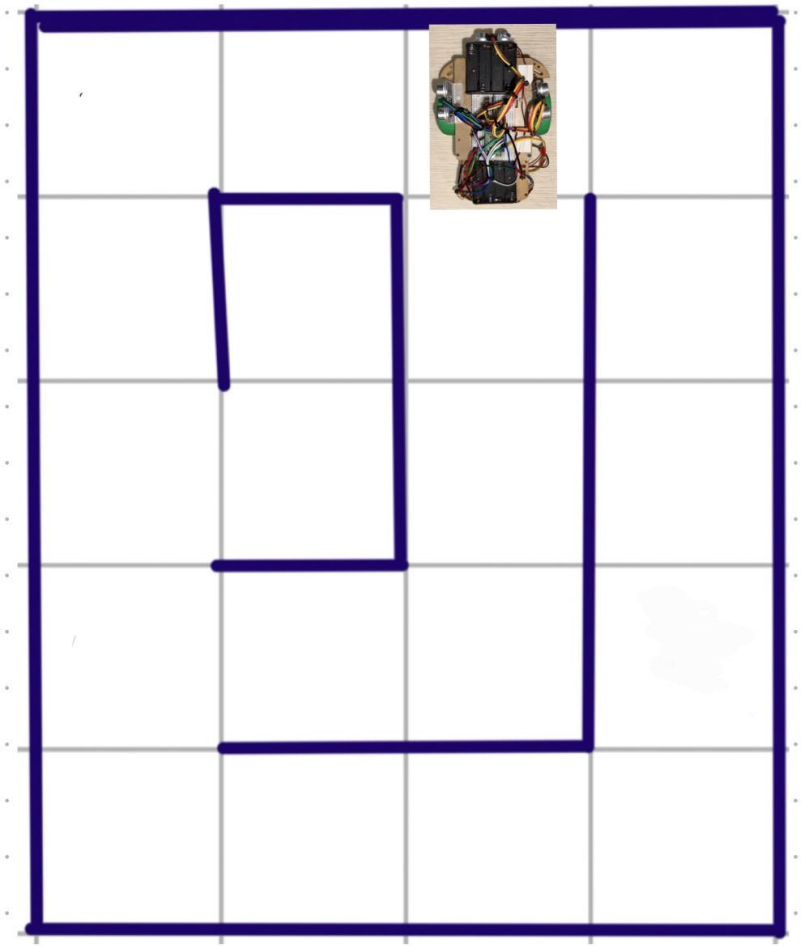
Mapping & Navigation

Ng Geok Ting, Jonalyn Koh, Loo Xian Hui

Table of Contents

01
Flowchart

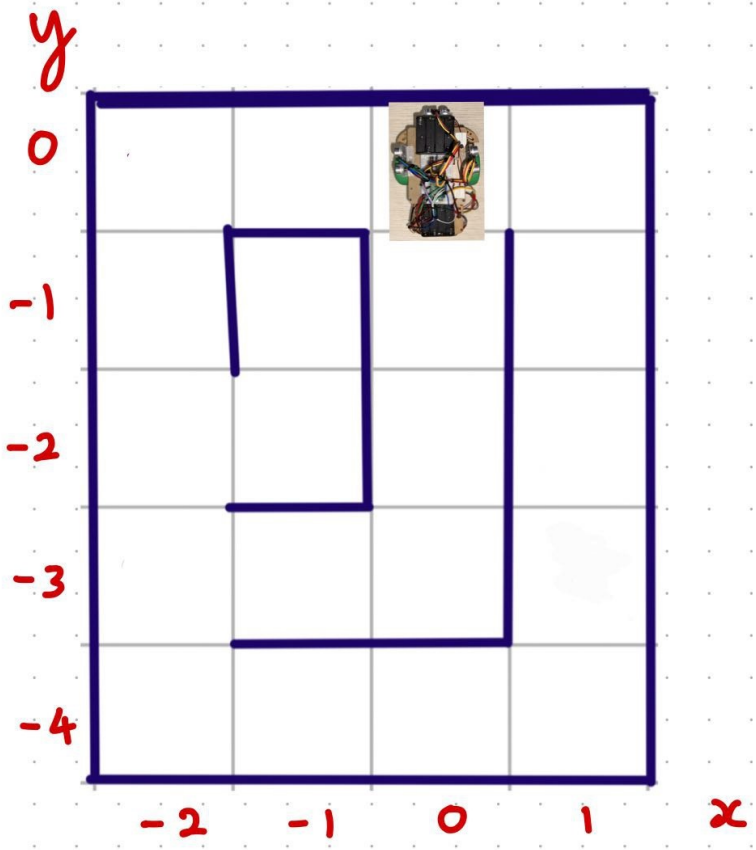
02
Mapping



short int direction = 1;
short int isExit[2];
short int numberOfNodes = -1;
short int numberOfMoves = -1;
short int currentXCoord = 0;
short int currentYCoord = 0;
short int debtCounter = -1;

nodeArray =
debtArray =
pathArray =
openingCounter = 0
debt_coord =

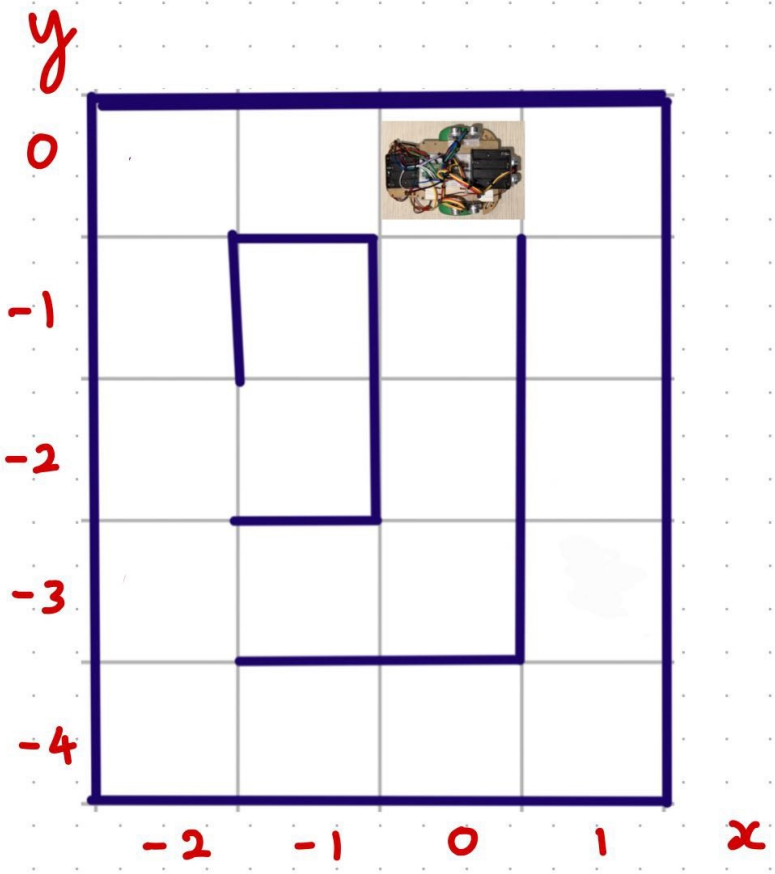
flag = 1



short int direction = 1;
short int isExit[2];
short int numberOfNodes = -1;
short int numberOfMoves =-1;
short int currentXCoord = 0;
short int currentYCoord = 0;
short int debtCounter = -1;

nodeArray =
debtArray =
pathArray =
openingCounter = 0
debt_coord =

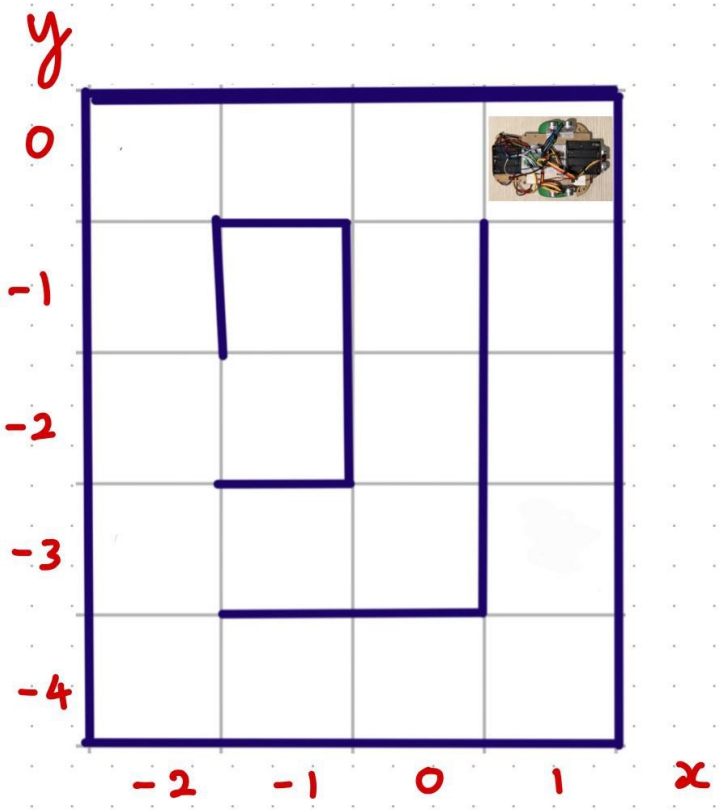
flag = 1



short int direction = 2;
short int isExit[2];
short int numberOfNodes = 0;
short int numberOfMoves = 0;
short int currentXCoord = 0;
short int currentYCoord = 0;
short int debtCounter = 1;

nodeArray = [0,0, 0010]
debtArray = [0,0,3] , [0,0,0]
pathArray = [0,0]
openingCounter = 0
debt_coord =

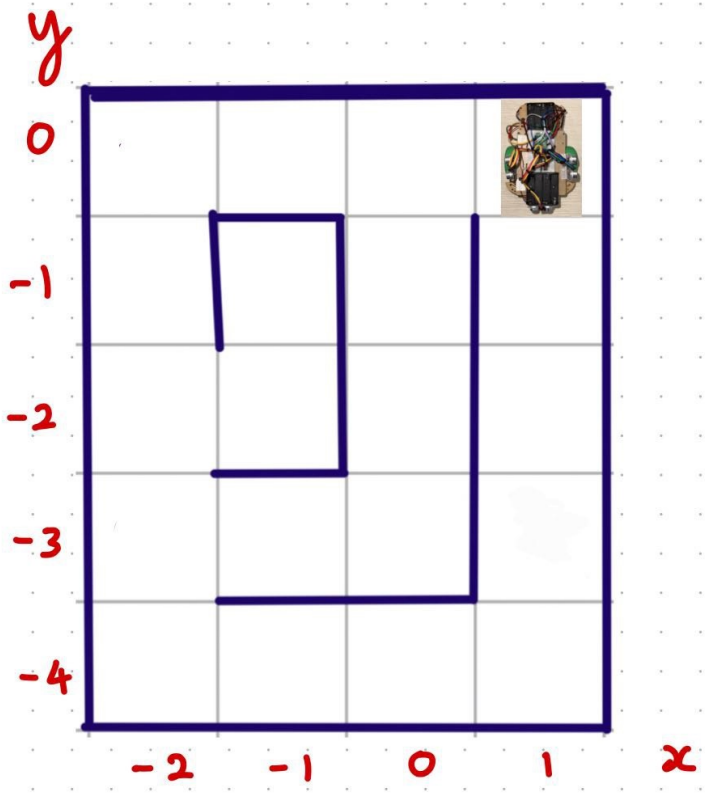
flag = 1



short int direction = 2;
short int isExit[2];
short int numberOfNodes = 1;
short int numberOfMoves = 0;
short int currentXCoord = 1;
short int currentYCoord = 0;
short int debtCounter = 1;

nodeArray = [0,0, 0010] , [1,0, 0110]
debtArray = [0,0,3] , [0,0,0]
pathArray = [0,0]
openingCounter = 0
debt_coord =

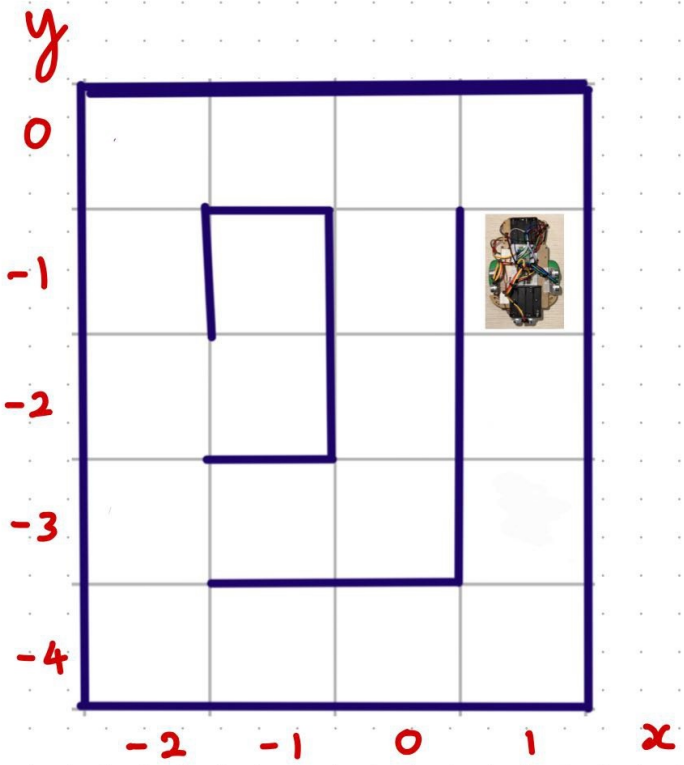
flag = 1



short int direction = 3;
short int isExit[2];
short int numberOfNodes = 1;
short int numberOfMoves = 1;
short int currentXCoord = 1;
short int currentYCoord = 0;
short int debtCounter = 1;

nodeArray = [0,0, 0010] , [1,0, 0110]
debtArray = [0,0,3] , [0,0,0]
pathArray = [0,0] , [1,0]
openingCounter = 0
debt_coord =

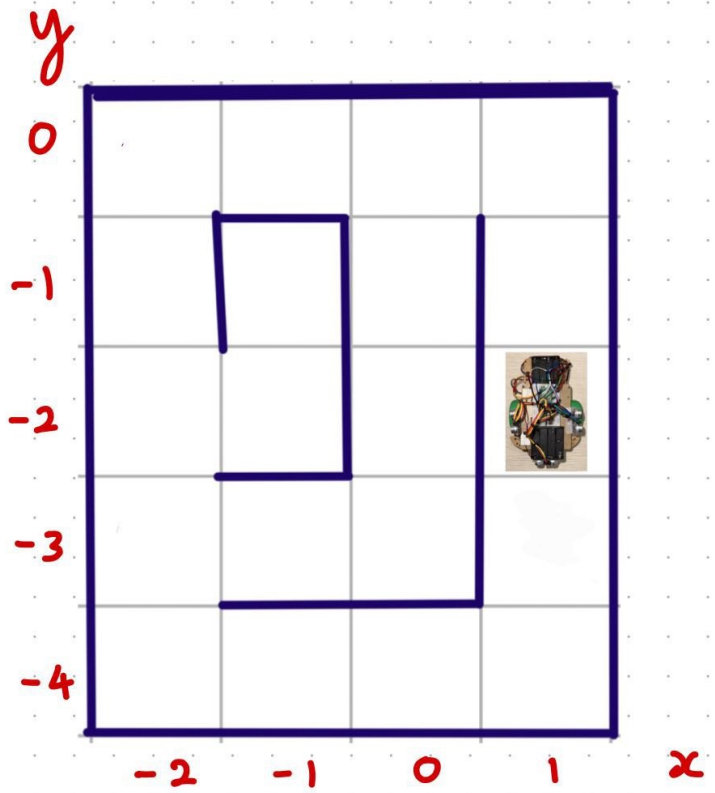
flag = 1



short int direction = 3;
short int isExit[2];
short int numberOfNodes = 2;
short int numberOfMoves = 2;
short int currentXCoord = 1;
short int currentYCoord = -1;
short int debtCounter = 1;

nodeArray = [0,0, 0010] , [1,0, 0110] , [1,-1, 0101]
debtArray = [0,0,3] , [0,0,0]
pathArray = [0,0] , [1,0] , [1,-1]
openingCounter = 0
debt_coord =

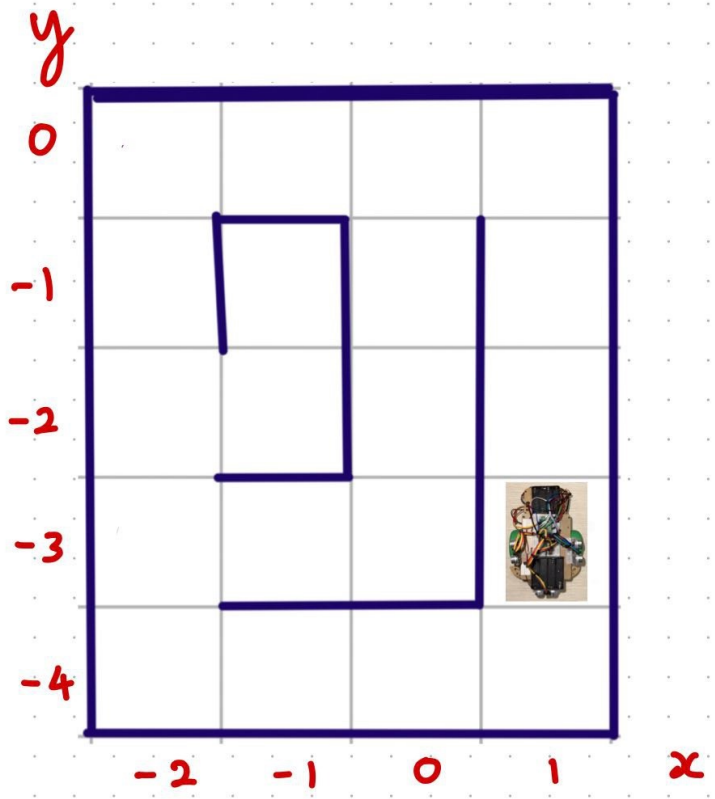
flag = 1



short int direction = 3;
short int isExit[2];
short int numberOfNodes = 3;
short int numberOfMoves = 3;
short int currentXCoord = 1;
short int currentYCoord = -2;
short int debtCounter = 1;

nodeArray = [0,0, 0010] , [1,0, 0110] , [1,-1, 0101] , [1,-2, 0101]
debtArray = [0,0,3] , [0,0,0]
pathArray = [0,0] , [1,0] , [1,-1] , [1,-2]
openingCounter = 0
debt_coord =

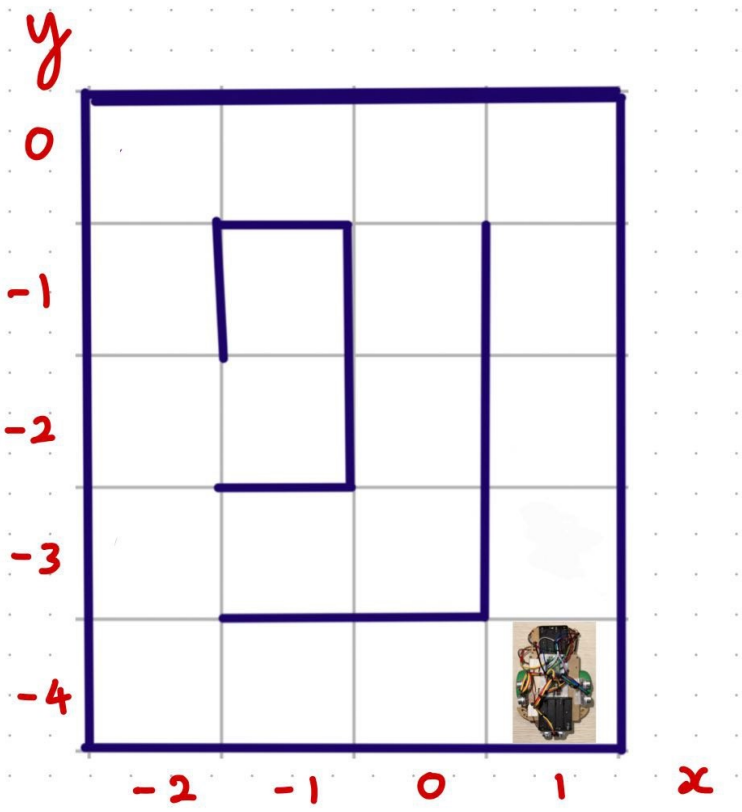
flag = 1



short int direction = 3;
short int isExit[2];
short int numberOfNodes = 4;
short int numberOfMoves = 4;
short int currentXCoord = 1;
short int currentYCoord = -3;
short int debtCounter = 1;

nodeArray = [0,0, 0010] , [1,0, 0110] , [1,-1, 0101] , [1,-2, 0101] , [1,-3, 0101]
debtArray = [0,0,3] , [0,0,0]
pathArray = [0,0] , [1,0] , [1,-1] , [1,-2] , [1,-3]
openingCounter = 0
debt_coord =

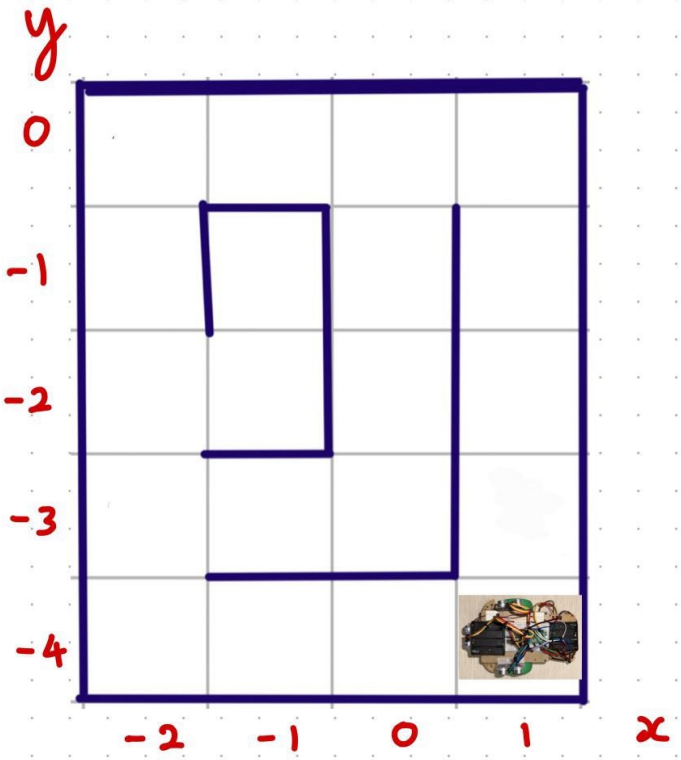
flag = 1



short int direction = 3;
short int isExit[2];
short int numberOfNodes = 5;
short int numberOfMoves = 4;
short int currentXCoord = 1;
short int currentYCoord = -4;
short int debtCounter = 1;

nodeArray = [0,0, 0010] , [1,0, 0110] , [1,-1, 0101] , [1,-2, 0101] , [1,-3, 0101] , [1,-4, 1100] ,
debtArray = [0,0,3] , [0,0,0]
pathArray = [0,0] , [1,0] , [1,-1] , [1,-2] , [1,-3]
openingCounter = 0
debt_coord =

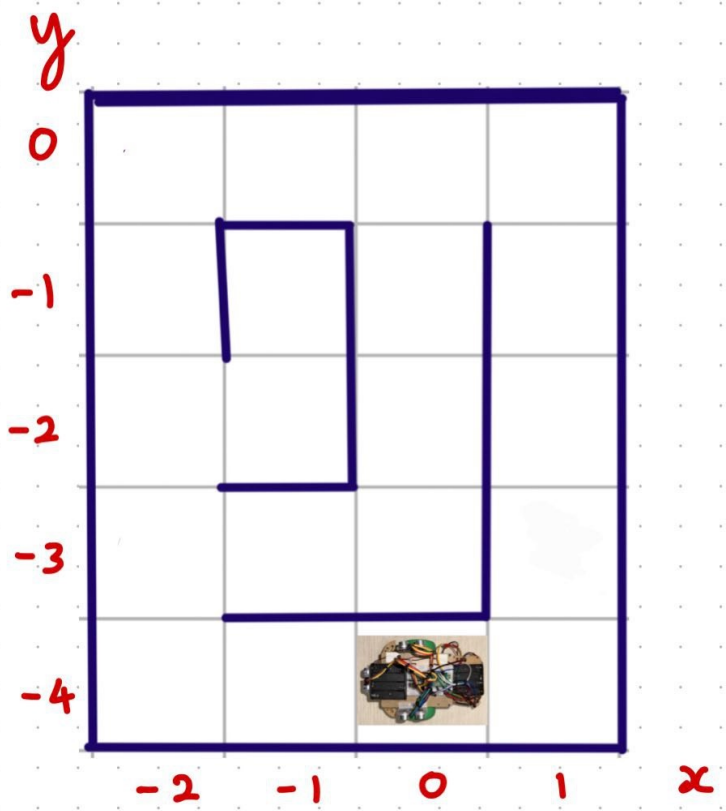
flag = 1



short int direction = 0;
short int isExit[2];
short int numberOfNodes = 5;
short int numberOfMoves = 5;
short int currentXCoord = 1;
short int currentYCoord = -4;
short int debtCounter = 1;

nodeArray = [0,0, 0010] , [1,0, 0110] , [1,-1, 0101] , [1,-2, 0101] , [1,-3, 0101] , [1,-4, 1100] ,
debtArray = [0,0,3] , [0,0,0]
pathArray = [0,0] , [1,0] , [1,-1] , [1,-2] , [1,-3] , [1,-4]
openingCounter = 0
debt_coord =

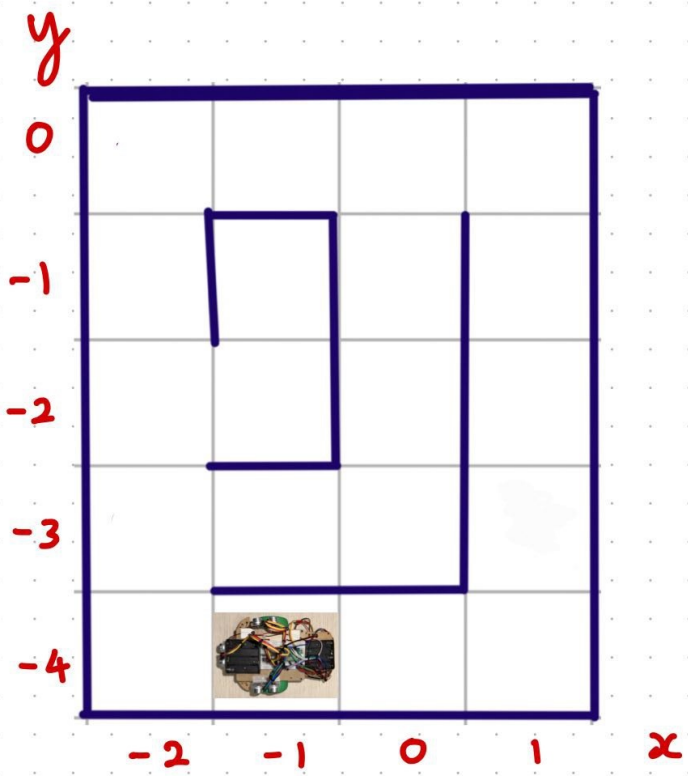
flag = 1



short int direction = 0;
short int isExit[2];
short int numberOfNodes = 6;
short int numberOfMoves = 6;
short int currentXCoord = 0;
short int currentYCoord = -4;
short int debtCounter = 1;

nodeArray = [0,0, 0010] , [1,0, 0110] , [1,-1, 0101] , [1,-2, 0101] , [1,-3, 0101] , [1,-4, 1100] , [0,-4, 1010]
debtArray = [0,0,3] , [0,0,0]
pathArray = [0,0] , [1,0] , [1,-1] , [1,-2] , [1,-3] , [1,-4] , [0,-4]
openingCounter = 0
debt_coord =

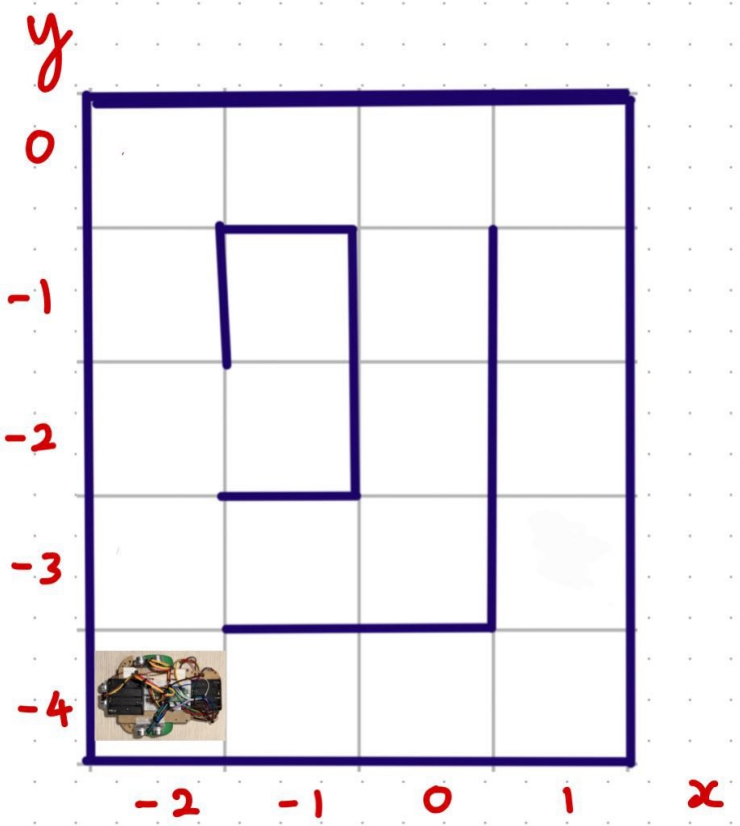
flag = 1



short int direction = 0;
short int isExit[2];
short int numberOfNodes = 7;
short int numberOfMoves = 7;
short int currentXCoord = -1;
short int currentYCoord = -4;
short int debtCounter = 1;

nodeArray = [0,0, 0010] , [1,0, 0110] , [1,-1, 0101] , [1,-2, 0101] , [1,-3, 0101] , [1,-4, 1100] , [0,-4, 1010] , [-1,-4, 1010]
debtArray = [0,0,3] , [0,0,0]
pathArray = [0,0] , [1,0] , [1,-1] , [1,-2] , [1,-3] , [1,-4] , [0,-4], [-1,-4]
openingCounter = 0
debt_coord =

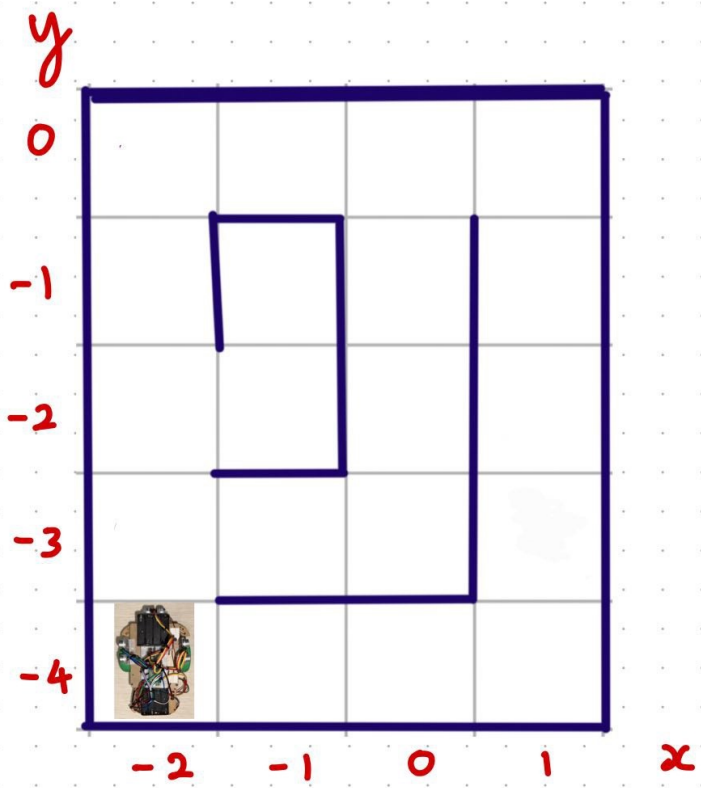
flag = 1



short int direction = 0;
short int isExit[2];
short int numberOfNodes = 8;
short int numberOfMoves = 7;
short int currentXCoord = -2;
short int currentYCoord = -4;
short int debtCounter = 1;

nodeArray = [0,0, 0010] , [1,0, 0110] , [1,-1, 0101] , [1,-2, 0101] , [1,-3, 0101] , [1,-4, 1100] , [0,-4, 1010] , [-1,-4, 1010] , [-2,-4, 1001]
debtArray = [0,0,3] , [0,0,0]
pathArray = [0,0] , [1,0] , [1,-1] , [1,-2] , [1,-3] , [1,-4] , [0,-4], [-1,-4]
openingCounter = 0
debt_coord =

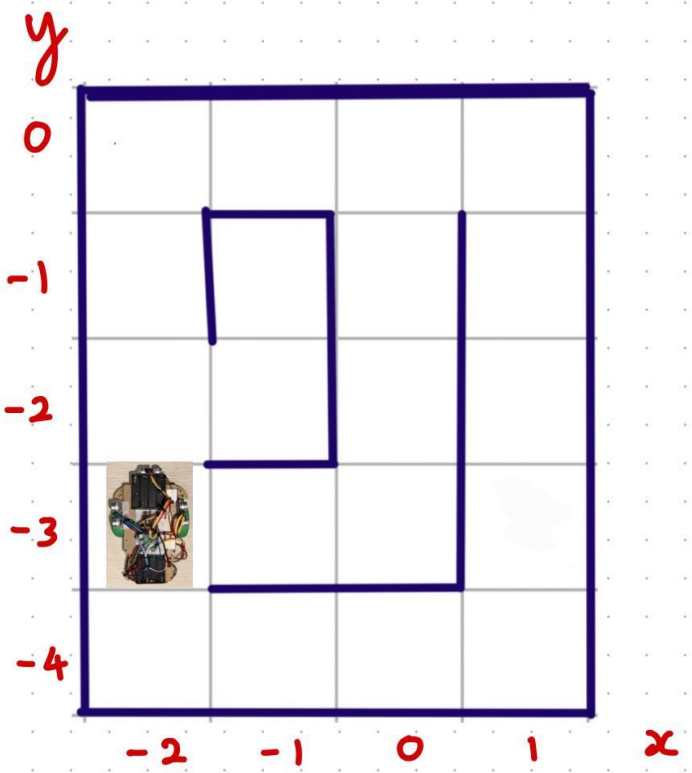
flag = 1



short int direction = 1;
short int isExit[2];
short int numberOfNodes = 8;
short int numberOfMoves = 8;
short int currentXCoord = -2;
short int currentYCoord = -4;
short int debtCounter = 1;

nodeArray = [0,0, 0010] , [1,0, 0110] , [1,-1, 0101] , [1,-2, 0101] , [1,-3, 0101] , [1,-4, 1100] , [0,-4, 1010] , [-1,-4, 1010] , [-2,-4, 1001]
debtArray = [0,0,3] , [0,0,0]
pathArray = [0,0] , [1,0] , [1,-1] , [1,-2] , [1,-3] , [1,-4] , [0,-4], [-1,-4] , [-2,-4]
openingCounter = 0
debt_coord =

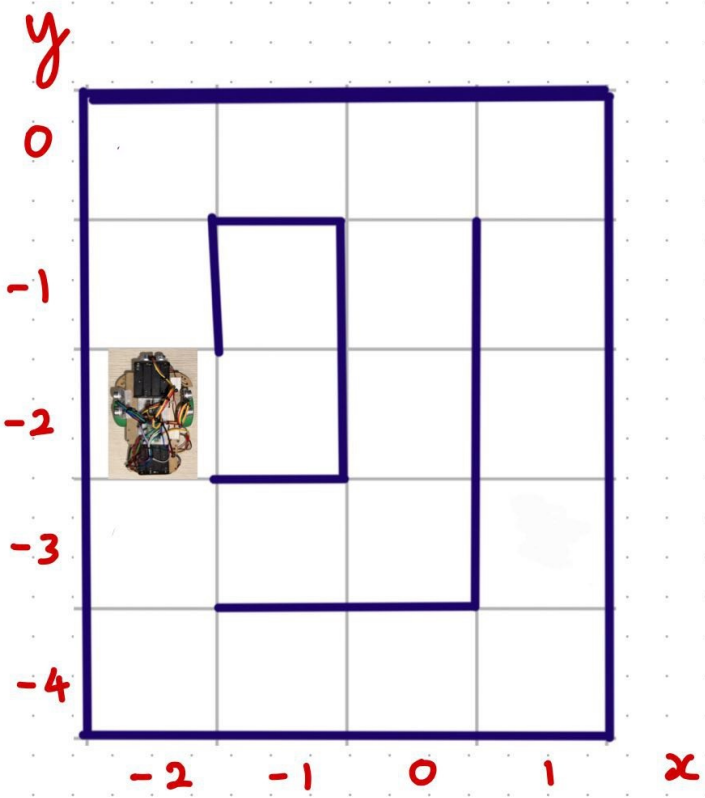
flag = 1



short int direction = 1;
short int isExit[2];
short int numberOfNodes = 9;
short int numberOfMoves = 9;
short int currentXCoord = -2;
short int currentYCoord = -3;
short int debtCounter = 2;

nodeArray = [0,0, 0010] , [1,0, 0110] , [1,-1, 0101] , [1,-2, 0101] , [1,-3, 0101] , [1,-4, 1100] , [0,-4, 1010] , [-1,-4, 1010] , [-2,-4, 1001] , [-2, -3, 0001]
debtArray = [0,0,3] , [0,0,0] , [-2,-3,2]
pathArray = [0,0] , [1,0] , [1,-1] , [1,-2] , [1,-3] , [1,-4] , [0,-4], [-1,-4] , [-2,-4], [-2,-3]
openingCounter = 2
debt_coord =

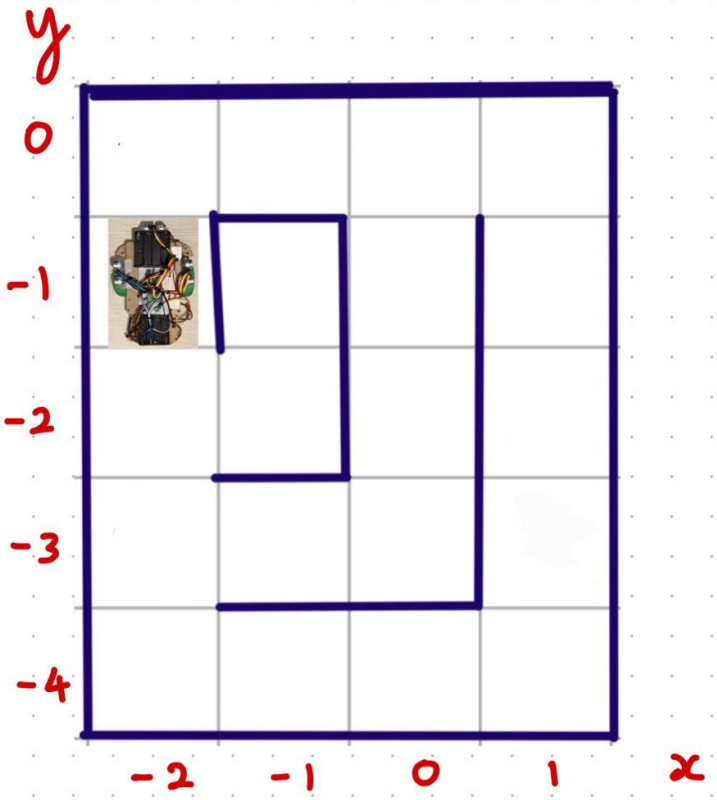
flag = 1



short int direction = 1;
short int isExit[2];
short int numberOfNodes = 10;
short int numberOfMoves = 10;
short int currentXCoord = -2;
short int currentYCoord = -2;
short int debtCounter = 3;

nodeArray = [0,0, 0010] , [1,0, 0110] , [1,-1, 0101] , [1,-2, 0101] , [1,-3, 0101] , [1,-4, 1100] , [0,-4, 1010] , [-1,-4, 1010] , [-2,-4, 1001] , [-2,-3, 0001] , [-2,-2,0001] ,
debtArray = [0,0,3] , [0,0,0] , [-2,-3,2] , [-2,-2,2]
pathArray = [0,0] , [1,0] , [1,-1] , [1,-2] , [1,-3] , [1,-4] , [0,-4], [-1,-4] , [-2,-4] , [-2,-3] , [-2,-2]
openingCounter = 2
debt_coord =

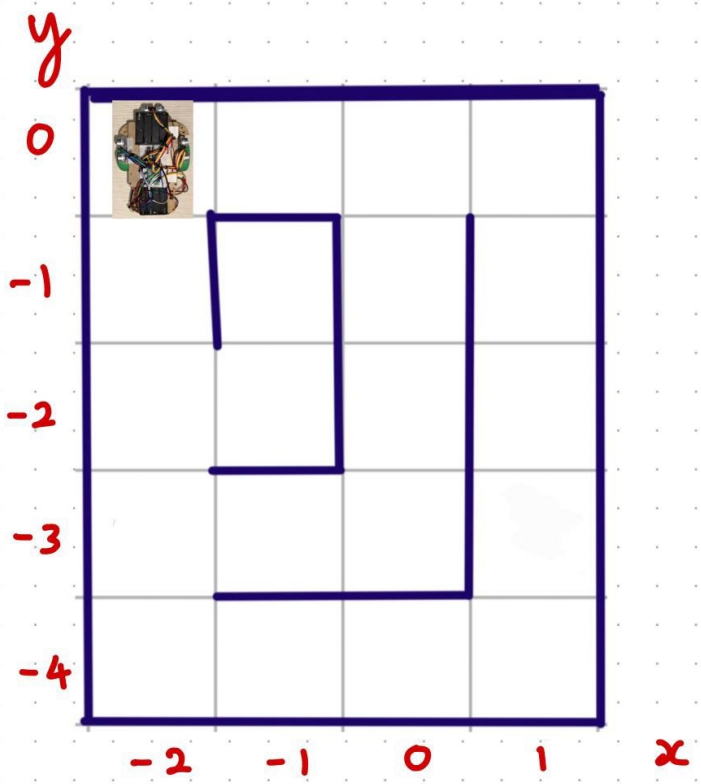
flag = 1



short int direction = 1;
short int isExit[2];
short int numberOfNodes = 10;
short int numberOfMoves = 10;
short int currentXCoord = -2;
short int currentYCoord = -2;
short int debtCounter = 3;

nodeArray = [0,0, 0010] , [1,0, 0110] , [1,-1, 0101] , [1,-2, 0101] , [1,-3, 0101] , [1,-4, 1100] , [0,-4, 1010] , [-1,-4, 1010] , [-2,-4, 1001] , [-2,-3, 0001] , [-2,-2,0001] ,
debtArray = [0,0,3] , [0,0,0] , [-2,-3,2] , [-2,-2,2]
pathArray = [0,0] , [1,0] , [1,-1] , [1,-2] , [1,-3] , [1,-4] , [0,-4], [-1,-4] , [-2,-4] , [-2,-3] , [-2,-2]
openingCounter = 2
debt_coord =

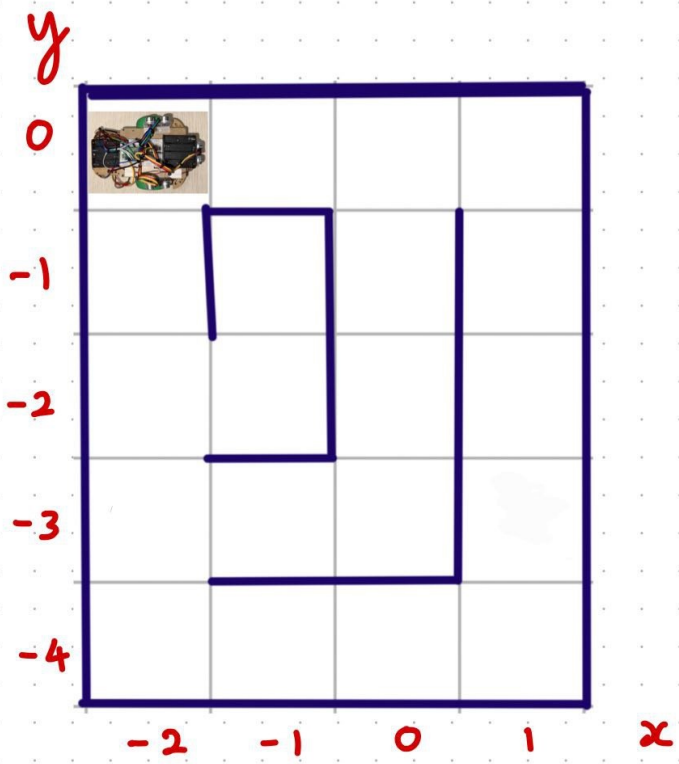
flag = 1



short int direction = 1;
short int isExit[2];
short int numberOfNodes = 12;
short int numberOfMoves = 11;
short int currentXCoord = -2;
short int currentYCoord = 0;
short int debtCounter = 3;

nodeArray = [0,0, 0010] , [1,0, 0110] , [1,-1, 0101] , [1,-2, 0101] , [1,-3, 0101] , [1,-4, 1100] , [0,-4, 1010] , [-1,-4, 1010] , [-2,-4, 1001] , [-2,-3, 0001] , [-2,-2,0001] , [-2,-1, 0101] , [-2,0, 0011]
debtArray = [0,0,3] , [0,0,0] , [-2,-3,2] , [-2,-2,2]
pathArray = [0,0] , [1,0] , [1,-1] , [1,-2] , [1,-3] , [1,-4] , [0,-4], [-1,-4] , [-2,-4] , [-2,-3] , [-2,-2] , [-2,-1]
openingCounter = 1
debt_coord =

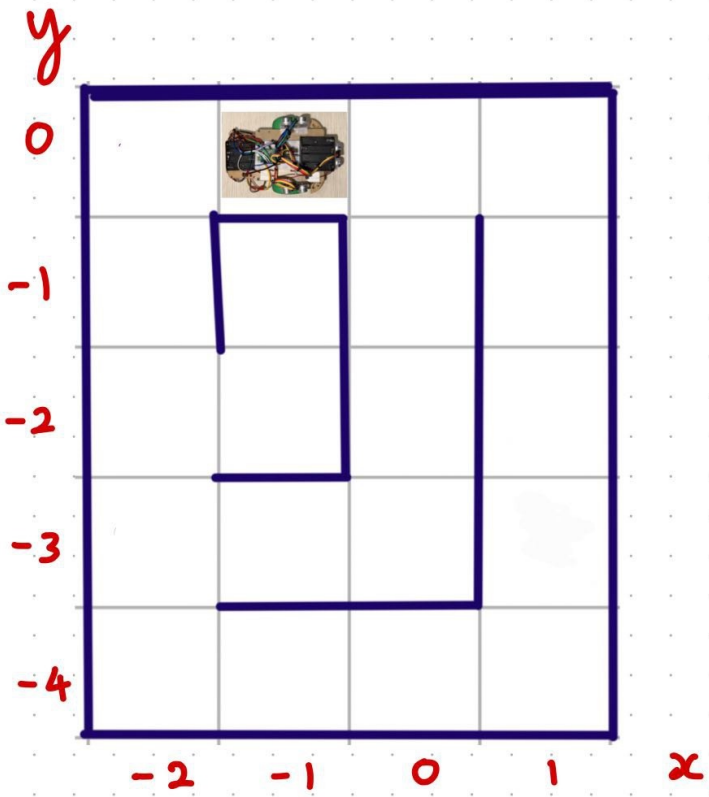
flag = 1



short int direction = 2;
short int isExit[2];
short int numberOfNodes = 12;
short int numberOfMoves = 12;
short int currentXCoord = -2;
short int currentYCoord = 0;
short int debtCounter = 3;

nodeArray = [0,0, 0010] , [1,0, 0110] , [1,-1, 0101] , [1,-2, 0101] , [1,-3, 0101] , [1,-4, 1100] , [0,-4, 1010] , [-1,-4, 1010] , [-2,-4, 1001] , [-2,-3, 0001] , [-2,-2,0001] , [-2,-1, 0101] , [-2,0, 0011] ,
debtArray = [0,0,3] , [0,0,0] , [-2,-3,2] , [-2,-2,2]
pathArray = [0,0] , [1,0] , [1,-1] , [1,-2] , [1,-3] , [1,-4] , [0,-4], [-1,-4] , [-2,-4] , [-2,-3] , [-2,-2] , [-2,-1] , [-2,0]
openingCounter = 1
debt_coord =

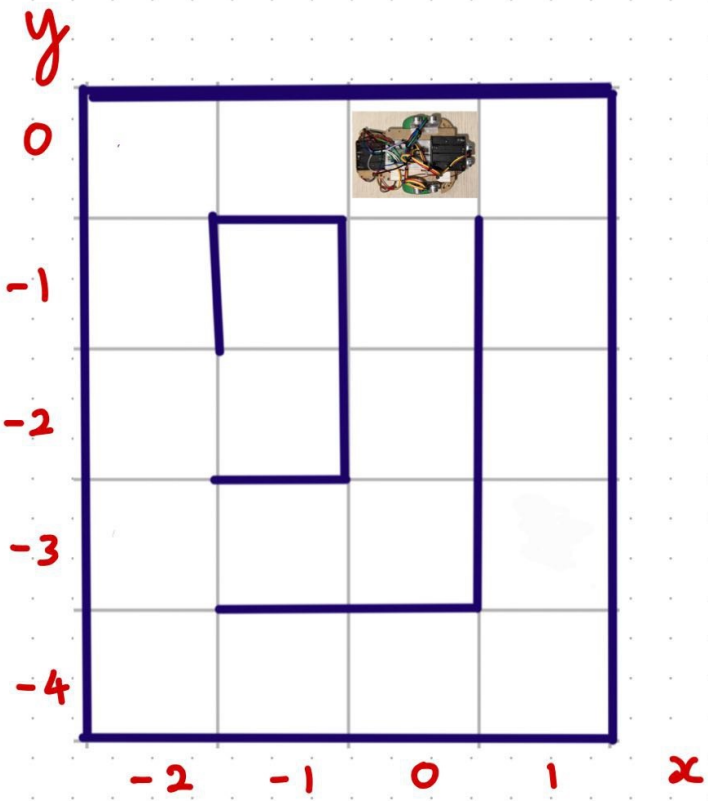
flag = 1



short int direction = 2;
short int isExit[2];
short int numberOfNodes = 13;
short int numberOfMoves = 13;
short int currentXCoord = -1;
short int currentYCoord = 0;
short int debtCounter = 3;

nodeArray = [0,0, 0010] , [1,0, 0110] , [1,-1, 0101] , [1,-2, 0101] , [1,-3, 0101] , [1,-4, 1100] , [0,-4, 1010] , [-1,-4, 1010] , [-2,-4, 1001] , [-2,-3, 0001] , [-2,-2,0001] , [-2,-1, 0101] , [-2,0, 0011] , [-1,0, 1010]
debtArray = [0,0,3] , [0,0,0] , [-2,-3,2] , [-2,-2,2]
pathArray = [0,0] , [1,0] , [1,-1] , [1,-2] , [1,-3] , [1,-4] , [0,-4], [-1,-4] , [-2,-4] , [-2,-3] , [-2,-2] , [-2,-1] , [-2,0] , [-1,0]
openingCounter = 1
debt_coord =

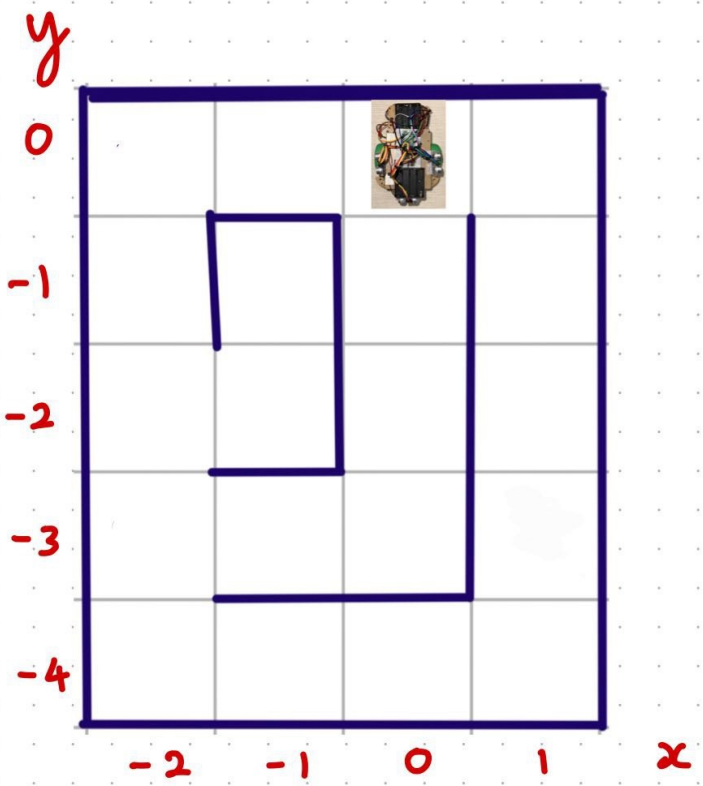
flag = 1



short int direction = 2;
short int isExit[2];
short int numberOfNodes = 13;
short int numberOfMoves = 13;
short int currentXCoord = 0;
short int currentYCoord = 0;
short int debtCounter = 3;

nodeArray = [0,0, 0010] , [1,0, 0110] , [1,-1, 0101] , [1,-2, 0101] , [1,-3, 0101] , [1,-4, 1100] , [0,-4, 1010] , [-1,-4, 1010] , [-2,-4, 1001] , [-2,-3, 0001] , [-2,-2,0001] , [-2,-1, 0101] , [-2,0, 0011] , [-1,0, 1010]
debtArray = [0,0,3] , [0,0,0] , [-2,-3,2] , [-2,-2,2]
pathArray = [0,0] , [1,0] , [1,-1] , [1,-2] , [1,-3] , [1,-4] , [0,-4], [-1,-4] , [-2,-4] , [-2,-3] , [-2,-2] , [-2,-1] , [-2,0] , [-1,0]
openingCounter = 1
debt_coord =

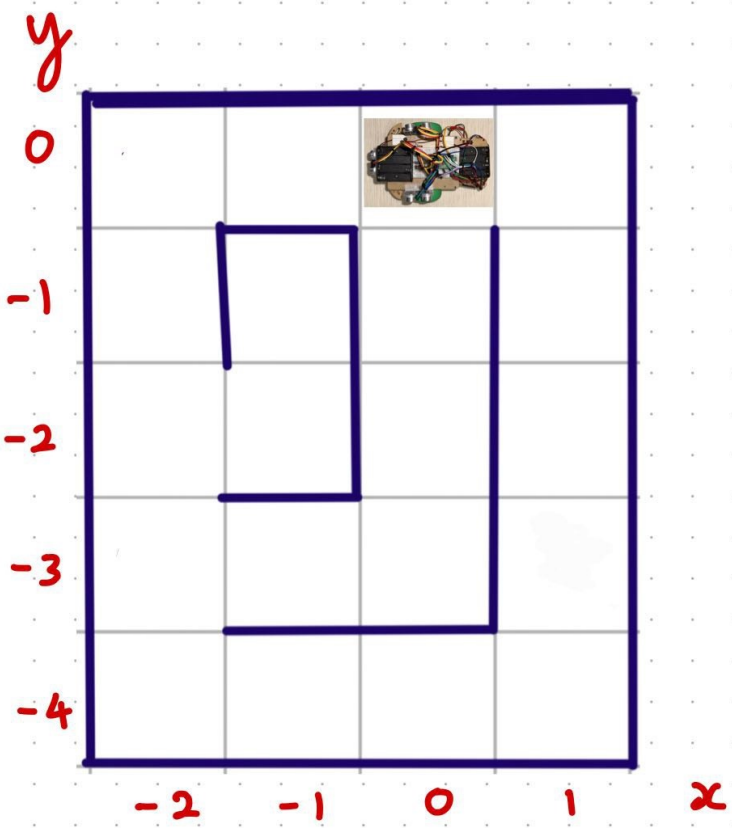
flag = 0



short int direction = 3;
short int isExit[2];
short int numberOfNodes = 13;
short int numberOfMoves = 13;
short int currentXCoord = 0;
short int currentYCoord = 0;
short int debtCounter = 3;

nodeArray = [0,0, 0010] , [1,0, 0110] , [1,-1, 0101] , [1,-2, 0101] , [1,-3, 0101] , [1,-4, 1100] , [0,-4, 1010] , [-1,-4, 1010] , [-2,-4, 1001] , [-2,-3, 0001] , [-2,-2,0001] , [-2,-1, 0101] , [-2,0, 0011] , [-1,0, 1010]
debtArray = [0,0,3] , [0,0,0] , [-2,-3,2] , [-2,-2,2]
pathArray = [0,0] , [1,0] , [1,-1] , [1,-2] , [1,-3] , [1,-4] , [0,-4], [-1,-4] , [-2,-4] , [-2,-3] , [-2,-2] , [-2,-1] , [-2,0] , [-1,0]
openingCounter = 1
debt_coord =

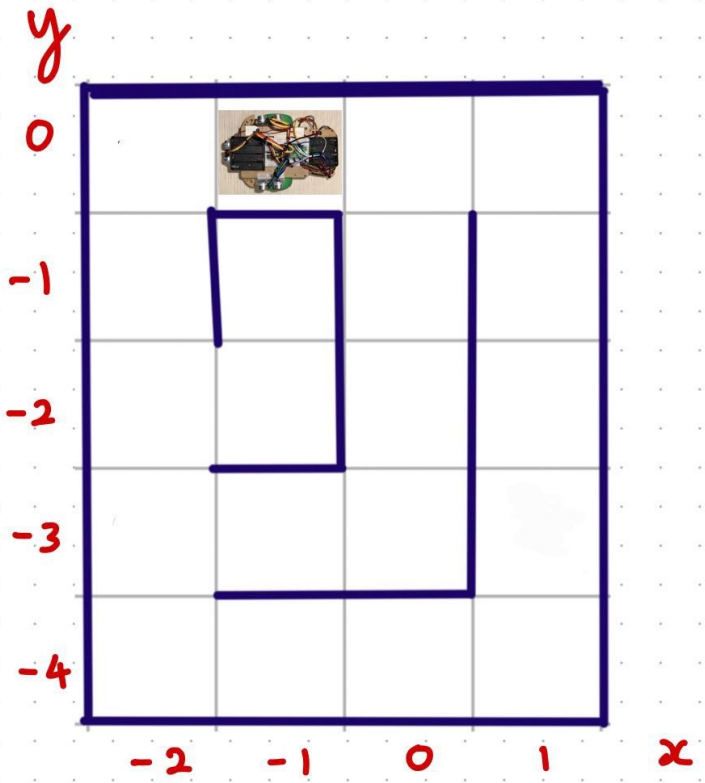
flag = 0



short int direction = 0;
short int isExit[2];
short int numberOfNodes = 13;
short int numberOfMoves = 13;
short int currentXCoord = 0;
short int currentYCoord = 0;
short int debtCounter = 3;

nodeArray = [0,0, 0010] , [1,0, 0110] , [1,-1, 0101] , [1,-2, 0101] , [1,-3, 0101] , [1,-4, 1100] , [0,-4, 1010] , [-1,-4, 1010] , [-2,-4, 1001] , [-2,-3, 0001] , [-2,-2,0001] , [-2,-1, 0101] , [-2,0, 0011] , [-1,0, 1010]
debtArray = [0,0,3] , [0,0,0] , [-2,-3,2] , [-2,-2,2]
pathArray = [0,0] , [1,0] , [1,-1] , [1,-2] , [1,-3] , [1,-4] , [0,-4], [-1,-4] , [-2,-4] , [-2,-3] , [-2,-2] , [-2,-1] , [-2,0] , [-1,0]
openingCounter = 1
debt_coord = [-2,-2,2]

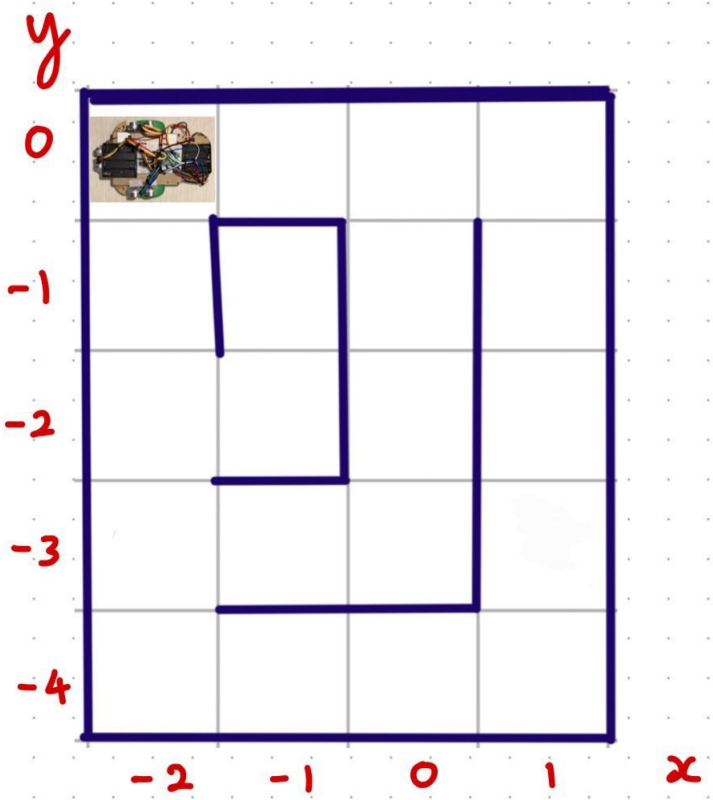
flag = 0



short int direction = 0;
short int isExit[2];
short int numberOfNodes = 13;
short int numberOfMoves = 12;
short int currentXCoord = -1;
short int currentYCoord = 0;
short int debtCounter = 3;

nodeArray = [0,0, 0010] , [1,0, 0110] , [1,-1, 0101] , [1,-2, 0101] , [1,-3, 0101] , [1,-4, 1100] , [0,-4, 1010] , [-1,-4, 1010] , [-2,-4, 1001] , [-2,-3, 0001] , [-2,-2,0001] , [-2,-1, 0101] , [-2,0, 0011] , [-1,0, 1010]
debtArray = [0,0,3] , [0,0,0] , [-2,-3,2] , [-2,-2,2]
pathArray = [0,0] , [1,0] , [1,-1] , [1,-2] , [1,-3] , [1,-4] , [0,-4], [-1,-4] , [-2,-4] , [-2,-3] , [-2,-2] , [-2,-1] , [-2,0]
openingCounter = 1
debt_coord = [-2,-2,2]

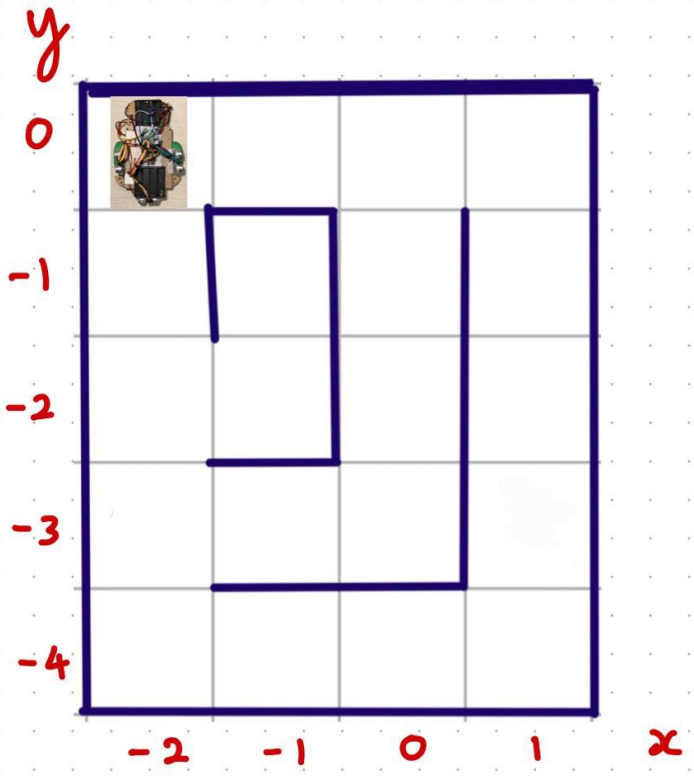
flag = 0



short int direction = 0;
short int isExit[2];
short int numberOfNodes = 13;
short int numberOfMoves = 11;
short int currentXCoord = -2;
short int currentYCoord = 0;
short int debtCounter = 3;

nodeArray = [0,0, 0010] , [1,0, 0110] , [1,-1, 0101] , [1,-2, 0101] , [1,-3, 0101] , [1,-4, 1100] , [0,-4, 1010] , [-1,-4, 1010] , [-2,-4, 1001] , [-2,-3, 0001] , [-2,-2,0001] , [-2,-1, 0101] , [-2,0, 0011] , [-1,0, 1010]
debtArray = [0,0,3] , [0,0,0] , [-2,-3,2] , [-2,-2,2]
pathArray = [0,0] , [1,0] , [1,-1] , [1,-2] , [1,-3] , [1,-4] , [0,-4], [-1,-4] , [-2,-4] , [-2,-3] , [-2,-2] , [-2,-1]
openingCounter = 1
debt_coord = [-2,-2,2]

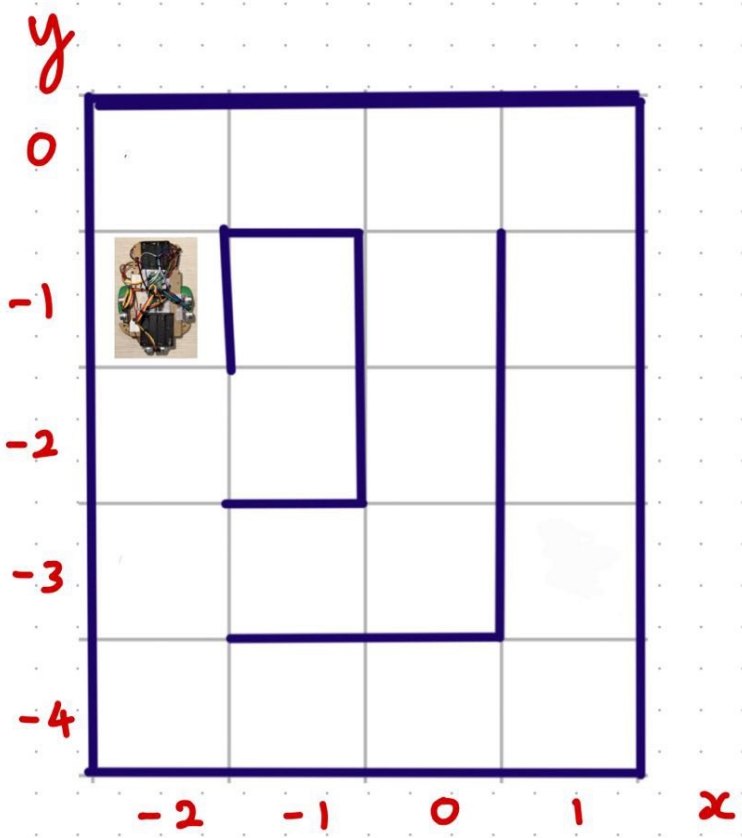
flag = 0



short int direction; = 3
short int isExit[2];
short int numberOfNodes = 13;
short int numberOfMoves = 11;
short int currentXCoord = -2;
short int currentYCoord = 0;
short int debtCounter = 3;

nodeArray = [0,0, 0010] , [1,0, 0110] , [1,-1, 0101] , [1,-2, 0101] , [1,-3, 0101] , [1,-4, 1100] , [0,-4, 1010] , [-1,-4, 1010] , [-2,-4, 1001] , [-2,-3, 0001] , [-2,-2,0001] , [-2,-1, 0101] , [-2,0, 0011] , [-1,0, 1010]
debtArray = [0,0,3] , [0,0,0] , [-2,-3,2] , [-2,-2,2]
pathArray = [0,0] , [1,0] , [1,-1] , [1,-2] , [1,-3] , [1,-4] , [0,-4], [-1,-4] , [-2,-4] , [-2,-3] , [-2,-2] , [-2,-1]
openingCounter = 1
debt_coord = [-2,-2,2]

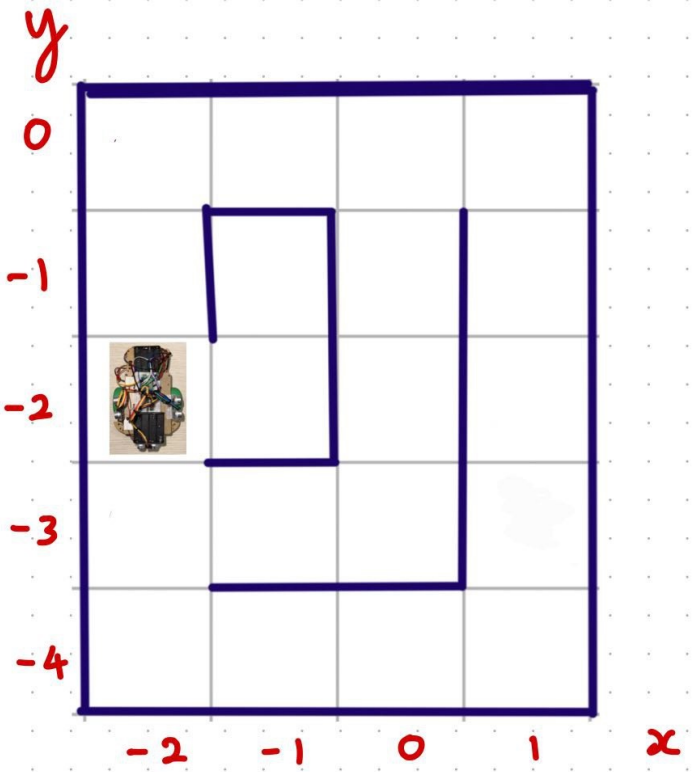
flag = 0



short int direction; = 3
short int isExit[2];
short int numberOfNodes = 13;
short int numberOfMoves = 10;
short int currentXCoord = -2;
short int currentYCoord = -1;
short int debtCounter = 3;

nodeArray = [0,0, 0010] , [1,0, 0110] , [1,-1, 0101] , [1,-2, 0101] , [1,-3, 0101] , [1,-4, 1100] , [0,-4, 1010] , [-1,-4, 1010] , [-2,-4, 1001] , [-2,-3, 0001] , [-2,-2,0001] , [-2,-1, 0101] , [-2,0, 0011] , [-1,0, 1010]
debtArray = [0,0,3] , [0,0,0] , [-2,-3,2] , [-2,-2,2]
pathArray = [0,0] , [1,0] , [1,-1] , [1,-2] , [1,-3] , [1,-4] , [0,-4], [-1,-4] , [-2,-4] , [-2,-3] , [-2,-2]
openingCounter = 1
debt_coord = [-2,-2,2]

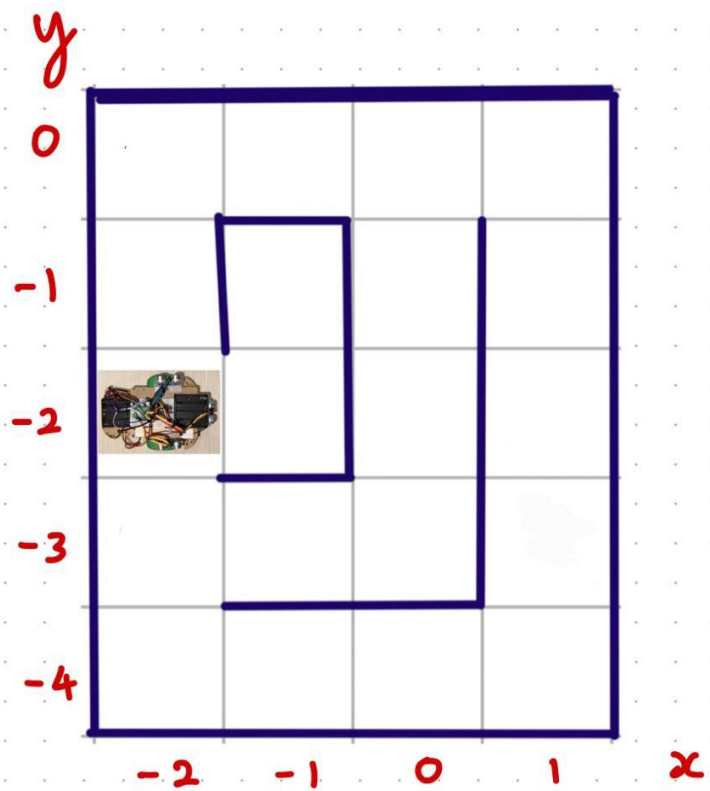
flag = 0



short int direction; = 3
short int isExit[2];
short int numberOfNodes = 13;
short int numberOfMoves = 10;
short int currentXCoord = -2;
short int currentYCoord = -2;
short int debtCounter = 3;

nodeArray = [0,0, 0010] , [1,0, 0110] , [1,-1, 0101] , [1,-2, 0101] , [1,-3, 0101] , [1,-4, 1100] , [0,-4, 1010] , [-1,-4, 1010] , [-2,-4, 1001] , [-2,-3, 0001] , [-2,-2,0001] , [-2,-1, 0101] , [-2,0, 0011] , [-1,0, 1010]
debtArray = [0,0,3] , [0,0,0] , [-2,-3,2] , [-2,-2,2]
pathArray = [0,0] , [1,0] , [1,-1] , [1,-2] , [1,-3] , [1,-4] , [0,-4], [-1,-4] , [-2,-4] , [-2,-3] , [-2,-2]
openingCounter = 1
debt_coord = [-2,-2,2]

