Codeforces Round #153 (Div. 1)

A. Points on Line

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Little Petya likes points a lot. Recently his mom has presented him n points lying on the line OX. Now Petya is wondering in how many ways he can choose three distinct points so that the distance between the two farthest of them doesn't exceed d.

Note that the order of the points inside the group of three chosen points doesn't matter.

Input

The first line contains two integers: n and d ($1 \le n \le 10^5$; $1 \le d \le 10^9$). The next line contains n integers $x_1, x_2, ..., x_n$, their absolute value doesn't exceed 10^9 — the x-coordinates of the points that Petya has got.

It is guaranteed that the coordinates of the points in the input **strictly increase**.

Output

Print a single integer — the number of groups of three points, where the distance between two farthest points doesn't exceed d.

Please do not use the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use the `cin`, `cout` streams or the `%I64d` specifier.

Sample test(s)

input
4 3 1 2 3 4
output
4

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

input
5 19 1 10 20 30 50
output
1

Note

In the first sample any group of three points meets our conditions.

In the seconds sample only 2 groups of three points meet our conditions: `{-3, -2, -1}` and `{-2, -1, 0}`.

In the third sample only one group does: `{1, 10, 20}`.

B. Playing with Permutations

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Little Petya likes permutations a lot. Recently his mom has presented him permutation $q_1, q_2, ..., q_n$ of length n.

A *permutation* a of length n is a sequence of integers $a_1, a_2, ..., a_n$ $(1 \le a_i \le n)$, all integers there are distinct.

There is only one thing Petya likes more than permutations: playing with little Masha. As it turns out, Masha also has a permutation of length n. Petya decided to get the same permutation, whatever the cost may be. For that, he devised a game with the following rules:

- Before the beginning of the game Petya writes permutation $1, 2, ..., n$ on the blackboard. After that Petya makes exactly k moves, which are described below.
- During a move Petya tosses a coin. If the coin shows heads, he performs point 1, if the coin shows tails, he performs point 2.

1. Let's assume that the board contains permutation $p_1, p_2, ..., p_n$ at the given moment. Then Petya removes the written permutation p from the board and writes another one instead: $p_{q_1}, p_{q_2}, ..., p_{q_n}$. In other words, Petya applies permutation q (which he has got from his mother) to permutation p.
2. All actions are similar to point 1, except that Petya writes permutation t on the board, such that: $t_{q_i} = p_i$ for all i from 1 to n. In other words, Petya applies a permutation that is inverse to q to permutation p.

We know that after the k-th move the board contained Masha's permutation $s_1, s_2, ..., s_n$. Besides, we know that throughout the game process Masha's permutation **never occurred on the board** before the k-th move. Note that the game has exactly k moves, that is, throughout the game the coin was tossed exactly k times.

Your task is to determine whether the described situation is possible or else state that Petya was mistaken somewhere. See samples and notes to them for a better understanding.

Input

The first line contains two integers n and k $(1 \le n, k \le 100)$. The second line contains n space-separated integers $q_1, q_2, ..., q_n$ $(1 \le q_i \le n)$ — the permutation that Petya's got as a present. The third line contains Masha's permutation s, in the similar format.

It is guaranteed that the given sequences q and s are correct permutations.

Output

If the situation that is described in the statement is possible, print "`YES`" (without the quotes), otherwise print "`NO`" (without the quotes).

Sample test(s)

input
4 1 2 3 4 1 1 2 3 4
output
NO

input
4 1 4 3 1 2 3 4 2 1
output
YES

input
4 3 4 3 1 2 3 4 2 1
output
YES

input
4 2 4 3 1 2 2 1 4 3
output
YES

input

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

output

```
NO
```

Note

In the first sample Masha's permutation coincides with the permutation that was written on the board before the beginning of the game. Consequently, that violates the condition that Masha's permutation never occurred on the board before k moves were performed.

In the second sample the described situation is possible, in case if after we toss a coin, we get tails.

In the third sample the possible coin tossing sequence is: heads-tails-tails.

In the fourth sample the possible coin tossing sequence is: heads-heads.

C. Number Transformation

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Little Petya likes positive integers a lot. Recently his mom has presented him a positive integer a. There's only one thing Petya likes more than numbers: playing with little Masha. It turned out that Masha already has a positive integer b. Petya decided to turn his number a into the number b consecutively performing the operations of the following two types:

1. Subtract 1 from his number.
2. Choose any integer x from 2 to k, inclusive. Then subtract number $(a \ mod \ x)$ from his number a. Operation $a \ mod \ x$ means taking the remainder from division of number a by number x.

Petya performs one operation per second. Each time he chooses an operation to perform during the current move, no matter what kind of operations he has performed by that moment. In particular, this implies that he can perform the same operation any number of times in a row.

