Locomotive

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

A railway network consists of nodes connected by a set of railway tracks. Each track can be passed in one direction only. Two nodes may be connected by an arbitrary number of tracks, but no track connects a node to itself.

When the first carriage of the train (called the locomotive) reaches the end-node of a track, an engine driver should select a track going out from this node and move to it. However, there are some restrictions that come from the type of the track that the train is passing now:

1. Straight track — for this type of tracks no choice can be made as there is only one possible track to go forward, regardless of how many tracks go out of this node.
2. Track with a switch — at the end of this track the driver is to decide which one of two tracks to take.

Regardless of the type all tracks have a length of 1 kilometer.

You are given a long cargo train consisting of k carriages (including the locomotive). Each carriage has a length of 25 meters, while the space between carriages can be ignored and not taken into account in the following calculations. Initially, the train is positioned in the departure depot which is located outside of the scope of the railway network, and the head of locomotive is exactly at the node 1.

At the beginning of the ride train can take any track that starts from the node 1. The goal is to ride to the node n as fast as possible. When the head of the train reaches the node n locomotive and all other carriages go to arrival depot outside of the scope of the railway network. All the carriages move simultaneously and with the constant speed.

What is the minimal distance the train must go in order to finish the trip? There is an important restriction that the train cannot intersect itself: only one point of the train can be present in any node at any fixed moment of time. The only exception is: the very first point of the locomotive and the very last point of the train's tail can share the same node. See the samples for further clarification.

Input

The first line of the input contains three integers n, m and k ($2 \le n, m, k \le 500$) — the number of nodes and tracks in the given railway network, and the number of carriages in a train respectively.

Following that are m lines describing the tracks. Every track description starts with three integers u_i, v_i and c_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$, $1 \le c_i \le 2$) — the starting node of this track, the finishing node and the number of tracks the driver can choose to continue his path. Next go c_i distinct integers $r_{i,j}$ ($1 \le r_{i,j} \le m$) — the track numbers the train can go to after passing the i-th track. It's guaranteed that all these tracks are starting in the node v_i.

Output

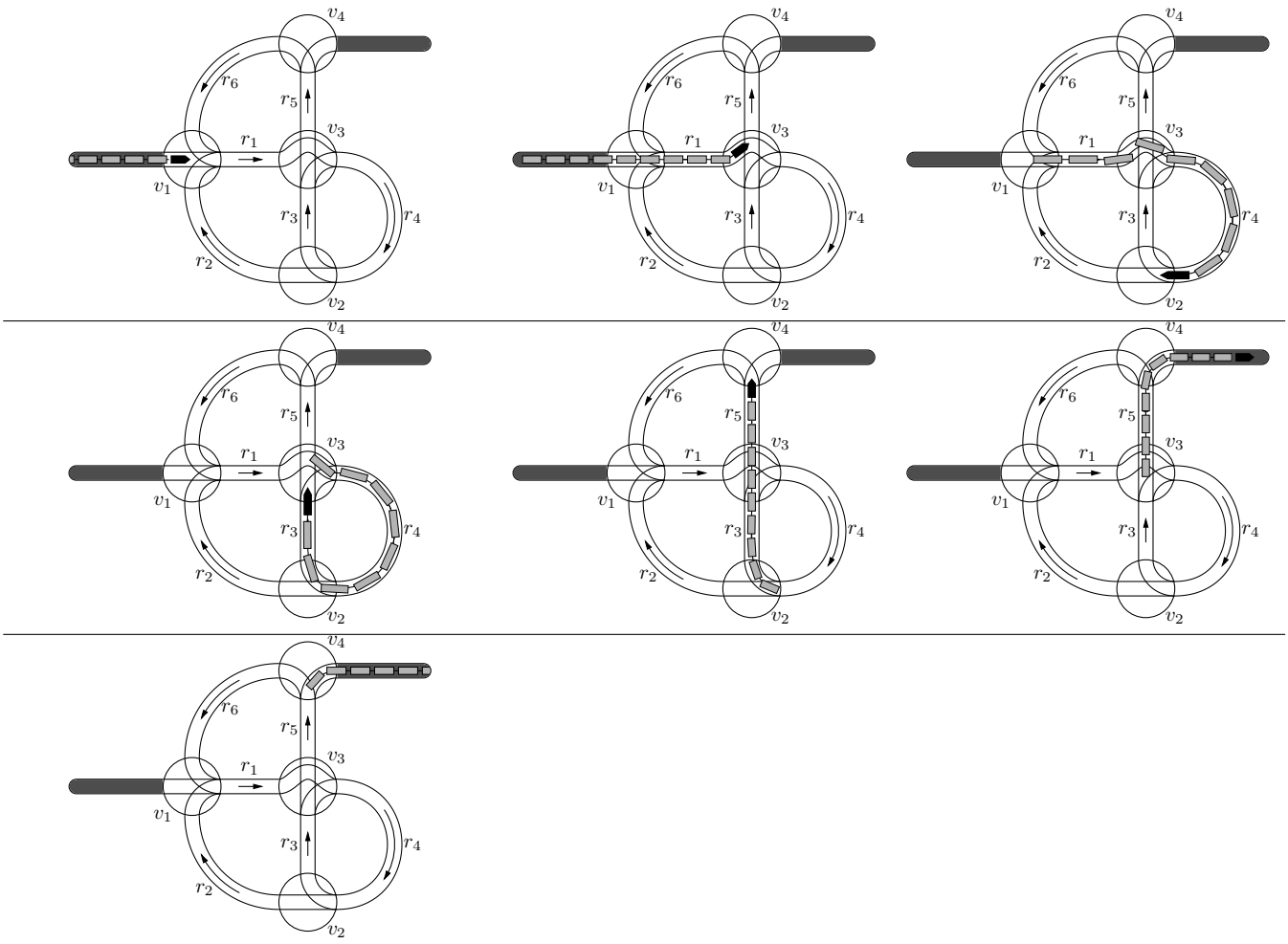
Output a single integer — the minimal distance in meters that the train must pass. If there is no way for the train to finish the journey in the depot located at the node n print "`No chance`" (without quotes).

Examples

standard input	standard output
4 6 80 1 3 1 4 2 1 1 1 2 3 2 4 5 3 2 2 2 3 3 4 1 6 4 1 1 1	6000
6 7 150 1 2 1 2 2 3 1 3 3 4 1 4 4 5 1 5 5 3 1 6 3 6 1 7 6 3 1 6	No chance

Note

In the first sample the optimal path of the train is `departure depot` $\to 1 \to 3 \to 2 \to 3 \to 4 \to$ `arrival depot`. The train movement is shown on the pictures below, where the train is depicted with 10 carriages.



In the second sample the train cannot avoid intersecting itself in the node 3, so there is no possible way to reach the arrival depot.