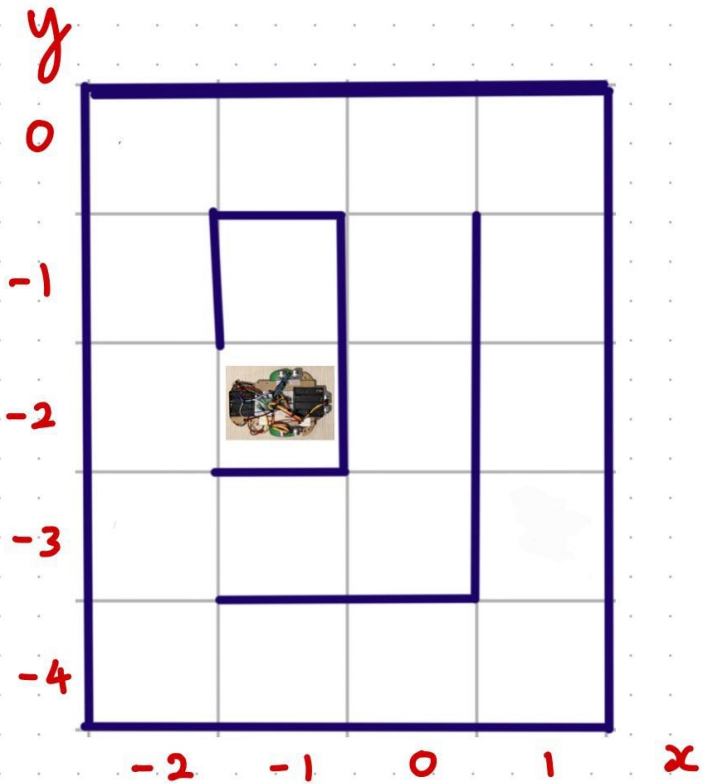
flag = 0



short int direction; = 2
short int isExit[2];
short int numberOfNodes = 13;
short int numberOfMoves = 10;
short int currentXCoord = -2;
short int currentYCoord = -2;
short int debtCounter = 2;

nodeArray = [0,0, 0010] , [1,0, 0110] , [1,-1, 0101] , [1,-2, 0101] , [1,-3, 0101] , [1,-4, 1100] , [0,-4, 1010] , [-1,-4, 1010] , [-2,-4, 1001] , [-2,-3, 0001] , [-2,-2,0001] , [-2,-1, 0101] , [-2,0, 0011] , [-1,0, 1010]
debtArray = [0,0,3] , [0,0,0] , [-2,-3,2]
pathArray = [0,0] , [1,0] , [1,-1] , [1,-2] , [1,-3] , [1,-4] , [0,-4], [-1,-4] , [-2,-4] , [-2,-3] , [-2,-2]
openingCounter = 1
debt_coord = [-2,-2,2]

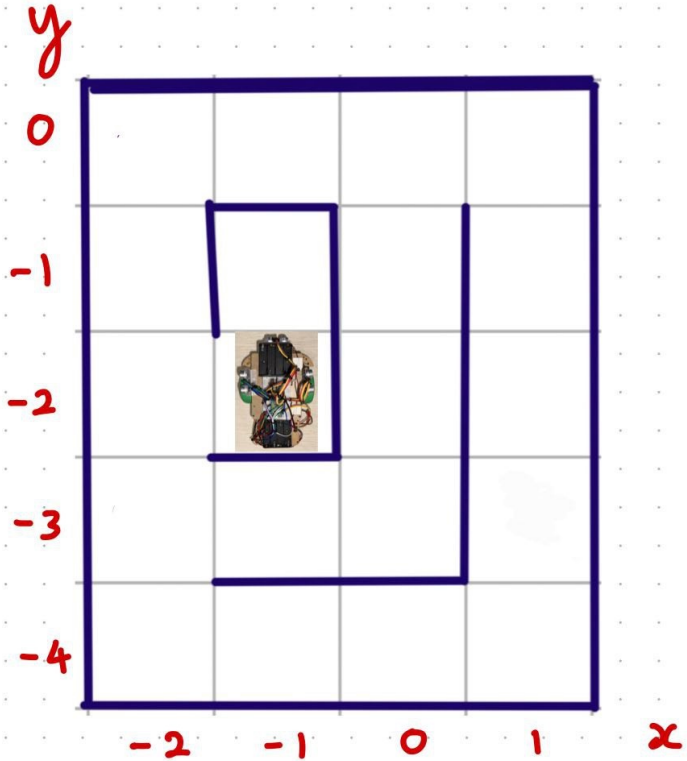
flag = 0



short int direction; = 2
short int isExit[2];
short int numberOfNodes = 14;
short int numberOfMoves = 10;
short int currentXCoord = -1;
short int currentYCoord = -2;
short int debtCounter = 2;

nodeArray = [0,0, 0010] , [1,0, 0110] , [1,-1, 0101] , [1,-2, 0101] , [1,-3, 0101] , [1,-4, 1100] , [0,-4, 1010] , [-1,-4, 1010] , [-2,-4, 1001] , [-2,-3, 0001] , [-2,-2,0001] , [-2,-1, 0101] , [-2,0, 0011] , [-1,0, 1010] , [-1,-2, 1100]
debtArray = [0,0,3] , [0,0,0] , [-2,-3,2]
pathArray = [0,0] , [1,0] , [1,-1] , [1,-2] , [1,-3] , [1,-4] , [0,-4], [-1,-4] , [-2,-4] , [-2,-3] , [-2,-2]
openingCounter = 1
debt_coord = [-2,-2,2]

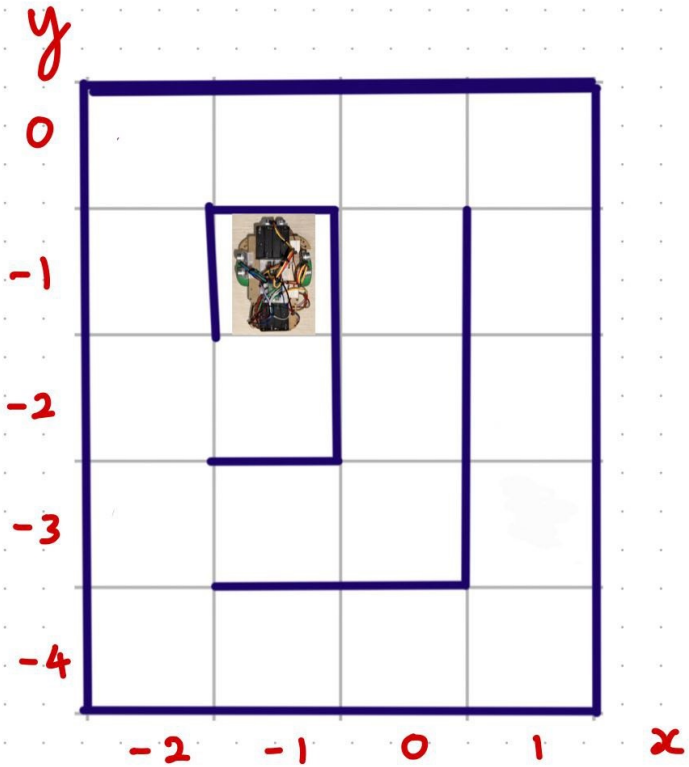
flag = 1



short int direction; = 1
short int isExit[2];
short int numberOfNodes = 14;
short int numberOfMoves = 11;
short int currentXCoord = -1;
short int currentYCoord = -2;
short int debtCounter = 2;

nodeArray = [0,0, 0010] , [1,0, 0110] , [1,-1, 0101] , [1,-2, 0101] , [1,-3, 0101] , [1,-4, 1100] , [0,-4, 1010] , [-1,-4, 1010] , [-2,-4, 1001] , [-2,-3, 0001] , [-2,-2,0001] , [-2,-1, 0101] , [-2,0, 0011] , [-1,0, 1010] , [-1,-2, 1100]
debtArray = [0,0,3] , [0,0,0] , [-2,-3,2]
pathArray = [0,0] , [1,0] , [1,-1] , [1,-2] , [1,-3] , [1,-4] , [0,-4], [-1,-4] , [-2,-4] , [-2,-3] , [-2,-2] , [-1,-2]
openingCounter = 1
debt_coord = [-2,-2,2]

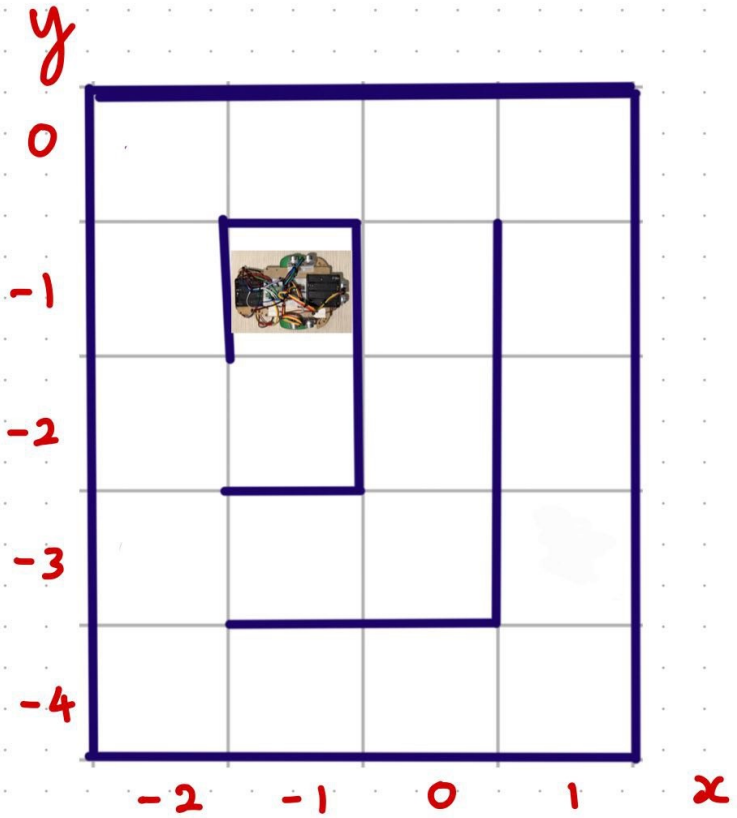
flag = 1



short int direction; = 1;
short int isExit[2];
short int numberOfNodes = 15;
short int numberOfMoves = 11;
short int currentXCoord = -1;
short int currentYCoord = -1;
short int debtCounter = 2;

nodeArray = [0,0, 0010] , [1,0, 0110] , [1,-1, 0101] , [1,-2, 0101] , [1,-3, 0101] , [1,-4, 1100] , [0,-4, 1010] , [-1,-4, 1010] , [-2,-4, 1001] , [-2,-3, 0001] , [-2,-2,0001] , [-2,-1, 0101] , [-2,0, 0011] , [-1,0, 1010] , [-1,-2, 1100] , [-1,-1, 0111]
debtArray = [0,0,3] , [0,0,0] , [-2,-3,2]
pathArray = [0,0] , [1,0] , [1,-1] , [1,-2] , [1,-3] , [1,-4] , [0,-4], [-1,-4] , [-2,-4] , [-2,-3] , [-2,-2] , [-1,-2]
openingCounter = 1
debt_coord = [-2,-2,2]

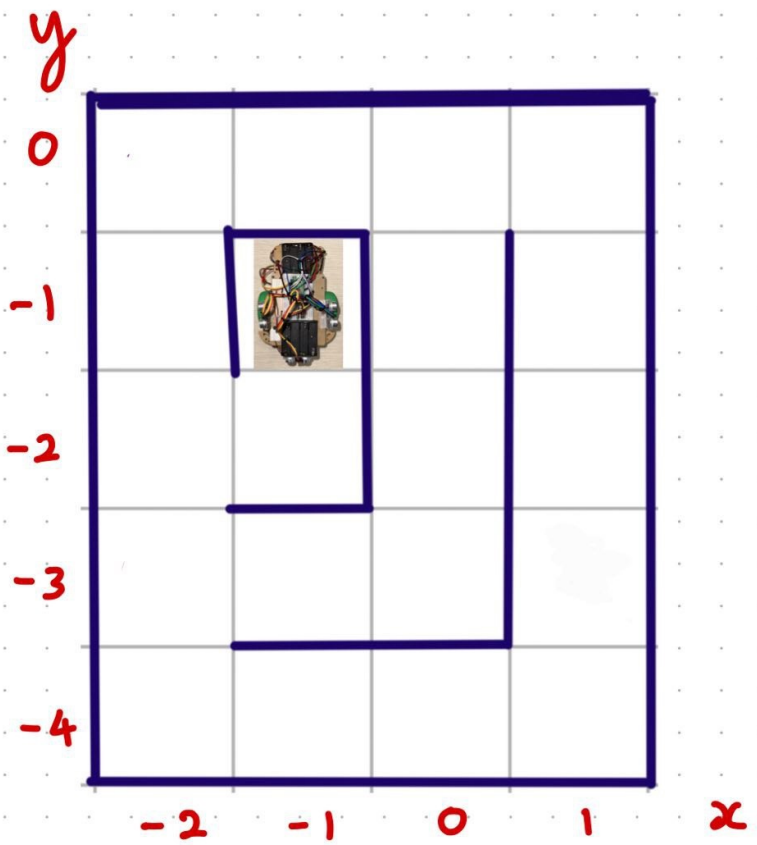
flag = 1



short int direction; = 2;
short int isExit[2];
short int numberOfNodes = 15;
short int numberOfMoves = 11;
short int currentXCoord = -1;
short int currentYCoord = -1;
short int debtCounter = 2;

nodeArray = [0,0, 0010] , [1,0, 0110] , [1,-1, 0101] , [1,-2, 0101] , [1,-3, 0101] , [1,-4, 1100] , [0,-4, 1010] , [-1,-4, 1010] , [-2,-4, 1001] , [-2,-3, 0001] , [-2,-2,0001] , [-2,-1, 0101] , [-2,0, 0011] , [-1,0, 1010] , [-1,-2, 1100] , [-1,-1, 0111]
debtArray = [0,0,3] , [0,0,0] , [-2,-3,2]
pathArray = [0,0] , [1,0] , [1,-1] , [1,-2] , [1,-3] , [1,-4] , [0,-4], [-1,-4] , [-2,-4] , [-2,-3] , [-2,-2] , [-1,-2]
openingCounter = 1
debt_coord = [-2,-2,2]

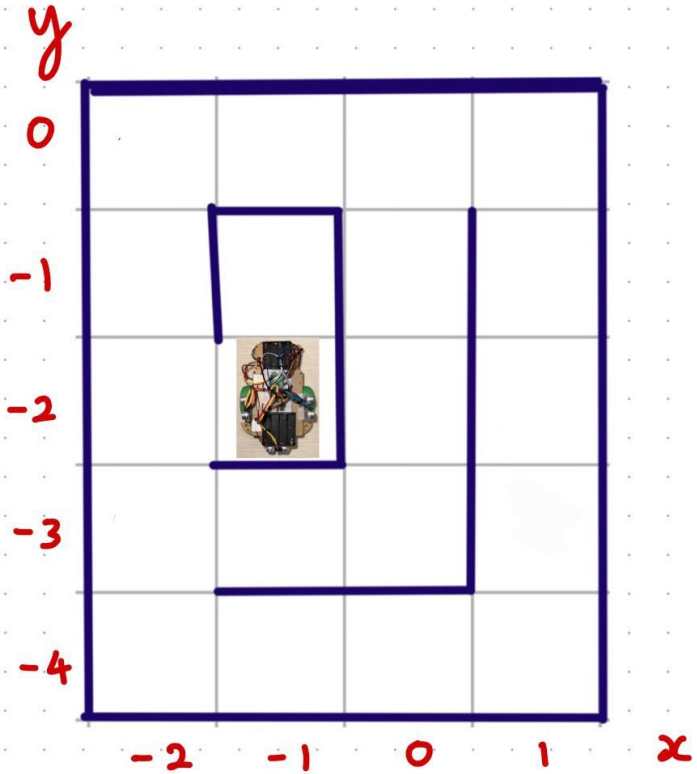
flag = 1



short int direction; = 3;
short int isExit[2];
short int numberOfNodes = 15;
short int numberOfMoves = 11;
short int currentXCoord = -1;
short int currentYCoord = -1;
short int debtCounter = 2;

nodeArray = [0,0, 0010] , [1,0, 0110] , [1,-1, 0101] , [1,-2, 0101] , [1,-3, 0101] , [1,-4, 1100] , [0,-4, 1010] , [-1,-4, 1010] , [-2,-4, 1001] , [-2,-3, 0001] , [-2,-2,0001] , [-2,-1, 0101] , [-2,0, 0011] , [-1,0, 1010] , [-1,-2, 1100] , [-1,-1, 0111]
debtArray = [0,0,3] , [0,0,0] , [-2,-3,2]
pathArray = [0,0] , [1,0] , [1,-1] , [1,-2] , [1,-3] , [1,-4] , [0,-4], [-1,-4] , [-2,-4] , [-2,-3] , [-2,-2] , [-1,-2]
openingCounter = 1
debt_coord = [-2,-3,2]

