A. Selection of Personnel

time limit per test: 0.5 seconds
memory limit per test: 64 megabytes

One company of IT City decided to create a group of innovative developments consisting from 5 to 7 people and hire new employees for it. After placing an advertisment the company received n resumes. Now the HR department has to evaluate each possible group composition and select one of them. Your task is to count the number of variants of group composition to evaluate.

Input

The only line of the input contains one integer n ($7 \le n \le 777$) — the number of potential employees that sent resumes.

Output

Output one integer — the number of different variants of group composition.

Examples

input	output
7	29

B. Caesar's Legions

time limit per test: 2 seconds

memory limit per test: 256 megabytes

Gaius Julius Caesar, a famous general, loved to line up his soldiers. Overall the army had n_1 footmen and n_2 horsemen. Caesar thought that an arrangement is **not** beautiful if somewhere in the line there are strictly more that k_1 footmen standing successively one after another, or there are strictly more than k_2 horsemen standing successively one after another. Find the number of beautiful arrangements of the soldiers.

Note that all $n_1 + n_2$ warriors should be present at each arrangement. All footmen are considered indistinguishable among themselves. Similarly, all horsemen are considered indistinguishable among themselves.

Input

The only line contains four space-separated integers n_1, n_2, k_1, k_2 $(1 \le n_1, n_2 \le 100, 1 \le k_1, k_2 \le 10)$ which represent how many footmen and horsemen there are and the largest acceptable number of footmen and horsemen standing in succession, correspondingly.

Output

Print the number of beautiful arrangements of the army modulo 100000000 (10^8). That is, print the number of such ways to line up the soldiers, that no more than k_1 footmen stand successively, and no more than k_2 horsemen stand successively.

Examples

input	output
2 1 1 10	1

input	output
2 3 1 2	5

input	output
2 4 1 1	0

Note

Let's mark a footman as `1`, and a horseman as `2`.

In the first sample the only beautiful line-up is: `121`

In the second sample 5 beautiful line-ups exist: `12122`, `12212`, `21212`, `21221`, `22121`

C. Directed Roads

time limit per test: 2 seconds
memory limit per test: 256 megabytes

ZS the Coder and Chris the Baboon has explored Udayland for quite some time. They realize that it consists of n towns numbered from 1 to n.

There are n directed roads in the Udayland. i-th of them goes from town i to some other town a_i ($a_i \neq i$). ZS the Coder can flip the direction of any road in Udayland, i.e. if it goes from town A to town B before the flip, it will go from town B to town A after.

ZS the Coder considers the roads in the Udayland *confusing*, if there is a sequence of distinct towns $A_1, A_2, ..., A_k$ ($k > 1$) such that for every $1 \le i < k$ there is a road from town A_i to town A_{i+1} and another road from town A_k to town A_1. In other words, the roads are confusing if **some of them** form a directed cycle of some towns.

Now ZS the Coder wonders how many sets of roads (there are 2^n variants) in initial configuration can he choose to flip such that after flipping each road in the set exactly once, the resulting network will **not** be confusing.

Note that it is allowed that after the flipping there are more than one directed road from some town and possibly some towns with no roads leading out of it, or multiple roads between any pair of cities.

Input

The first line of the input contains single integer n ($2 \le n \le 2 \cdot 10^5$) — the number of towns in Udayland.

The next line contains n integers $a_1, a_2, ..., a_n$ ($1 \le a_i \le n, a_i \neq i$), a_i denotes a road going from town i to town a_i.

Output

Print a single integer — the number of ways to flip some set of the roads so that the resulting whole set of all roads is not confusing. Since this number may be too large, print the answer modulo $10^9 + 7$.

Examples

input	output
3 2 3 1	6

input	output
4 2 1 1 1	8

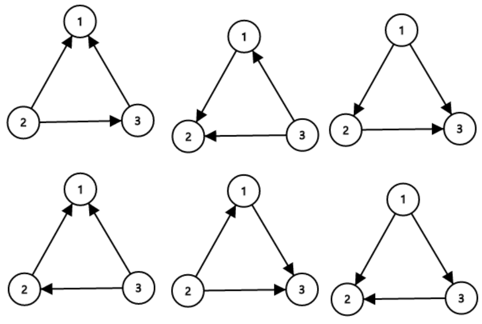
input	output
5 2 4 2 5 3	28

Note

Consider the first sample case. There are 3 towns and 3 roads. The towns are numbered from 1 to 3 and the roads are $1 \to 2$, $2 \to 3$, $3 \to 1$ initially. Number the roads 1 to 3 in this order.

The sets of roads that ZS the Coder can flip (to make them not confusing) are $\{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}$. Note that the empty set is invalid because if no roads are flipped, then towns $1, 2, 3$ is form a directed cycle, so it is confusing. Similarly, flipping all roads is confusing too. Thus, there are a total of 6 possible sets ZS the Coder can flip.

The sample image shows all possible ways of orienting the roads from the first sample such that the network is **not confusing**.



