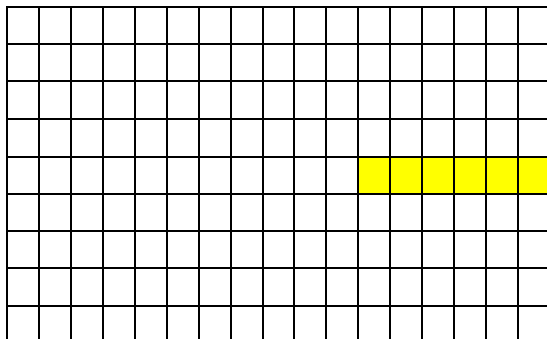


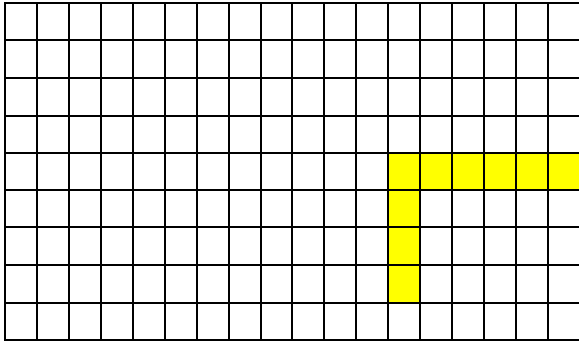
Atypon assignment report

Karel the robot

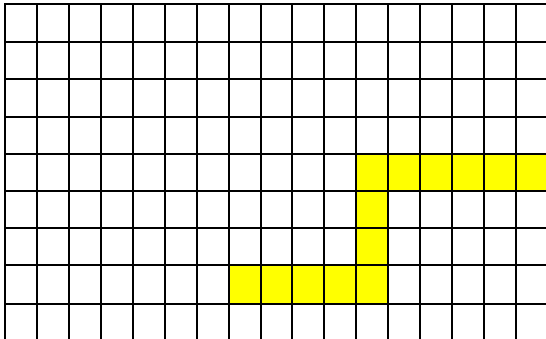
By Rayyan al Hourani



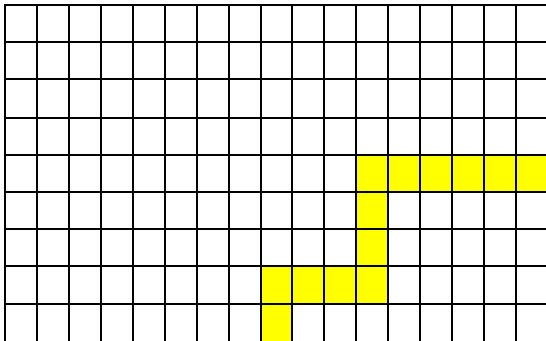
6- then the robot will draw the lower middle of the right side.



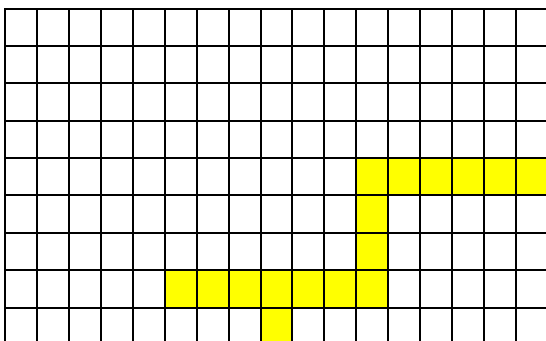
7- then the robot will draw the right middle of lower side



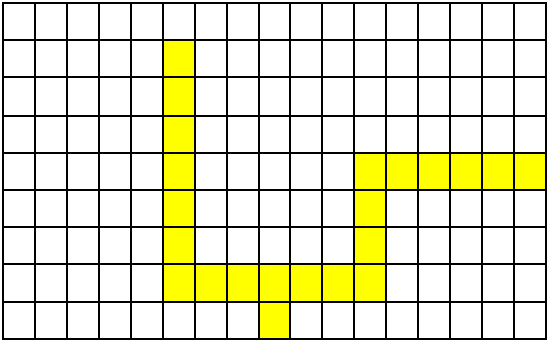
8- then the robot will draw the beepers that sticks to longest side.



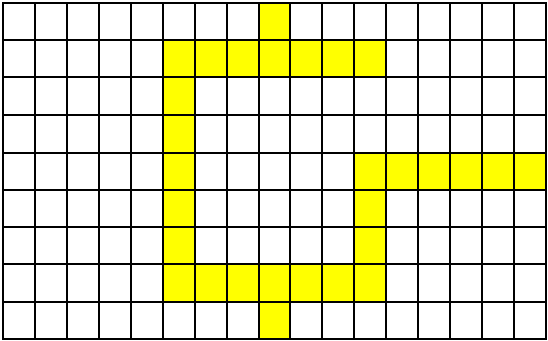
9- then the robot will draw the rest of lower side.



