ECE172A
Jiseok Jeong
A14281757

Homwork 1

Problem 1. MATLAB basics (5 points)

(i)

```
>> A

A =

     2    59     2     5
    41    11     0     4
    18     2     3     9
     6    23    27    10
     5     8     5     1

>> B

B =

     0     1     0     1
     0     1     1     1
     0     0     0     1
     1     1     0     1
     0     1     0     0
```

(ii)

```
>> C

C =

     0    59     0     5
     0    11     0     4
     0     0     0     9
     6    23     0    10
     0     8     0     0
```

(iii)

```
>> inner_product = dot(A(2,:),A(5,:));
>> inner_product

inner_product =

   297
```

(iv)

```
>> min_value

min_value =

     0

>> max_value

max_value =

    59
```

```
>> [min_i,min_j]

ans =

     1     1
     2     1
     3     1
     5     1
     3     2
     1     3
     2     3
     3     3
     4     3
     5     3
     5     4
```

```
>> [max_i,max_j]

ans =

     1     2
```

(v)

ECE172A
Jiseok Jeong
A14281757

```
D =

     0     0     0     0
     0   -48     0    -1
     0   -59     0     4
     6   -36     0     5
     0   -51     0    -5
```

(vi)

```
>> min_value_D

min_value_D =

   -59

>> max_value_D

max_value_D =

     6
```

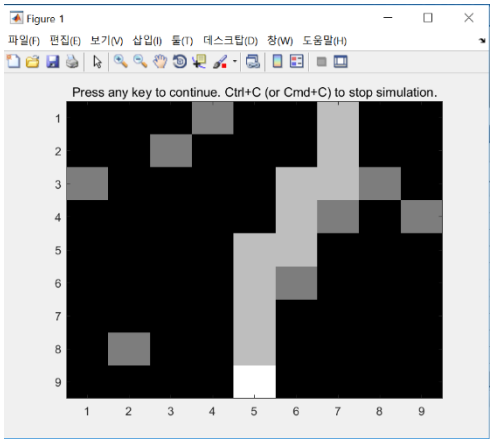
```
>> [min_i_D,min_j_D]

ans =

     3     2

>> [max_i_D,max_j_D]

ans =

     4     1
```

Problem 2. Robot traversal (15 points)

(i)

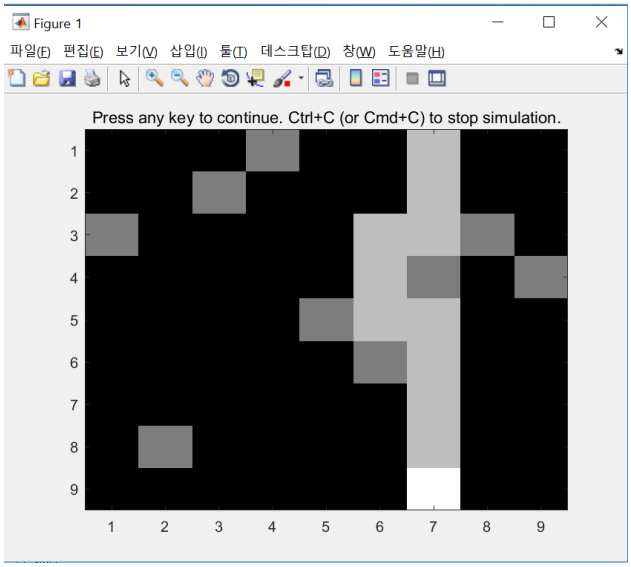
- ✓ **'loc' keeps track of the index of the path the robot goes through. So we can find the robot's present location and previous location as well.**
- ✓ **By adding [1,0], the robot moves towards south.**
- ✓ **The new object blocks the path of the robot. The robot cannot move to south because we did not implement an algorithm to avoid that kind of obstacles.**
- ✓ **It doesn't have the ability to solve novel problems. Whenever environments around it changes, we need to modify its system. So it would not be considered an intelligent system.**

(ii) This improved robot can detect the obstacle and act more rationally than the previous one does.