D. Riding in a Lift

time limit per test: 2 seconds
memory limit per test: 256 megabytes

Imagine that you are in a building that has exactly n floors. You can move between the floors in a lift. Let's number the floors from bottom to top with integers from 1 to n. Now you're on the floor number a. You are very bored, so you want to take the lift. Floor number b has a secret lab, the entry is forbidden. However, you already are in the mood and decide to make k consecutive trips in the lift.

Let us suppose that at the moment you are on the floor number x (initially, you were on floor a). For another trip between floors you choose some floor with number y ($y \neq x$) and the lift travels to this floor. As you cannot visit floor b with the secret lab, you decided that the distance from the current floor x to the chosen y must be strictly less than the distance from the current floor x to floor b with the secret lab. Formally, it means that the following inequation must fulfill: $|x - y| < |x - b|$. After the lift successfully transports you to floor y, you write down number y in your notepad.

Your task is to find the number of distinct number sequences that you could have written in the notebook as the result of k trips in the lift. As the sought number of trips can be rather large, find the remainder after dividing the number by 1000000007 ($10^9 + 7$).

Input

The first line of the input contains four space-separated integers n, a, b, k ($2 \le n \le 5000$, $1 \le k \le 5000$, $1 \le a, b \le n$, $a \neq b$).

Output

Print a single integer — the remainder after dividing the sought number of sequences by 1000000007 ($10^9 + 7$).

Examples

input	output
5 2 4 1	2

input	output
5 2 4 2	2

input	output
5 3 4 1	0

Note

Two sequences $p_1, p_2, ..., p_k$ and $q_1, q_2, ..., q_k$ are *distinct*, if there is such integer j ($1 \le j \le k$), that $p_j \neq q_j$.

Notes to the samples:

1. In the first sample after the first trip you are either on floor 1, or on floor 3, because $|1 - 2| < |2 - 4|$ and $|3 - 2| < |2 - 4|$.
2. In the second sample there are two possible sequences: $(1, 2)$; $(1, 3)$. You cannot choose floor 3 for the first trip because in this case no floor can be the floor for the second trip.
3. In the third sample there are no sought sequences, because you cannot choose the floor for the first trip.

E. Clues

time limit per test: 2 seconds
memory limit per test: 256 megabytes

As Sherlock Holmes was investigating another crime, he found a certain number of clues. Also, he has already found direct links between some of those clues. The direct links between the clues are mutual. That is, the direct link between clues A and B and the direct link between clues B and A is the same thing. No more than one direct link can exist between two clues.

Of course Sherlock is able to find direct links between all clues. But it will take too much time and the criminals can use this extra time to hide. To solve the crime, Sherlock needs each clue to be linked to all other clues (maybe not directly, via some other clues). Clues A and B are considered linked either if there is a direct link between them or if there is a direct link between A and some other clue C which is linked to B.

Sherlock Holmes counted the minimum number of additional direct links that he needs to find to solve the crime. As it turns out, it equals T.

Please count the number of different ways to find exactly T direct links between the clues so that the crime is solved in the end. Two ways to find direct links are considered different if there exist two clues which have a direct link in one way and do not have a direct link in the other way.

As the number of different ways can turn out rather big, print it modulo k.

Input

The first line contains three space-separated integers n, m, k ($1 \le n \le 10^5$, $0 \le m \le 10^5$, $1 \le k \le 10^9$) — the number of clues, the number of direct clue links that Holmes has already found and the divisor for the modulo operation.

Each of next m lines contains two integers a and b ($1 \le a, b \le n$, $a \ne b$), that represent a direct link between clues. It is guaranteed that any two clues are linked by no more than one direct link. Note that the direct links between the clues are mutual.

Output

Print the single number — the answer to the problem modulo k.

Examples

input	output
2 0 1000000000	1

input	output
3 0 100	3

input	output
4 1 1000000000 1 4	8

Note

The first sample only has two clues and Sherlock hasn't found any direct link between them yet. The only way to solve the crime is to find the link.

The second sample has three clues and Sherlock hasn't found any direct links between them. He has to find two of three possible direct links between clues to solve the crime — there are 3 ways to do it.

The third sample has four clues and the detective has already found one direct link between the first and the fourth clue. There are 8 ways to find two remaining clues to solve the crime.

F – Counting heaps

We are given a rooted tree of n vertices. The vertices are to be labeled with numbers $1, 2, \ldots, n$ so that each label is unique and the heap condition holds, i.e. the label of any vertex is less than the label of its parent. How many such labellings exist? Since this number may be quite large, calculate only its remainder modulo m.

Input

The input contains several tree descriptions. The first line contains the number of input trees t ($t \leq 250$). Each tree description begins with a line containing the size of the tree n ($1 \leq n \leq 500000$) and an integer m ($2 \leq m \leq 10^9$). $n-1$ lines follow, i-th of which contains $p(i+1)$, the number of the parent of the $i+1$-th vertex ($1 \leq p(i+1) \leq i$). Vertex number 1 will be the root in each tree, so its parent will not be given. Total size of the input will not exceed 50MB.

Output

For each tree output the number of its valid labellings modulo given m.

Example

Input	Output
4 3 1000000 1 1 4 1000000 1 1 1 5 1000000 1 2 3 4 5 1000000 1 1 3 3	2 6 1 8

The 8 possible labellings from the last example test case are as follows: