

ENGLISH

Speed Limits

In our busy world, we are not usually interested in taking the shortest path, but rather the path that takes the shortest time. When driving a car, this means that the speed limits on different roads are crucial. Imagine now that some speed limit signs are missing. Since you cannot expect the driver to know speed limits by heart, the only reasonable conclusion is that whatever speed limit he/she obeyed before will still hold after passing a missing sign. You are to write a program that calculates the fastest path by taking advantage of the missing signs.

You are given a description of the road network in a highly motorized area. To make things easier the network consists of *crossings* and *roads*. Every road is one-way, connects exactly two crossings and has at most one speed limit sign, located in the beginning of the road. For any two crossings A and B, there exists at most one road from A to B. You may assume that acceleration is instantaneous and that no other traffic will affect you. And of course you never drive faster than the current speed limit.

Input data

The first line of the input file speed.in consists of three integers N, M and D, where N ($2 \le N \le 150$) is the number of crossings numbered from 0 to N-1, M is the number of roads and D is the label of the crossing that is your destination. Each of the following M lines of the input file describes one road. Every line consists of four integers A ($0 \le A < N$), B ($0 \le B < N$), V ($0 \le V \le 500$) and L ($1 \le L \le 500$), signifying that the road goes from the crossing labelled by A to crossing B, has speed limit V and length L. If V is zero, this means that the speed limit sign is missing. The time T taken for a given road is thus T = L / V if $V \ne 0$, otherwise $T = L / V_{old}$, where V_{old} is the speed limit you obeyed before you arrived at the crossing. Note that the division should be done with floating-point numbers to avoid unnecessary rounding. In the beginning you are at crossing 0 and your current speed is 70.

Output data

The output file speed.out must consist of one single line of integers, describing the crossings you pass on the fastest possible path from 0 to D. The crossings should be written in the exact order in which you pass them, starting with 0 and ending with D. There will never be two fastest paths taking the same time.

Task Description BOI 2002 Vilnius Lithuania



ENGLISH

Example

speed.in	speed.out	Comments
6 15 1 0 1 25 68	0 5 2 3 1	The required time is in this case 2,628 units.
0 2 30 50		
0 5 0 101		
1 2 70 77		
1 3 35 42		
2 0 0 22		
2 1 40 86		
2 3 0 23		
2 4 45 40		