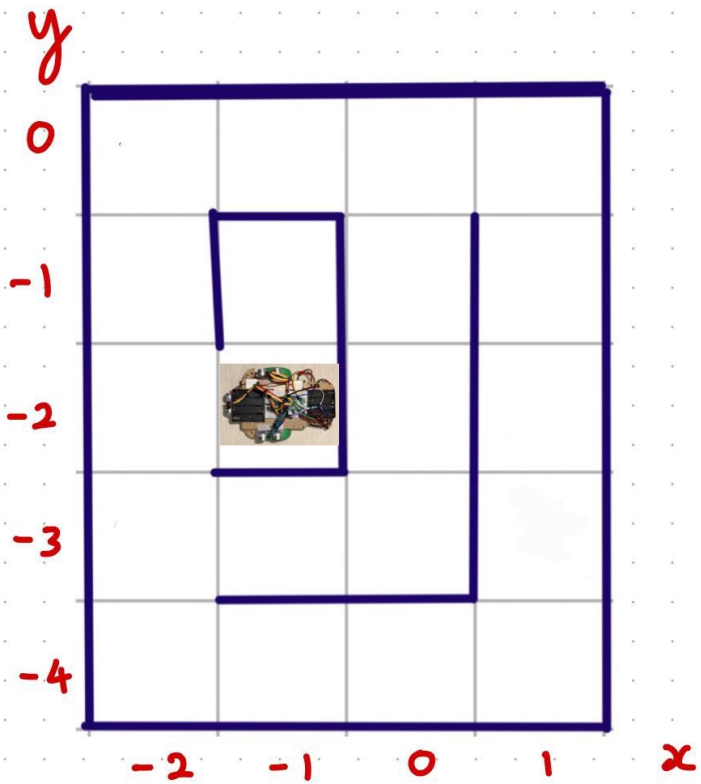
flag = 1



short int direction; = 3;
short int isExit[2];
short int numberOfNodes = 15;
short int numberOfMoves = 10;
short int currentXCoord = -1;
short int currentYCoord = -2;
short int debtCounter = 2;

nodeArray = [0,0, 0010] , [1,0, 0110] , [1,-1, 0101] , [1,-2, 0101] , [1,-3, 0101] , [1,-4, 1100] , [0,-4, 1010] , [-1,-4, 1010] , [-2,-4, 1001] , [-2,-3, 0001] , [-2,-2,0001] , [-2,-1, 0101] , [-2,0, 0011] , [-1,0, 1010] , [-1,-2, 1100] , [-1,-1, 0111]
debtArray = [0,0,3] , [0,0,0] , [-2,-3,2]
pathArray = [0,0] , [1,0] , [1,-1] , [1,-2] , [1,-3] , [1,-4] , [0,-4], [-1,-4] , [-2,-4] , [-2,-3] , [-2,-2]
openingCounter = 1
debt_coord = [-2,-3,2]

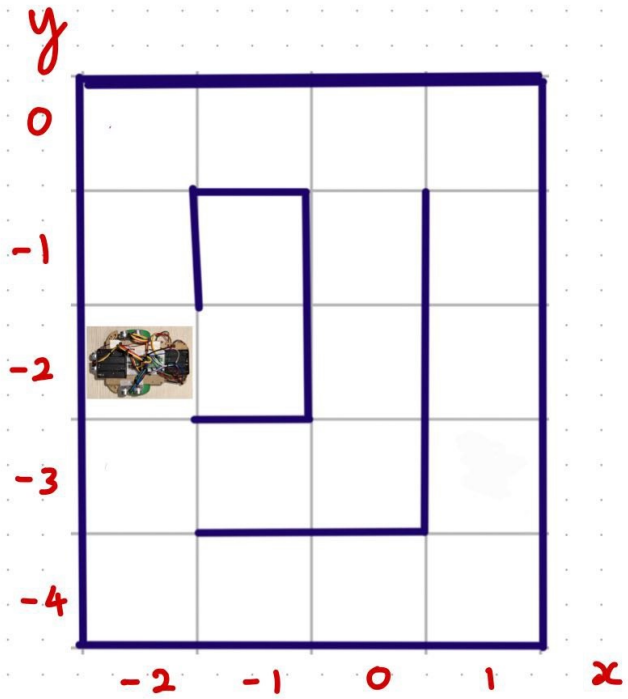
flag = 1



short int direction; = 0;
short int isExit[2];
short int numberOfNodes = 15;
short int numberOfMoves = 10;
short int currentXCoord = -1;
short int currentYCoord = -2;
short int debtCounter = 2;

nodeArray = [0,0, 0010] , [1,0, 0110] , [1,-1, 0101] , [1,-2, 0101] , [1,-3, 0101] , [1,-4, 1100] , [0,-4, 1010] , [-1,-4, 1010] , [-2,-4, 1001] , [-2,-3, 0001] , [-2,-2,0001] , [-2,-1, 0101] , [-2,0, 0011] , [-1,0, 1010] , [-1,-2, 1100] , [-1,-1, 0111]
debtArray = [0,0,3] , [0,0,0] , [-2,-3,2]
pathArray = [0,0] , [1,0] , [1,-1] , [1,-2] , [1,-3] , [1,-4] , [0,-4], [-1,-4] , [-2,-4] , [-2,-3] , [-2,-2]
openingCounter = 1
debt_coord = [-2,-3,2]

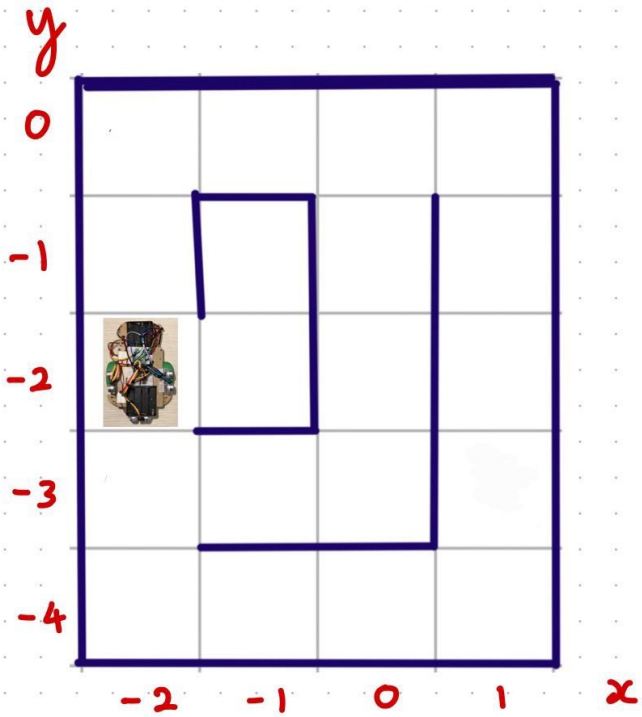
flag = 1



short int direction; = 0;
short int isExit[2];
short int numberOfNodes = 15;
short int numberOfMoves = 9;
short int currentXCoord = -2;
short int currentYCoord = -2;
short int debtCounter = 2;

nodeArray = [0,0, 0010] , [1,0, 0110] , [1,-1, 0101] , [1,-2, 0101] , [1,-3, 0101] , [1,-4, 1100] , [0,-4, 1010] , [-1,-4, 1010] , [-2,-4, 1001] , [-2,-3, 0001] , [-2,-2,0001] , [-2,-1, 0101] , [-2,0, 0011] , [-1,0, 1010] , [-1,-2, 1100] , [-1,-1, 0111]
debtArray = [0,0,3] , [0,0,0] , [-2,-3,2]
pathArray = [0,0] , [1,0] , [1,-1] , [1,-2] , [1,-3] , [1,-4] , [0,-4], [-1,-4] , [-2,-4] , [-2,-3]
openingCounter = 1
debt_coord = [-2,-3,2]

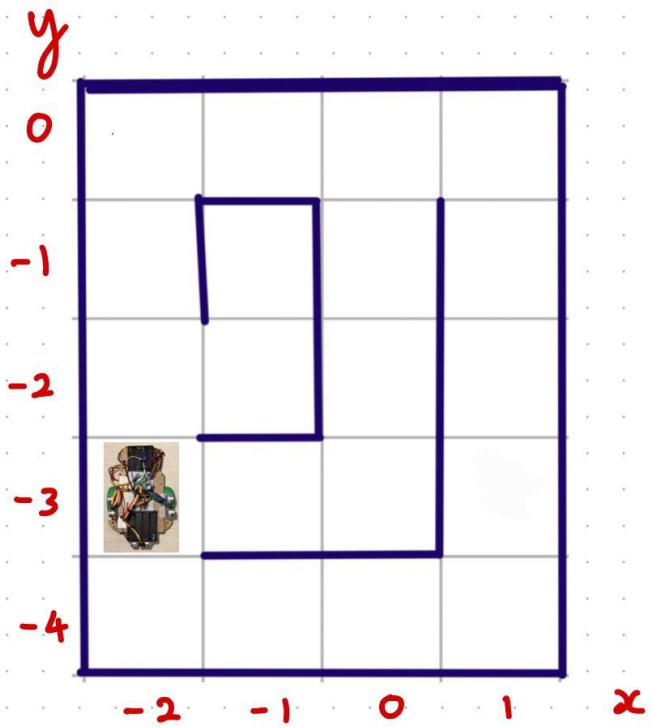
flag = 1



short int direction; = 3;
short int isExit[2];
short int numberOfNodes = 15;
short int numberOfMoves = 9;
short int currentXCoord = -2;
short int currentYCoord = -2;
short int debtCounter = 2;

nodeArray = [0,0, 0010] , [1,0, 0110] , [1,-1, 0101] , [1,-2, 0101] , [1,-3, 0101] , [1,-4, 1100] , [0,-4, 1010] , [-1,-4, 1010] , [-2,-4, 1001] , [-2,-3, 0001] , [-2,-2,0001] , [-2,-1, 0101] , [-2,0, 0011] , [-1,0, 1010] , [-1,-2, 1100] , [-1,-1, 0111]
debtArray = [0,0,3] , [0,0,0] , [-2,-3,2]
pathArray = [0,0] , [1,0] , [1,-1] , [1,-2] , [1,-3] , [1,-4] , [0,-4], [-1,-4] , [-2,-4] , [-2,-3]
openingCounter = 1
debt_coord = [-2,-3,2]

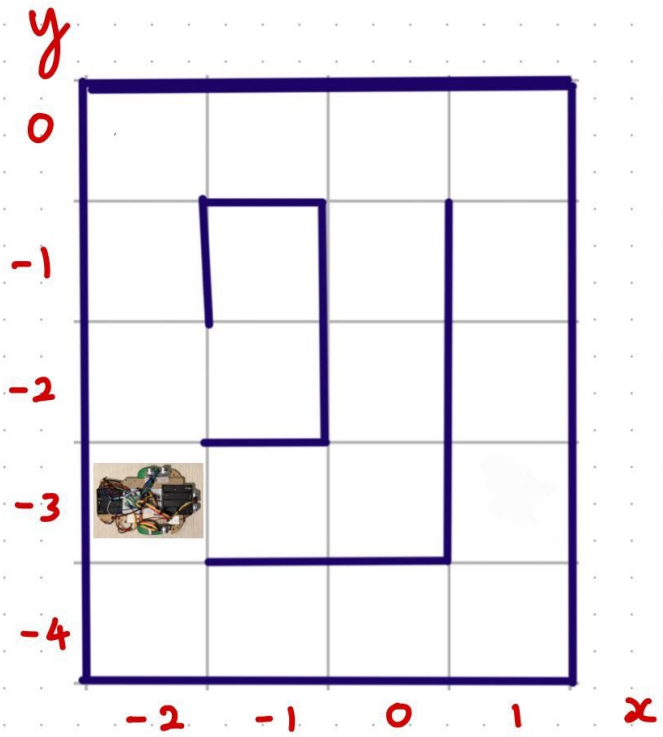
flag = 1



short int direction; = 3;
short int isExit[2];
short int numberOfNodes = 15;
short int numberOfMoves = 9;
short int currentXCoord = -2;
short int currentYCoord = -3;
short int debtCounter = 2;

nodeArray = [0,0, 0010] , [1,0, 0110] , [1,-1, 0101] , [1,-2, 0101] , [1,-3, 0101] , [1,-4, 1100] , [0,-4, 1010] , [-1,-4, 1010] , [-2,-4, 1001] , [-2,-3, 0001] , [-2,-2,0001] , [-2,-1, 0101] , [-2,0, 0011] , [-1,0, 1010] , [-1,-2, 1100] , [-1,-1, 0111]
debtArray = [0,0,3] , [0,0,0] , [-2,-3,2]
pathArray = [0,0] , [1,0] , [1,-1] , [1,-2] , [1,-3] , [1,-4] , [0,-4], [-1,-4] , [-2,-4] , [-2,-3]
openingCounter = 1
debt_coord = [-2,-3,2]

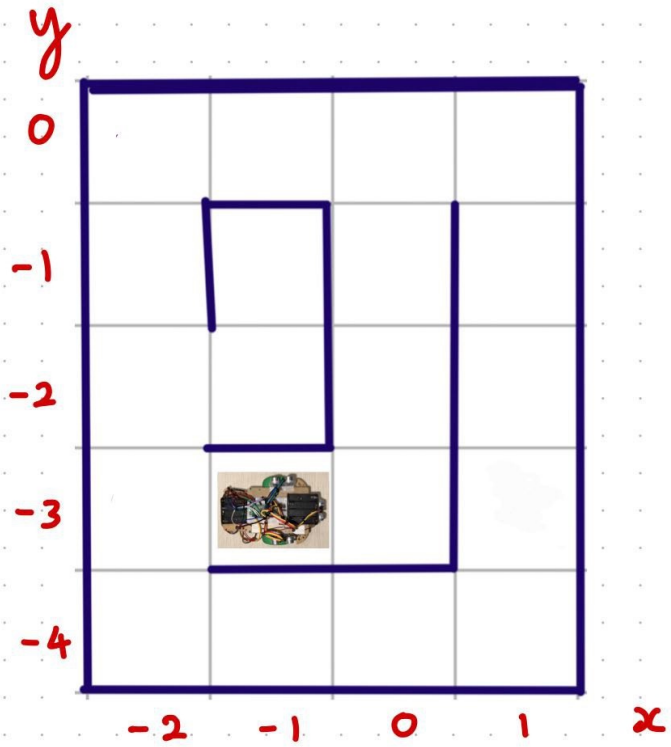
flag = 1



short int direction; = 2;
short int isExit[2];
short int numberOfNodes = 15;
short int numberOfMoves = 9;
short int currentXCoord = -2;
short int currentYCoord = -3;
short int debtCounter = 1;

nodeArray = [0,0, 0010] , [1,0, 0110] , [1,-1, 0101] , [1,-2, 0101] , [1,-3, 0101] , [1,-4, 1100] , [0,-4, 1010] , [-1,-4, 1010] , [-2,-4, 1001] , [-2,-3, 0001] , [-2,-2,0001] , [-2,-1, 0101] , [-2,0, 0011] , [-1,0, 1010] , [-1,-2, 1100] , [-1,-1, 0111]
debtArray = [0,0,3] , [0,0,0]
pathArray = [0,0] , [1,0] , [1,-1] , [1,-2] , [1,-3] , [1,-4] , [0,-4], [-1,-4] , [-2,-4] , [-2,-3]
openingCounter = 1
debt_coord = [-2,-3,2]

