1)

I experienced a few notable obstacles. Being able to parse a string of characters and to correctly divide and separate route segments from each other was challenging. Thinking of an algorithm to find each route segment was difficult and required precise loops. I also had trouble with keeping the int& nsteps variable updated. I needed to be able to store different values into nsteps based on whether or not the robot made it to the end or not. This was tough, and I often got lost in my own nested loops; I was confused at first because I wrote nsteps = tempLength instead of nsteps += tempLength. This caused nsteps to continually change values instead of adding and storing each tempLength value into nsteps.

2)

isRouteWellFormed(string route)

returns true for empty strings

checks to make sure first char is a letter

repeatedly:

checks to see if it’s alphanumeric

checks to see if the letter is n,e,s,w

keeps number of digits that follows a direction under or equal to 2

returns true if all the requirements are met

int navigateSegment(int r, int c, char dir, int maxSteps)

checks for valid parameter inputs, and checks to see if robot is standing on a wall

determines the step by step movement of robot for each direction

repeatedly (4x):

robot takes individual steps, checks if it is wall, and records number of steps

returns number of steps taken if it hits a wall

returns maxSteps if it is under the total possible steps in a given direction

returns error if something other than n,e,s,w is put into the dir parameter

int navigateRoute(int sr, int sc, int er, int ec, string route, int& nsteps)

checks for bad input, and calls isRouteWellFormed to check string, returns 2 if bad input

repeatedly:

loops through each character in the string

finds each route segment by separately storing the letter and number into variables

if a letter is followed by another letter, length of route becomes 1 for the first letter

checks to see if robot hits boundary, returns error if it hits wall or goes out of bounds

adjusts row and col to locate robot’s position after every route segment

records total steps taken during entire trip, adds the steps taken into nsteps after every iteration

checks if final position of the robot is same as proposed location

returns 0 if robot successfully lands in the proposed end location

returns 1 if robot does not make it to proposed end location

3)

ASSUMING:

setSize(3, 4);

setWall(1, 4);

setWall(2, 2);

setWall(3, 2);

(also assuming that len is set to -999 before every navigateRoute test)

assert(!isRouteWellFormed(“1n2e1”));

Reason: checks to see if it catches strings that start with a number

Expected Results: I expect no abort message because well formed routes cannot start with a number

assert(!isRouteWellFormed(“n123es2”));

Reason: checks to see if it catches a route segment that has more than two digits following a letter

Expected Results: I expect no abort message because route segments can only have 2 digits

assert(!isRouteWellFormed(“n12xe5”));

Reason: checks to see if it catches a non-directional character

Expected Results: I expect no abort message because route segments can only have n,s,e,w letters

assert(navigateSegment(1, 1, ‘E’, 10) == 2);

Reason: checks to see if the code stops before it hits a set wall, and checks to see if it counts steps

Expected Results: I expect no abort message because my code checks for walls before it steps

assert(navigateSegment(2, 2, ‘W’, 2) == -1);

Reason: checks to see if initialized position is a wall, returns -1 if it starts at a wall

Expected Results: I expect no abort message because my code checks the initial position of the robot

assert(navigateRoute(2, 1, 2, 4, “Ne1E01n0se1”, len) == 0 && len == 5);

Reason: checks to see if navigates properly, if it ends in right place, and if steps were counted

Expected Results: I expect no abort message because robot navigates maze without hitting walls

assert(navigateRoute(2, 1, 2, 4, “Ne1E01n0se0”, len) == 1 && len == 4);

Reason: checks to see if returns 1 if robot doesn’t make it to expected end location

Expected Results: I expect no abort message, returns 1 because it never makes it to 2,4, it took 4 steps

assert(navigateRoute(2, 1, 3, 4, “NX1E01n0se0”, len) == 2 && len == -999);

Reason: checks the string to see if it is valid

Expected Results: I expect no abort message, returns 2 because the string is not syntactically valid

assert(navigateRoute(3, 1, 1, 1, “n03”, len) == 3 && len == 2);

Reason: checks to see if it knows when boundary is hit, and checks to see if steps were being counted

Expected Results: no abort message because robot hits a wall after taking 2 steps up

assert(navigateRoute(1, 1, 3, 4, “S2e3”, len) == 3 && len == 2);

Reason: checks to see if it knows when a wall is hit, and checks to see if steps were being counted

Expected Results: no abort message because the robot hits a wall after taking 2 steps, so returns 3

assert(navigateRoute(3, 4, 1, 1, "Nw1nw02", len) == 0 && len == 5);

Reason: checks to see if it completes a given route that goes around a wall and sees if it reaches end

Expected Results: no abort message because the robot successfully navigates the maze and reaches end