AIM Tech Round 3 (Div. 1)

A. Letters Cyclic Shift

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

You are given a non-empty string s consisting of lowercase English letters. You have to pick **exactly one non-empty substring** of s and shift all its letters 'z' 'y' 'x' 'b' 'a' 'z'. In other words, each character is replaced with the previous character of English alphabet and 'a' is replaced with 'z'.

What is the lexicographically minimum string that can be obtained from s by performing this shift exactly once?

Input

The only line of the input contains the string s ($1 \le |s| \le 100\,000$) consisting of lowercase English letters.

Output

Print the lexicographically minimum string that can be obtained from s by shifting letters of exactly one non-empty substring.

Examples

input
codeforces
output
bncdenqbdr

input
abacaba
output
aaacaba

Note

String s is lexicographically smaller than some other string t of the same length if there exists some $1 \le i \le |s|$, such that $s_1 = t_1, s_2 = t_2, \ldots, s_{i-1} = t_{i-1}$, and $s_i < t_i$.

B. Recover the String

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

For each string s consisting of characters '0' and '1' one can define four integers a_{00}, a_{01}, a_{10} and a_{11}, where a_{xy} is the number of **subsequences** of length 2 of the string s equal to the sequence $\{x, y\}$.

In these problem you are given four integers a_{00}, a_{01}, a_{10}, a_{11} and have to find any non-empty string s that matches them, or determine that there is no such string. One can prove that if at least one answer exists, there exists an answer of length no more than $1\,000\,000$.

Input

The only line of the input contains four non-negative integers a_{00}, a_{01}, a_{10} and a_{11}. Each of them doesn't exceed 10^9.

Output

If there exists a non-empty string that matches four integers from the input, print it in the only line of the output. Otherwise, print "Impossible". The length of your answer must not exceed $1\,000\,000$.

Examples

input
1 2 3 4
output
Impossible

input
1 2 2 1
output
0110

C. Centroids

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

Tree is a connected acyclic graph. Suppose you are given a tree consisting of n vertices. The vertex of this tree is called *centroid* if the size of each connected component that appears if this vertex is removed from the tree doesn't exceed .

You are given a tree of size n and can perform no more than one edge replacement. *Edge replacement* is the operation of removing one edge from the tree (without deleting incident vertices) and inserting one new edge (without adding new vertices) in such a way that the graph remains a tree. For each vertex you have to determine if it's possible to make it centroid by performing no more than one edge replacement.

Input

The first line of the input contains an integer n ($2 \le n \le 400\,000$) — the number of vertices in the tree. Each of the next $n - 1$ lines contains a pair of vertex indices u_i and v_i ($1 \le u_i, v_i \le n$) — endpoints of the corresponding edge.

Output

Print n integers. The i-th of them should be equal to 1 if the i-th vertex can be made centroid by replacing no more than one edge, and should be equal to 0 otherwise.

Examples

input
3 1 2 2 3
output
1 1 1

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

Note

In the first sample each vertex can be made a centroid. For example, in order to turn vertex 1 to centroid one have to replace the edge $(2, 3)$ with the edge $(1, 3)$.

D. Incorrect Flow

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

At the entrance examination for the magistracy of the MSU Cyber-Mechanics Department Sasha got the question about Ford-Fulkerson algorithm. He knew the topic perfectly as he worked with it many times on programming competition. As the task for the question he was given a network with partially build flow that he had to use in order to demonstrate the workflow of the algorithm. He quickly finished to write the text and took a look at the problem only to understand that the given network is incorrect!

Suppose you are given a directed graph $G(V, E)$ with two special nodes s and t called source and sink. We denote as n the number of nodes in the graph, i.e. $n = |V|$ and m stands for the number of directed edges in the graph, i.e. $m = |E|$. For the purpose of this problem we always consider node 1 to be the source and node n to be the sink. In addition, for each edge of the graph e we define the capacity function $c(e)$ and flow function $f(e)$. Function $f(e)$ represents the correct flow if the following conditions are satisfied:

1. For each edge the flow is non-negative and does not exceed capacity $c(e)$, i.e. $0 \le f(e) \le c(e)$.
2. For each node , that is not source or sink ($v \neq s$ and $v \neq t$) the sum of flows of all edges going in v is equal to the sum of the flows among all edges going out from v. In other words, there is no flow stuck in v.

