
1 2