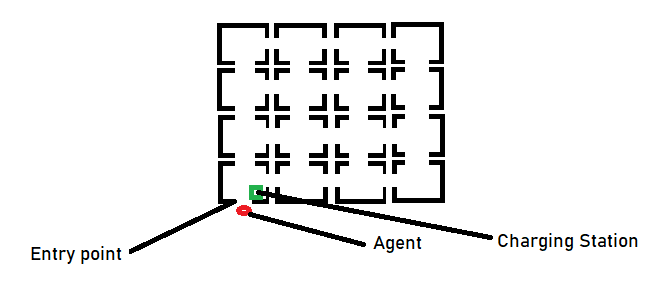
**Lab 2 Instructions**

Imagine a square grid of NxN rooms. Each room has doors leading to its neighboring rooms (NSEW neighbors only). One room at the edge opens to the outside world i.e. the entry point. This room also has a charging station (we will relax this condition later). One room has a pot of gold in it. An agent has to start from the entry point, explore the rooms, locate the pot of gold and bring it to the entry point. There is an additional constraint. The agent is powered by a battery. The battery (on full charge) has enough charge to visit X rooms where X < N2. So, suppose the agent is currently in the room (y, z) then it has to decide whether it should explore further or return back to the entry point to recharge itself. The following diagram depicts the situation for N = 4. Due to boundary conditions given at the end, N = 4 does not work in the present case. The minimum value of N is 5.



Write the agent program.

You will have to define the environment (the NxN) rooms along with the doors, entry point etc. It will help if you identify each room by its (x, y) location starting with (0, 0) for the entry point. Obviously, x and y are integer variables with 0 ≤ x, y < N. The inputs will be N, X and the location of the charging point (which does not have to be (0,0)) and the location of pot of gold (not known to the agent till it reaches the location).

You will have to define the suitable actions that the agent can take i.e. direction of movement (like forward, backward, turn right, turn left). The actual actions can be implemented using the following assignments: x = x + 1, x = x -1, y = y + 1, y = y – 1 where (x, y) is the current location. You will have to define the conditions under which the agent will perform these actions. For example, if x = N – 1 then we should not increment the value of x any more, without first decrementing it. The agent should store the current location in its memory. Some of the rules to be used are:

1. If the agent hits a right wall then the agent turns left (North)
2. If the agent hits a left wall then the agent turns right (North)
3. If the agent hits the upper wall then the agent turns takes a step back (South)
4. If the agent hits the lowerer wall then the agent turns takes a step forward (North)

You will have to update its memory (i.e. which rooms have been explored and found empty).

On output –

1. print the total time taken (= total number of rooms visited where some rooms may be visited more than once)
2. if the agent had to go to the charging station then the location of the room where that decision was taken

Assumptions:

1. The agent moves at a constant speed

2. The agent can sense the pot of gold when it reaches the room containing it

3. The agent can sense its current location

2. The time to recharge is zero. In case you do not like this then add a factor.

3. 4N < X < N2.

**Upload your work on the assignment link created on Moodle.**