It was clear that as the exam was prepared last night and there are plenty of mistakes in the tasks. Sasha asked one of the professors to fix the network or give the correct task, but the reply was that the magistrate student should be able to fix the network himself. As the professor doesn't want the task to become easier, he asks Sasha to fix the network in a such way that the total number of changes is minimum possible. Sasha is not allowed to remove edges, add new ones or reverse the direction of existing edges. The only thing he is able to do is to change capacity function $c(e)$ and flow function $f(e)$. Moreover, all the values should remain non-negative integers. There is no requirement on the flow to be maximum in any sense.

Find the minimum possible total change of the functions $f(e)$ and $c(e)$ that Sasha has to make in order to make the flow correct. The total change is defined as the sum of absolute differences, i.e. if new functions are $f^*(e)$ and $c^*(e)$, then the total change is .

Input

The first line of the input contains two integers n and m ($2 \le n \le 100$, $0 \le m \le 100$) — the number of nodes and edges in the graph respectively. Each of the following m lines contains the description of the edges, consisting of four integers u_i, v_i, c_i and f_i ($1 \le u_i, v_i \le n$, $u_i \neq v_i$, $0 \le c_i, f_i \le 1\,000\,000$) — index of the node the edges starts from, the index of the node the edge goes to, current capacity and flow value.

Node number 1 is the source, and node number n is the sink. It's guaranteed that no edge goes to the source, and no edges starts in the sink.

Given graph contains no self-loops but may contain multiple edges.

Output

Print one integer — the minimum total sum of changes that Sasha has to do in order to get the correct flow description.

Examples

input
2 1 1 2 2 1
output
0

input
2 1 1 2 1 2
output
1

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

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

Note

In the first sample, the flow is initially correct. Note, that the flow is not maximum, but this is not required.

In the second sample, the flow value of the only edge is greater than its capacity. There are two ways to fix this: either increase the capacity up to 2 or reduce the flow down to 1.

In the third sample, there is only 1 unit of flow coming to vertex 2, but there are 2 units going out of it. One of the possible solutions is to reduce the value of the flow on the second edge by 1.

In the fourth sample, there is isolated circulation of flow, but this description is correct by definition.

E. Student's Camp

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Alex studied well and won the trip to student camp Alushta, located on the seashore.

Unfortunately, it's the period of the strong winds now and there is a chance the camp will be destroyed! Camp building can be represented as the rectangle of $n + 2$ concrete blocks height and m blocks width.

Every day there is a breeze blowing from the sea. Each block, except for the blocks of the upper and lower levers, such that there is no block to the left of it is destroyed with the probability . Similarly, each night the breeze blows in the direction to the sea. Thus, each block (again, except for the blocks of the upper and lower levers) such that there is no block to the right of it is destroyed with the same probability p. Note, that blocks of the upper and lower level are **indestructible**, so there are only $n \cdot m$ blocks that can be destroyed.

The period of the strong winds will last for k days and k nights. If during this period the building will split in at least two connected components, it will collapse and Alex will have to find another place to spend summer.

Find the probability that Alex won't have to look for other opportunities and will be able to spend the summer in this camp.

Input

The first line of the input contains two integers n and m $(1 \le n, m \le 1500)$ that define the size of the destructible part of building.

The second line of the input contains two integers a and b $(1 \le a \le b \le 10^9)$ that define the probability p. It's guaranteed that integers a and b are coprime.

The third line contains a single integer k $(0 \le k \le 100\,000)$ — the number of days and nights strong wind will blow for.

Output

Consider the answer as an irreducible fraction is equal to . Print one integer equal to . It's guaranteed that within the given constraints .

Examples

input
2 2 1 2 1
output
937500007

input
5 1 3 10 1
output
95964640

input
3 3 1 10 5
output
927188454

Note

In the first sample, each of the four blocks is destroyed with the probability . There are 7 scenarios that result in building not collapsing, and the probability we are looking for is equal to , so you should print