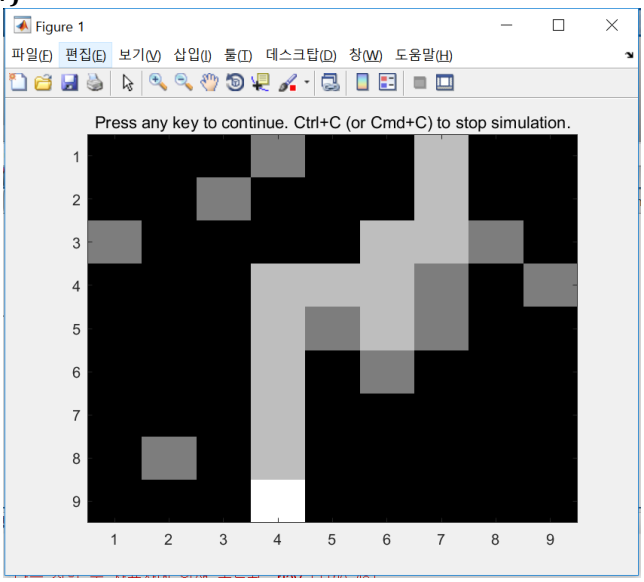


ECE172A
Jiseok Jeong
A14281757

(iii)



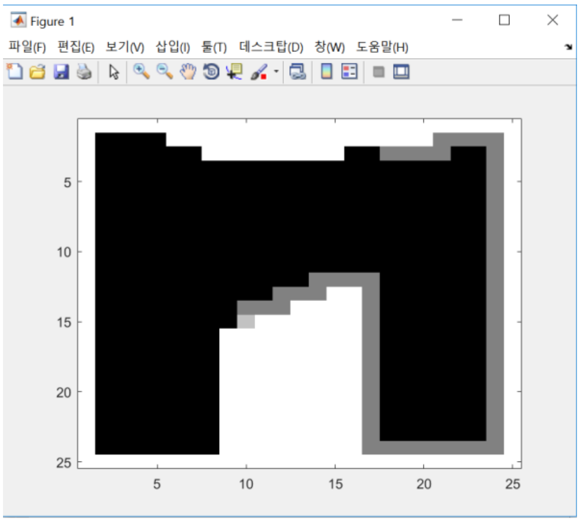
(iv)

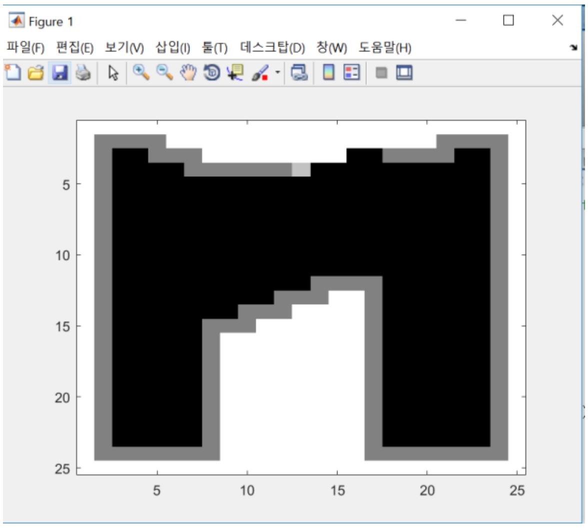


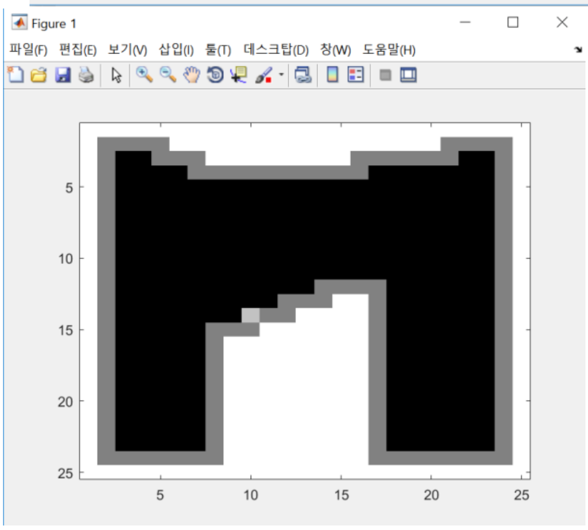
ECE172A
Jiseok Jeong
A14281757

Problem 3. Edge following (10 points)