Now he wonders in what minimum number of seconds he could transform his number a into number b. Please note that numbers x in the operations of the second type are selected anew each time, independently of each other.

Input

The only line contains three integers a, b $(1 \le b \le a \le 10^{18})$ and k $(2 \le k \le 15)$.

Please do not use the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use the `cin`, `cout` streams or the `%I64d` specifier.

Output

Print a single integer — the required minimum number of seconds needed to transform number a into number b.

Sample test(s)

input
10 1 4
output
6

input
6 3 10
output
2

input
1000000000000000000 1 3
output
666666666666666667

Note

In the first sample the sequence of numbers that Petya gets as he tries to obtain number b is as follows: 10 $\rightarrow$ 8 $\rightarrow$ 6 $\rightarrow$ 4 $\rightarrow$ 3 $\rightarrow$ 2 $\rightarrow$ 1.

In the second sample one of the possible sequences is as follows: 6 $\rightarrow$ 4 $\rightarrow$ 3.

D. Two Sets

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Little Petya likes numbers a lot. Recently his mother has presented him a collection of n non-negative integers. There's only one thing Petya likes more than numbers: playing with little Masha. He immediately decided to give a part of his new collection to her. To make the game even more interesting, Petya decided to give Masha such collection of numbers for which the following conditions fulfill:

- Let's introduce x_1 to denote the xor of all numbers Petya has got left; and let's introduce x_2 to denote the xor of all numbers he gave to Masha. Value $(x_1 + x_2)$ must be as large as possible.
- If there are multiple ways to divide the collection so that the previous condition fulfilled, then Petya minimizes the value x_1.

The xor operation is a bitwise excluding "OR", that is denoted as "xor" in the Pascal language and "^" in C/C++/Java.

Help Petya divide the collection as described above. If there are multiple suitable ways to divide it, find any of them. Please note that after Petya gives a part of his numbers to Masha, he may have no numbers left. The reverse situation is also possible, when Petya gives nothing to Masha. In both cases we must assume that the xor of an empty set of numbers equals 0.

Input

The first line contains integer n ($1 \le n \le 10^5$), showing how many numbers Petya's mother gave him. The second line contains the actual space-separated numbers. They are all integer, non-negative and do not exceed 10^{18}.

Output

Print n space-separated integers, the i-th of them should equal either 1, if Petya keeps the number that follows i-th in his collection, or it should equal 2, if Petya gives the corresponding number to Masha. The numbers are indexed in the order in which they are given in the input.

Sample test(s)

input
6 1 2 3 4 5 6
output
2 2 2 2 2 2

input
3 1000000000000 1000000000000 1000000000000
output
2 2 2

input
8 1 1 2 2 3 3 4 4
output
1 2 1 2 2 2 1 2

E. Tree and Table

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Little Petya likes trees a lot. Recently his mother has presented him a tree with $2n$ nodes. Petya immediately decided to place this tree on a rectangular table consisting of 2 rows and n columns so as to fulfill the following conditions:

1. Each cell of the table corresponds to exactly one tree node and vice versa, each tree node corresponds to exactly one table cell.
2. If two tree nodes are connected by an edge, then the corresponding cells have a common side.

Now Petya wonders how many ways are there to place his tree on the table. He calls two placements distinct if there is a tree node which corresponds to distinct table cells in these two placements. Since large numbers can scare Petya, print the answer modulo 1000000007 $(10^9 + 7)$.

Input

The first line contains a single integer n $(1 \le n \le 10^5)$. Next $(2n - 1)$ lines contain two integers each a_i and b_i $(1 \le a_i, b_i \le 2n;\ a_i \ne b_i)$ that determine the numbers of the vertices connected by the corresponding edge.

Consider the tree vertexes numbered by integers from 1 to $2n$. It is guaranteed that the graph given in the input is a tree, that is, a connected acyclic undirected graph.

Output

Print a single integer — the required number of ways to place the tree on the table modulo 1000000007 $(10^9 + 7)$.

Sample test(s)

input

```
3
1 3
2 3
4 3
5 1
6 2
```

output

```
12
```

input

```
4
1 2
2 3
3 4
4 5
5 6
6 7
7 8
```

output

```
28
```

input

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

output

```
0
```

Note

Note to the first sample (all 12 variants to place the tree on the table are given below):

```
1-3-2    2-3-1    5 4 6    6 4 5
| | |    | | |    | | |    | | |
5 4 6    6 4 5    1-3-2    2-3-1

4-3-2    2-3-4    5-1 6    6 1-5
  | |    | |        | |    | |
5-1 6    6 1-5    4-3-2    2-3-4

1-3-4    4-3-1    5 2-6    6-2 5
| |        | |    | |        | |
5 2-6    6-2 5    1-3-4    4-3-1
```