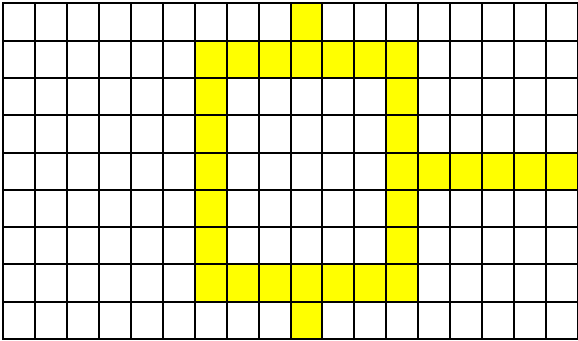
10- the the robot will draw the left side



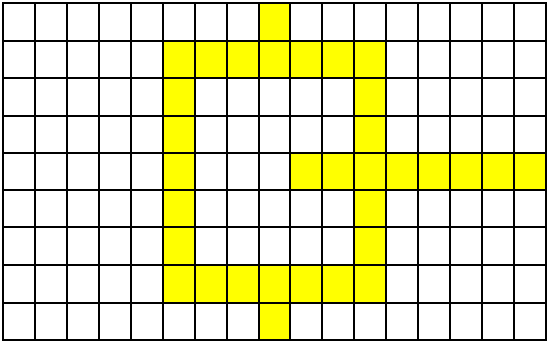
11- the the robot what he does in the lower side



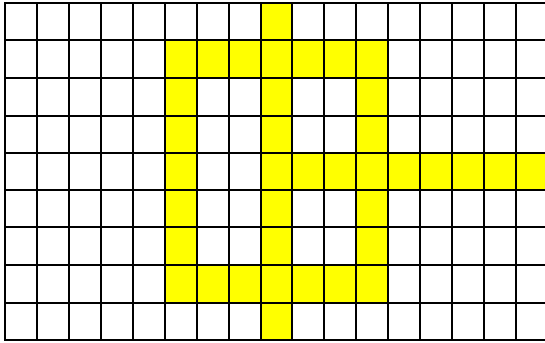
12- the draw the rest of right side



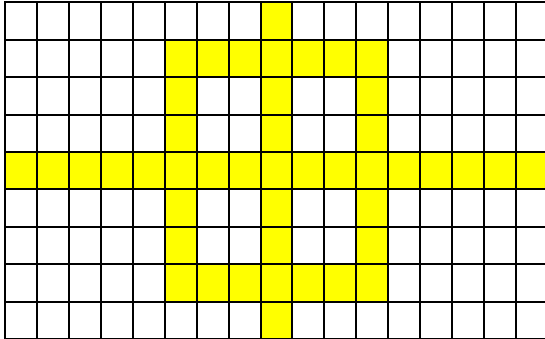
13- the draw the left middle of horizontal line



14- the draw the left middle of vertical line



15- the draw the rest of horizontal line



- The worst case is when the vertical line is just 1 line , so the robot will walk on it 2 times
- If the width or length is even, we draw double lines but horizontal line and vertical line different about how the robot will draw the double line but it's the same result.
- If the side length of inner chamber is odd, we maybe need some more moves than the even chambers side, and the Way of move to draw double line will be changed but the same results
- I will show all that difference in video and way of moves in video

The functions that are use :

- 1- **CountlengthNwidth()** : to count the length and width and assign the longest value and shortest value to 2 variables to apply the same algorithm if the height > width. And it used to calculate the inner chamber side length. And also reset a steps so that you can calculate the steps for many maps
- 2- **MoveToStartPoint()** : make robot go from top right and go to place that it has start form it, if the width > length he turns around and go to middle of left side , if width > length it turns left and go to middle of the top side.
- 3- **StartLine(int x)** : draw the start line that I mention it on the solution section, draw double or single line that depends if the side is even or odd.
- 4- **drawDoubleLine(int x)**: the robot draws double line that starts from point the go down the left the top and so on until it finish draw the line.
- 5- **drawDoubleLineFromDown (int x)**: the same of the previous function but it starts from down the go up then left then down and so on until it finishes draw the line.
- 6- **movesWithBeepers(int x)**: move many steps and put beeper in each step.

- 7- **moves(int x):** move many steps. And count the steps that he walks them.
- 8- **DrawSmallPart()** : draw the beepers that they stick to the longest side , its usually one or two beepers
- 9- **DrowChamberOutLine():** start draw the lower middle of right side then the lower side with the small part that have a functions above then the left side then draw a upper side with the small part then draw rest of right side.
- 10- **DrawUpperOrLowerLine()** : because the lower and upper side is the same , so that function draw them and call **DrawSmallPart()** when the robot reaches the middle of line
- 11- **DrawToCenterPoint()** : draw the right middle of horizontal line.
- 12- **PutAndReturn(int x):** draw a line then back to same point , I use it draw a vertical line when the width is odd.
- 13- **DrowVirticalLine()** : draw the vertical line if the width is odd I call the **PutAndReturn(int x)** function 2 times but of it is even there are 2 cases if the inner chamber side length is even the robot start draw from middle of line to top of the line then move to right then go to bottom of line then draw the rest of line to middle of line, else if the inner chamber side length is odd , its do the same thing but it its start draw to bottom.
- 14- **DrawHorizontalLine():** First the robot will move to start point and of the length is odd it will draw just 1 line using **movesWithBeepers(int x)** else if the length is even it will draw the rest of line using **drawDoubleLineFromDown()** and **movesWithBeepers()**.

Note : when I say the length and width I mean that always the width is bigger than length because that I assumed that I flip the map if the length > width.