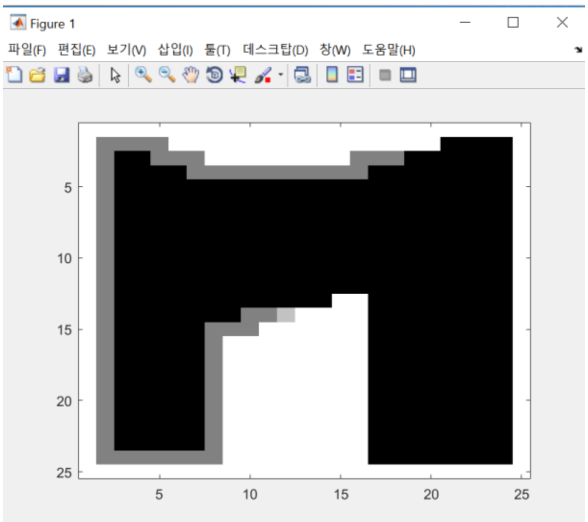
Clockwise

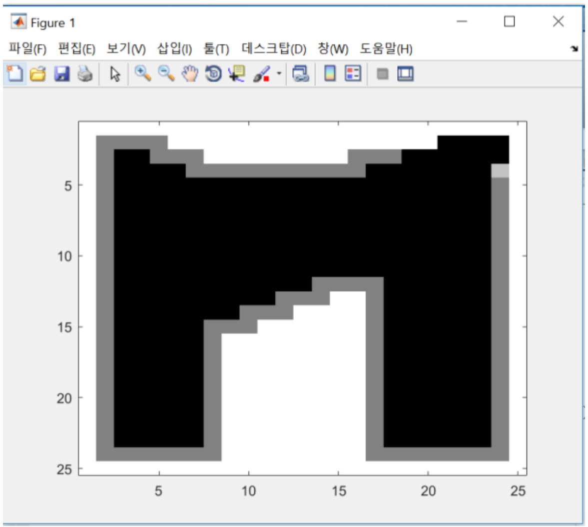




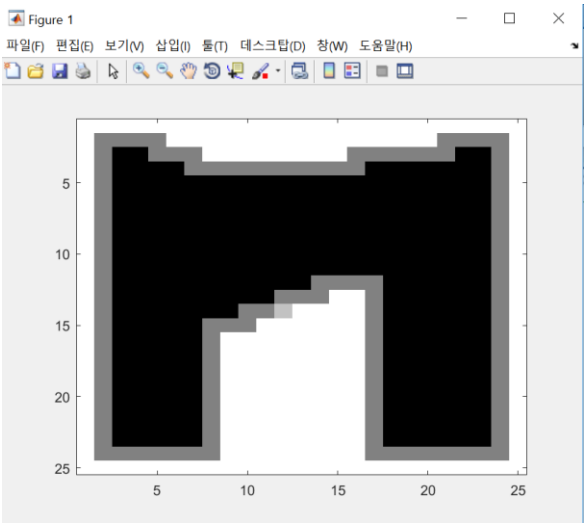


Anti-clockwise





ECE172A
Jiseok Jeong
A14281757



ECE172A
Jiseok Jeong
A14281757

Problem 4. Literature Survey(10 points)

Amazon fulfillment center robotics system(kiva robot)

That robot system is sensor-based (external) Robots and also co-operative (Human-Interactive) Robots. In this system, items are stored on portable storage units, 'Kiva robot' which are approximately 2 feet by 2.5 feet, and one foot high and capable of lifting 1000 pounds. When an order is entered into the Kiva database system, the software locates the closest Kiva robot to the item and directs it to retrieve it(planning). The robots navigate around the warehouse by following a series of computerized barcode stickers on the floor. Each Kiva robot has a sensor that prevents it from colliding with others by recognizing these barcode stickers and saving their present location. When the drive unit reaches the target location, it slides underneath the pod and lifts it off the ground through a corkscrew action(controllers). The robot then carries the pod to the specified human operator to pick the items, which makes that robot categorized as a human-interactive robot(co-coperative).

I think we need to extend the application of this kind of robot system in order to develop more efficient, fast and safe process which would finally be beneficial for consumers with lower price products. According to the statistics, it allowed amazon to store 50% more stuff in their warehouse and reduce human's error dramatically.

In the Amazon's warehouse system, there is also a computer vision system which detects how many and which kind of inventories are. Cameras get the images of their stocks and then recognize them by using deep learning algorithm that has been trained with a bunch of web information from Amazon web site. However, that system still has some challenges. There are always some possibilities of misrecognizing their inventories because of the incompleteness of their computer vision algorithm. Also, fatal errors on the system can happen. But the development of better image recognition algorithm with a lot of data set and better optimizing software can minimize these kinds of problems happening.

In making such robots, we might be worried that they will replace human's job. Yes, definitely they have already been replacing our job. But, at the same time, they have been reducing the rate of accidents in workplace by doing dangerous tasks instead of human. So, I think we would rather find a way of working and collaborating with these robots than just be against this trend because we always go forward in the direction to make the world advanced and cannot stop such advance.

