Report for Project 3

1. Notable Obstacles:

I always find the increment of a variable really hard to be determined, so at first there’s some bugs because of that. However, I managed to trace the route segment step by step, and to take a note, then I felt much better cause I know better about the logic inside my function, thus making the use of increment easily.

2. Description of the Design of My Program:

**function isRouteWellFormed:**

find and turn the first character to lower case

if not ’n’/’s’/‘w’/‘e’,

return false

increment the position

find the next character

if is ’n’/’s’/‘w’/‘e’,

continue

else if no remaining character,

return true

else if not digit,

return false

else if is digit,

increment the position

if is digit,

increment the position

increment the position

no exceptions occur, return true

**function navigateSegment:**

if the position is not valid,

return -1

if the maxSteps is not positive,

return -1

set the direction to lower case

if the direction is ’n’,

r-1

else if is ’s’,

r+1

else if is ’w’,

c-1

else if is ’e’,

c+1

else if not ’n’/’s’/‘w’/‘e’,

return -1

no exceptions occur, return ns

**function navigateRoute:**

start a loop here

if the position is not valid,

return 2

if the route is not well formed,

return 2

set nsteps equals to 0

find the next character

if is ’n’/’s’/‘w’/‘e’ or equals to string size,

set ShouldBeSteps = 1

else if is digit,

increment the position, set ShouldBeSteps equals to the digit

if the next character is digit,

set ShouldBeSteps equals to ten times that digit add the former digit

increment the position

call the function navigateSegment to get the number of TrueSteps

add TrueSteps to the value of nsteps

if TrueSteps less than ShouldBeSteps,

return 3

if the direction is ’n’,

nr-1

else if is ’s’,

nr+1

else if is ’w’,

nc-1

else if is ’e’,

nc+1

the loop ends

if the final start position is the same as the end point,

return 0

else,

return 1

3. List of the Test Data

**function isRouteWellFormed:**

empty string, “”

start with non-letter, “12ew34n5”

start with letter but not n/s/w/e, “x33w3e4”

start with right letter but with more than two consecutive digits, “w34556”

start with right letter but with non-right-letter, “e34w1s4t”

perfect, “nN34e1s22”

**function navigateSegment:**

r and c start beyond the grid, navigateSegment(-1, 1, 'w', 1)

the original position has obstacles, navigateSegment(2, 2, 'e', 1)

the direction is non-letter, navigateSegment(1, 1, '1', 1)

the direction is letter but not n/s/w/e, navigateSegment(1, 1, 'x', 1)

maxSteps less than 0, navigateSegment(1, 1, 'e', -1)

get blocked during the process, navigateSegment(1, 1, 'e', 3)

not get blocked, but go beyond the grid, navigateSegment(1, 3, 's', 5)

perfect, navigateSegment(1, 1, 'e', 2)

**function navigateRoute:**

no problems occur, but end in a position different to expected one,

// navigateRoute(3,1, 3,3, "N2eE01n0s2e1", len)

invalid starting position, navigateRoute(3,2, 3,3, "N2eE01n0s2e1", len)

invalid ending position, navigateRoute(3,1, 3,2, "e1", len)

the route isn’t well formed, navigateRoute(3,2, 3,3, "N2eE01n0s2e1x", len)

get blocked during the process, navigateRoute(3,1, 3,3, "N1eE01n0s2e1", len)

perfect, navigateRoute(2,4, 2, 3, "w", len).