

Dimensions

Today Ivancho got together with friends to play a game of “Dimensions”. The rules of the game are:

- There are **N** dimensions – each **N** by **N** in size.
- There are rocks (fields 1 by 1) in each dimension for the character to step on. Everything else is lava or chasms.
- The movement inside a dimension is determined by jumping between two rocks where there is a maximum jump range – **M**.
- The distance between two rocks is the direct distance between them.
- If there are two or more rocks with equal coordinates but different dimensions, the distance between any two of them is 1. This is the only way to move between dimensions.
- There are also a start and an end rock where the goal is to reach the end faster than the other players.

Unfortunately Ivancho is not as good as his friends at the game. So he asks you for a little cheat – write a program **dimensions**, which finds the shortest path from the beginning to the end by following the rules.

Input: On the first row of the input file **dimensions.in** are given three numbers – **N**, **M** and **P**. **N** is the number and the size of the dimensions, **M** is the length of the jump and is always an integer. **P** is the total number of rocks in the game. On the next **P** rows are given X_p , Y_p and Z_p – each rock’s coordinates where Z_p is the respective dimension. The next two rows contain the coordinates of the start and end rock – X_s , Y_s , Z_s and X_e , Y_e , Z_e .

Remark: All coordinates are indexed by numbers from 0 to $N-1$. There cannot be more than one rock using the same coordinates in the same dimension.

Output: The output file **dimensions.out** should contain a positive integer **K** – the number of rocks that determine the path. The next **K** rows should contain the coordinates of each rock where the first and the last one should be respectively the start and the end rock. If there isn’t a path **K** is 0. If there is more than one shortest path print any one of them.

Limits:

$N \leq 512$

$M \leq N$

$P \leq 800$

Time limit: 0.2 sec

Memory limit: 256 MB

Preliminary tests: 4

Final tests: 10

Example test:

dimensions.in	dimensions.out	
3 5 8 0 1 0 2 1 0 0 2 0 0 2 1 2 1 1 2 1 2 2 2 2 0 2 2 0 1 0 2 2 2	4 0 1 0 2 1 0 2 1 2 2 2 2	4 0 1 0 0 2 0 0 2 2 2 2 2

In this case both solutions are valid.