ECE172A
Jiseok Jeong
A14281757

APPENDIX

Problem 1

```
A = [2 59 2 5; 41 11 0 4 ; 18 2 3 9 ; 6 23 27 10; 5 8 5 1];
B = [0 1 0 1 ; 0 1 1 1 ; 0 0 0 1 ; 1 1 0 1 ; 0 1 0 0];
C = A.*B; % Point-wise mutiply A with B and set it to C
inner_product = dot(A(2,:),A(5,:)); %inner product
min_value = min(C(:));
max_value = max(C(:));`
[min_i,min_j] = find(C == min_value); % to get indices
[max_i,max_j] = find(C == max_value); % to get indices
D = bsxfun(@minus,C,C(1,:)); % Subtracing the first row of C from all its row

% Repeating for D
min_value_D = min(D(:));
max_value_D = max(D(:));
[min_i_D,min_j_D] = find(D == min_value_D);
[max_i_D,max_j_D] = find(D == max_value_D);
```

Problem 2

```
(2)if detectObject(loc, obj, 'S')
        nextStep = loc(end,:) + [0 -1];
    end

(3)
if detectObject(loc, obj, 'S')
        if detectObject(loc, obj, 'W')
            nextStep = loc(end,:) + [0 1];
        else
            nextStep = loc(end,:) + [0 -1];
        end
     end


(4)
if detectObject(loc, obj, 'S') || haveIBeenHereBefore(loc, loc(end, :) + [1 0])
        if detectObject(loc, obj, 'W')
            if detectObject(loc, obj, 'E')
                nextStep = loc(end,:) + [-1 0];

            else
                nextStep = loc(end,:) + [0 1];
            end
        else
            nextStep = loc(end,:) + [0 -1];
        end
```

```
    end
```

Problem 3

```
function [ newPos ] = get_new_pos( curPos, sensorInput, dir )
% sensorInput is a 3x3 matrix and follows the following convention.
% sensorInput = [     a     b     c     ;
%                           d     x     e     ;
%                           f     g     h     ];
% 'x' is the location of the bot.
% a to h are either '0' or '1' depending on the presence of obstacle in
% those blocks. For example take the image shown in Q2. At this location
% sensorInput = [1 1 1;
%                    0 0 0;
%                    0 0 0];
    newPos = curPos;
    if dir == 0
        if sensorInput(1,2) == 1
            if sensorInput(2,3) == 1
                newPos(1) = newPos(1) + 1;
            else
            newPos(2) = newPos(2) + 1; % Move Right
            end
        elseif sensorInput(2,3) == 1
                if sensorInput(3,2) == 1
                    newPos(2) = newPos(2) - 1;
                else
                    newPos(1) = newPos(1) + 1;
                end

        elseif sensorInput(3,2) == 1
                if sensorInput(2,1) == 1
                    newPos(1) = newPos(1) - 1;
                else
                    newPos(2) = newPos(2) - 1;
                end

        elseif sensorInput(2,1) == 1
            newPos(1) = newPos(1) - 1;
        else
                if sensorInput(1,1) == 1
                newPos(1) = newPos(1) - 1;
                elseif sensorInput(3,1) == 1
                    newPos(2) = newPos(2) - 1;
                    elseif sensorInput(3,3) == 1
                        newPos(1) = newPos(1) + 1;
                else newPos(2) = newPos(2) + 1;
```

```
                    end
            end

        elseif dir == 1
            if sensorInput(1,2) == 1
                    if sensorInput(2,1) == 1
                        newPos(1) = newPos(1) + 1;
                    else
                    newPos(2) = newPos(2) - 1; % Move Right
                    end
            elseif sensorInput(2,1) == 1
                        if sensorInput(3,2) == 1
                            newPos(2) = newPos(2) + 1;
                        else
                            newPos(1) = newPos(1) + 1;
                        end

            elseif sensorInput(3,2) == 1
                        if sensorInput(2,3) == 1
                            newPos(1) = newPos(1) - 1;
                        else
                            newPos(2) = newPos(2) + 1;
                        end

            elseif sensorInput(2,3) == 1
                    newPos(1) = newPos(1) - 1;
            else
                        if sensorInput(1,1) == 1
                        newPos(2) = newPos(2) - 1;
                        elseif sensorInput(3,1) == 1
                            newPos(1) = newPos(1) + 1;
                            elseif sensorInput(3,3) == 1
                                newPos(2) = newPos(2) + 1;
                        else newPos(1) = newPos(1) - 1;
                        end
            end
end
```