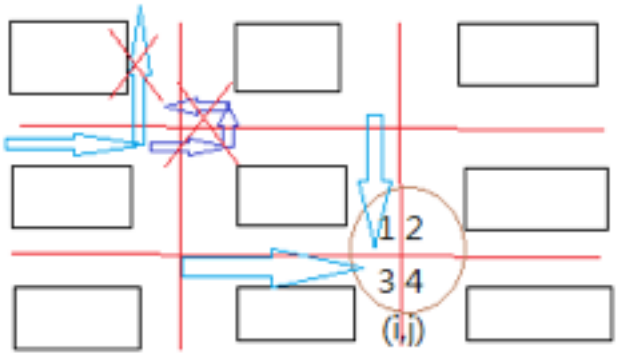
Momo's Final Travel

問題描述：

With many efforts, princess Momo finally got to Earth, where princess Lala lives in. However, Momo doesn't know how to get to Lala's house! Because the traffic on the Earth is very complicated! So Momo will take the taxi to her destination.

The road system is a 1000*1000 grid system. There are 1001 east-west(EW) roads and 1001 north-south(NS) roads. From west to east, each NS road is assigned a number from 0 to 1000, called NS0, NS1,…From south to north, each EW road is assigned a number from 0 to 1000, called EW0, EW1,…Intersection is defined by two road, for example, (1,2) is the intersection of NS1 and EW2. From one intersection(i,j) go to another intersection(i+1,j),(i-1,j),(i,j+1),(i,j-1) takes one minute. Each intersection would be partitioned into four region : up-left, up-right, down-left, down-right.

However, in this road system, there are three prohibitions. First, we can't make a left turn, because it's dangerous. For example, if we are now at (i,j), and go east to (i+1,j). Then from (i+1,j), we can't go to (i+1,j+1) directly, because we need a left turn in this intersection. Second, U-turn is also prohibited. In the above example, from (i,j) to (i+1,j), we can't make a U-turn and go back to (i,j) again or, from the region 3,4 of (i+1,j) and make a U-turn to reach region 1,2 of (i+1,j). See following picture as a clear view. Third, there are some intersections that are in maintenance, so we can't go through those intersections.



In this picture we can see that region 3 of intersection (i,j) can only come from (i-1,j) or (i,j+1).

See the following real example for a clear view:

We are at the down-right corner of (1,2), traveling east on EW 2.

(a) If we want to go to the down-right or down-left corner of (4,2), we simply continue traveling.

(b) If we want to go to the up-right corner of (4,2), a natural way is to go to the intersection of (4,2), and make a left turn. But we want to avoid left turn [since it is dangerous], and we can do so by making a right turn at NS 5, a right turn at EW 1, and another right turn at NS 4.

(c) If we want to go to the up-left corner of (4,2), a natural way is to go to the intersection of (4,2), and make a U-turn. But we also want to avoid U-turn [since it is even more dangerous than left turn]. So, we should make a right turn at NS 3, a right turn at EW 1, a right turn at NS 2, a right turn at EW 3, a right turn at NS 4.

Momo knows her intersection now (x1,y1) and the initial direction she would go, and she also knows the intersection of Lala's house (x2,y2) with region {up,down}-{left,right}. What's the minimal time to meet Lala?

輸入說明:

Input begins with an integer *T(1<=T<=25)*, the number of test case. Each test case would be in the following format.

Line 1 : x1 y1 dir : three numbers, Momo's intersection and initial direction.

Line 2 : x2 y2 region : three numbers, Lala's intersection and region

Line 3 : K(0<=K<=10000) : the number of intersection that are in maintenance.

Line 4~4+K-1 : x3 y3 : the position of intersection.

0<x1,y1,x2,y2,x3,y3<1000.

if dir = E, means the initial direction is east,
if dir = W, means the initial direction is west,
if dir = N, means the initial direction is north,
if dir = S, means the initial direction is south.

if region = 1, means up-left region,
if region = 2, means up-right region,
if region = 3, means down-left region,
if region = 4, means down-right region.

Input will guarantee that the start intersection would be different from end intersection. Also, start and end intersection won't be in maintenance.

輸出說明:

Each test case outputs one line, which is the shortest time Momo

need to spend to meet her sister Lala. If Momo can’t meet Lala then output “QQ”.

範例**:**

Sample Input:	**Sample Output:**
7 1 2 E 4 2 1 0 1 2 E 4 2 2 0 1 2 E 4 2 3 0 1 2 E 4 2 4 0 1 2 E 4 2 3 1 3 2 1 2 E 4 2 3 1 2 2 1 2 E 4 2 3 4 3 2 5 2 4 1 4 3	9 7 3 3 9 QQ QQ