



Finding your destination in a big unknown city can be challenging, especially if you are a computer scientist like Kirk, always trying to use the shortest possible path. Planning can help – given the map of the city Kirk wants to find the shortest path between his current position and his destination.

The map of the city can be represented in the plane as an infinite grid composed of unit squares.

Kirk is currently located at the square  $(0, 0)$  and his destination is the square  $(X, Y)$ .

There are  $N$  buildings in the city. Each building is a rectangle fully occupying a number of unit squares. **No two buildings touch or overlap**, i.e. Kirk can walk freely around every building. A building is defined by specifying the coordinates of two diagonally opposite squares occupied by the building.

In each step, Kirk can walk to one of the four neighboring squares, but he is not allowed to step onto a square occupied by a building. His current position is at the west entrance to the city and the  $x$  coordinate of every square occupied by a building is **strictly greater than zero**.

Write a program that, given the locations of the buildings, finds **one shortest possible path** from Kirk's current position to his destination. A path should be reported as a sequence of vertical and horizontal segments, with no two consecutive segments parallel. The length of a path is the number of squares contained in the path, excluding the initial square.

### INPUT

The first line of input contains two integers  $X, Y$  ( $1 \leq X \leq 10^6$ ,  $-10^6 \leq Y \leq 10^6$ ) – the coordinates of the destination square. The second line of input contains a single integer  $N$  ( $0 \leq N \leq 100\,000$ ) – the number of buildings in the city. Each of the following  $N$  lines contains four integers  $X1, Y1, X2, Y2$  ( $1 \leq X1, X2 \leq 10^6$ ,  $-10^6 \leq Y1, Y2 \leq 10^6$ ) – the coordinates of two diagonally opposite squares occupied by the building.

### OUTPUT

The first line of output should contain an integer  $L$  – the length of the shortest path to the destination.

The second line of output should contain an integer  $M$  – the number of segments in the shortest path. The number of segments  $M$  must not exceed 1,000,000.

Each of the following  $M$  lines should contain two integers,  $DX$  and  $DY$ , describing Kirk's relative movement in one segment. For each segment, **exactly one** of the values  $DX, DY$  should be zero, and no two consecutive segments should be parallel.

**Note:** if there are multiple solutions, you should output any one of them.

### GRADING

Partial credit is awarded for incorrect or incomplete solutions that correctly find the length of the shortest possible path.

If the minimum length is correct, you will receive 80% of the points for the corresponding test case.

If you choose to find the minimum length only, you do not need to output anything else.



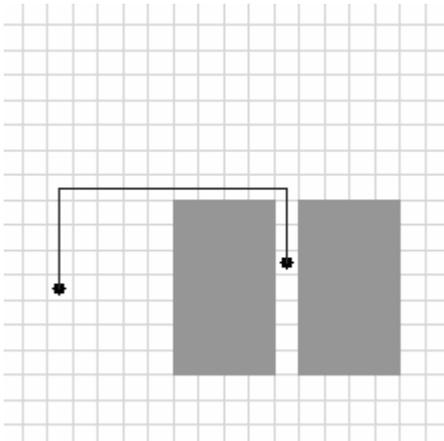
### EXAMPLES

**input**

```
9 1
2
5 -3 8 3
10 -3 13 3
```

**output**

```
16
3
0 4
9 0
0 -3
```

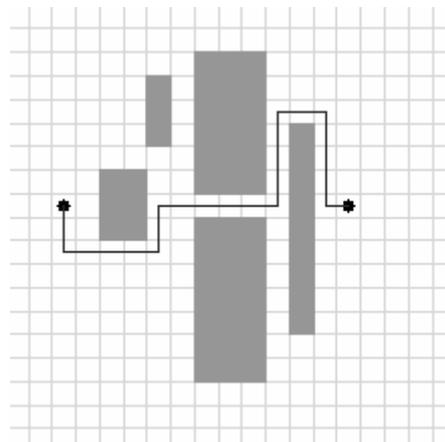


**input**

```
12 0
5
2 -1 3 1
6 -7 8 -1
6 1 8 6
4 3 4 5
10 -5 10 3
```

**output**

```
24
8
0 -2
4 0
0 2
5 0
0 4
2 0
0 -4
1 0
```



**Note:** each of the figures depicts the respective input, along with the supplied solution.