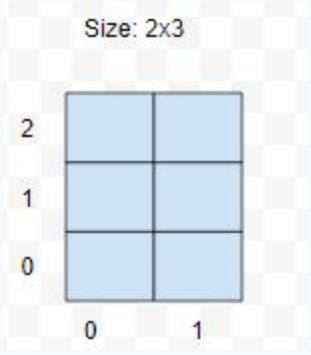
Solution's Author: Julio Yanes
Language: C#
Problem's Author: MyTeamAbroad
Order of solution: O(n) where n is the size of the command line.

Solution explanation

The first step is to think about the graphic solution:

For a test case in (x,y) dimensions where the dimension are (2,3) , we can see in the following image that the limits of the movements for x and y , so the rages are: -1<x<2 and -1<y<3 (x and y are integers).



With all the inputs we end with this statements:

We have the range of the movements
We have the orientation
We have the start point

Then we only have to test if it is a valid movement or not, returning false and exiting the program if not. For this there is no need of making a matrix, we only need a tuple (x,y) of the actual position and the orientation.

Depending of the orientation we will do this:

Add 1 to y if orientation is North
Subtract 1 to y if orientation is South
Add 1 to x if orientation is East
Subtract 1 to x if orientation is West

Depending of the command we will do this:

Add 1 to orientation if is R
Subtract 1 to orientation if is L

Using 1 for north, 2 for east, 3 for south and 4 for west. If the orientation is 5 we will change it to 1 and if it is 0 we will change it to 4.

Continuing the test case with this input:

2 3
0 0 N
ARA

In this case we start in the position (0,0) with a range of movement of -1<x<2 and -1<y<3, and North for the orientation. So we will change North to 1 and then we will process the commands.

For A we will add 1 to y , because the orientation is North. For R we will add 1 to the orientation, so it will change to 2 (that means East). Finally for A we will add 1 to x , because the orientation is East.

The output will be like this:

True,E,(1,1).

Code explanation

We start processing the input and checking for errors, for this we have 3 functions to check inputs errors. If there are inputs errors then the program will restart.

Then we will do the movement calculation, for that we loop the command line and we follow our logic explained in the solution explanation. We always check if the position is a valid one, if not then we will be returning false and exit the program. In other case we return the output.

List of functions

check Dimensions : Check the first line if it has negative numbers , characters or if it has more than 2 arguments.

checkCoorAndOrientation: Check in the second line that the coor is a positive integer and that the orientation is N, S , W or E.

checkCommandLine: Check the last line for char that are not A , R or L.

getOrientationToInt: Will get the char orientation to it int equivalent.

getOrientationToChar: Will get the int orientation to it char equivalent.

advance: Will be adding or subtracting to x or y depending on the orientation.

checkPosition: Will check if the position is a valid one.