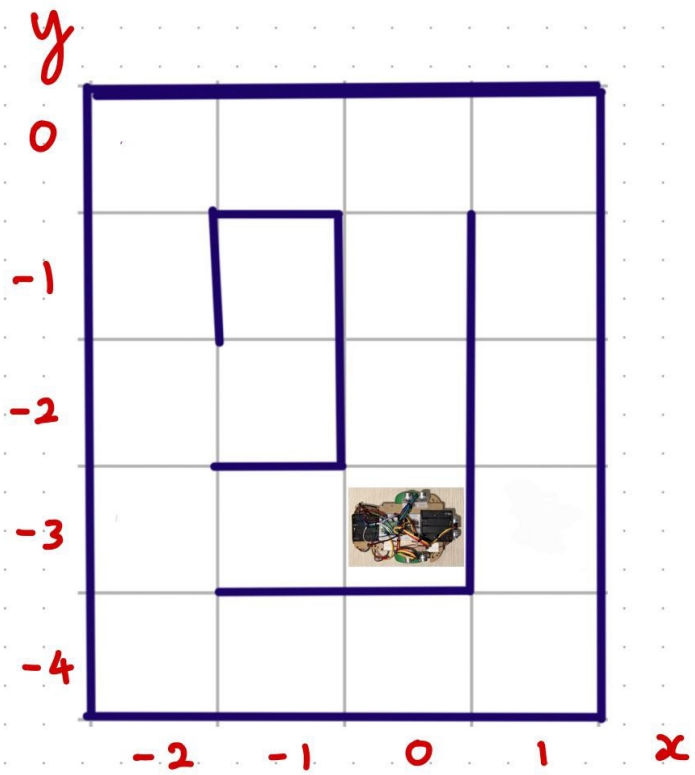
flag = 1



short int direction; = 2;
short int isExit[2];
short int numberOfNodes = 16;
short int numberOfMoves = 10;
short int currentXCoord = -1;
short int currentYCoord = -3;
short int debtCounter = 1;

nodeArray = [0,0, 0010] , [1,0, 0110] , [1,-1, 0101] , [1,-2, 0101] , [1,-3, 0101] , [1,-4, 1100] , [0,-4, 1010] , [-1,-4, 1010] , [-2,-4, 1001] , [-2,-3, 0001] , [-2,-2,0001] , [-2,-1, 0101] , [-2,0, 0011] , [-1,0, 1010] , [-1,-2, 1100] , [-1,-1, 0111] , [-1,-3, 1010]
debtArray = [0,0,3] , [0,0,0]
pathArray = [0,0] , [1,0] , [1,-1] , [1,-2] , [1,-3] , [1,-4] , [0,-4], [-1,-4] , [-2,-4] , [-2,-3] , [-1,-3]
openingCounter = 1
debt_coord = [-2,-3,2]

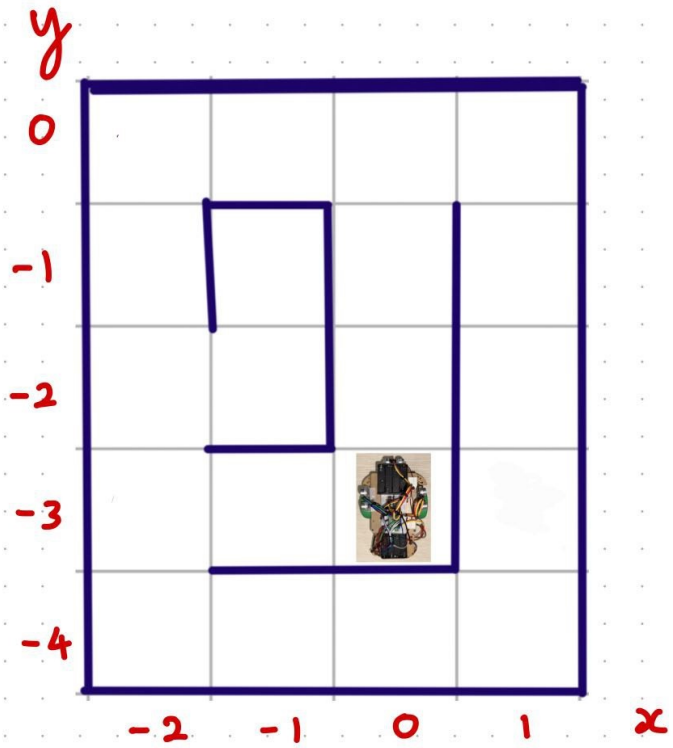
flag = 1



short int direction; = 2;
short int isExit[2];
short int numberOfNodes = 17;
short int numberOfMoves = 10;
short int currentXCoord = 0;
short int currentYCoord = -3;
short int debtCounter = 1;

nodeArray = [0,0, 0010] , [1,0, 0110] , [1,-1, 0101] , [1,-2, 0101] , [1,-3, 0101] , [1,-4, 1100] , [0,-4, 1010] , [-1,-4, 1010] , [-2,-4, 1001] , [-2,-3, 0001] , [-2,-2,0001] , [-2,-1, 0101] , [-2,0, 0011] , [-1,0, 1010] , [-1,-2, 1100] , [-1,-1, 0111] , [-1,-3, 1010] , [0,-3, 1100]
debtArray = [0,0,3] , [0,0,0]
pathArray = [0,0] , [1,0] , [1,-1] , [1,-2] , [1,-3] , [1,-4] , [0,-4], [-1,-4] , [-2,-4] , [-2,-3] , [-1,-3]
openingCounter = 1
debt_coord = [-2,-3,2]

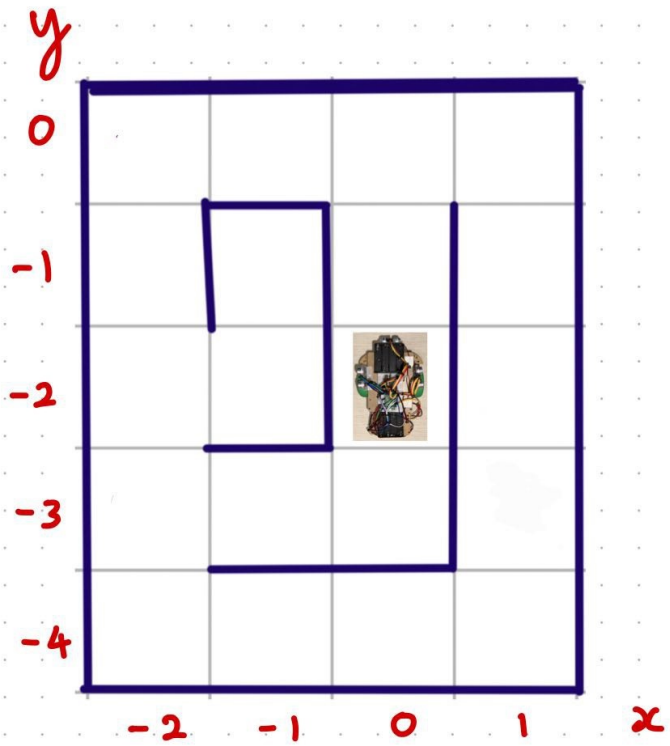
flag = 1



short int direction; = 1;
short int isExit[2];
short int numberOfNodes = 17;
short int numberOfMoves = 11;
short int currentXCoord = 0;
short int currentYCoord = -3;
short int debtCounter = 1;

nodeArray = [0,0, 0010] , [1,0, 0110] , [1,-1, 0101] , [1,-2, 0101] , [1,-3, 0101] , [1,-4, 1100] , [0,-4, 1010] , [-1,-4, 1010] , [-2,-4, 1001] , [-2,-3, 0001] , [-2,-2,0001] , [-2,-1, 0101] , [-2,0, 0011] , [-1,0, 1010] , [-1,-2, 1100] , [-1,-1, 0111] , [-1,-3, 1010] , [0,-3, 1100]
debtArray = [0,0,3] , [0,0,0]
pathArray = [0,0] , [1,0] , [1,-1] , [1,-2] , [1,-3] , [1,-4] , [0,-4], [-1,-4] , [-2,-4] , [-2,-3] , [-1,-3] , [0,-3]
openingCounter = 1
debt_coord = [-2,-3,2]

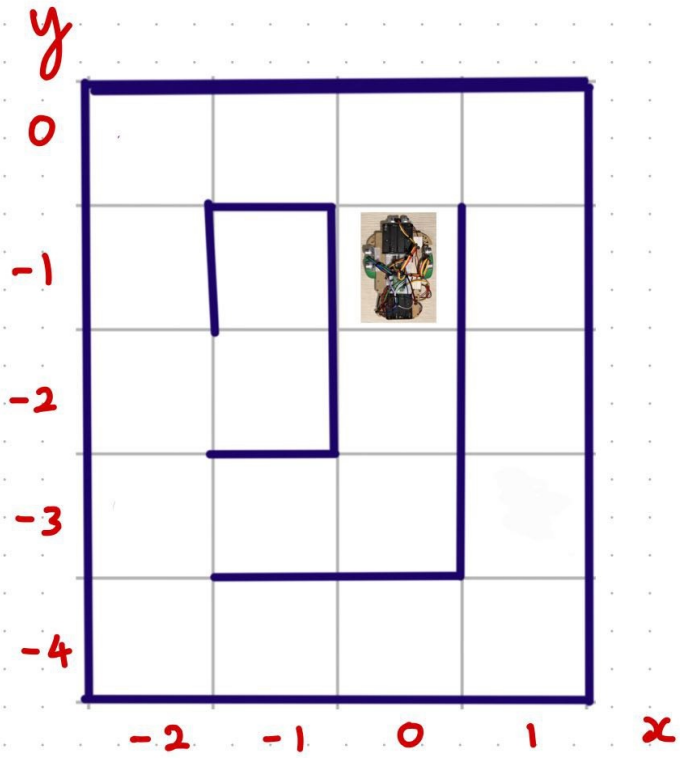
flag = 1



short int direction; = 1;
short int isExit[2];
short int numberOfNodes = 18;
short int numberOfMoves = 12;
short int currentXCoord = 0;
short int currentYCoord = -2;
short int debtCounter = 1;

nodeArray = [0,0, 0010] , [1,0, 0110] , [1,-1, 0101] , [1,-2, 0101] , [1,-3, 0101] , [1,-4, 1100] , [0,-4, 1010] , [-1,-4, 1010] , [-2,-4, 1001] , [-2,-3, 0001] , [-2,-2,0001] , [-2,-1, 0101] , [-2,0, 0011] , [-1,0, 1010] , [-1,-2, 1100] , [-1,-1, 0111] , [-1,-3, 1010] , [0,-3, 1100] , [0,-2, 0101]
debtArray = [0,0,3] , [0,0,0]
pathArray = [0,0] , [1,0] , [1,-1] , [1,-2] , [1,-3] , [1,-4] , [0,-4], [-1,-4] , [-2,-4] , [-2,-3] , [-1,-3] , [0,-3] , [-2,0]
openingCounter = 1
debt_coord = [-2,-3,2]

flag = 1

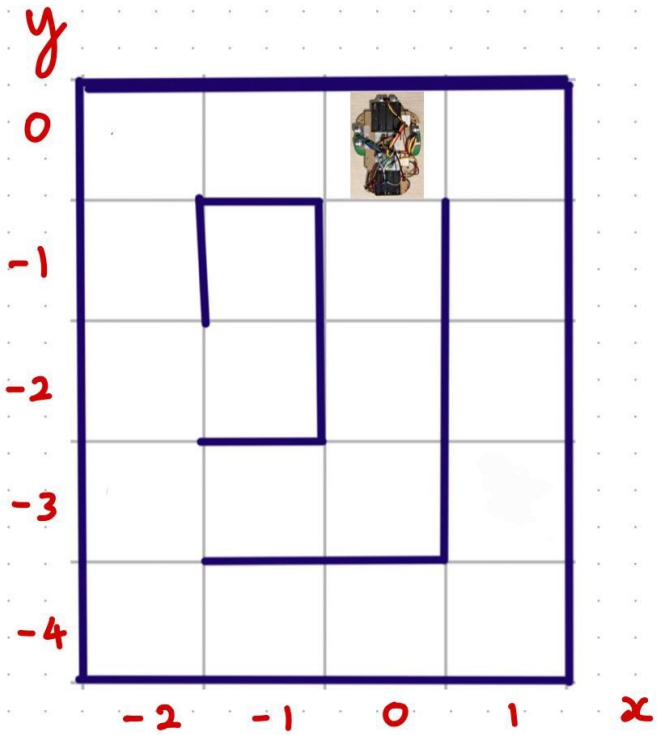


short int direction; = 1;
short int isExit[2];
short int numberOfNodes = 19;
short int numberOfMoves = 13;
short int currentXCoord = 0;
short int currentYCoord = -1;
short int debtCounter = 1;

nodeArray = [0,0, 0010] , [1,0, 0110] , [1,-1, 0101] , [1,-2, 0101] , [1,-3, 0101] , [1,-4, 1100] , [0,-4, 1010] , [-1,-4, 1010] , [-2,-4, 1001] , [-2,-3, 0001] , [-2,-2,0001] , [-2,-1, 0101] , [-2,0, 0011] , [-1,0, 1010] , [-1,-2, 1100] , [-1,-1, 0111] , [-1,-3, 1010] , [0,-3, 1100] , [0,-2, 0101] , [0,-1, 0101]

debtArray = [0,0,3] , [0,0,0]
pathArray = [0,0] , [1,0] , [1,-1] , [1,-2] , [1,-3] , [1,-4] , [0,-4], [-1,-4] , [-2,-4] , [-2,-3] , [-1,-3] , [0,-3] , [-2,0] , [-1,0]
openingCounter = 1
debt_coord = [-2,-3,2]

flag = 1



short int direction; = 1;
short int isExit[2];
short int numberOfNodes = 19;
short int numberOfMoves = 13;
short int currentXCoord = 0;
short int currentYCoord = 0;
short int debtCounter = 2;

nodeArray = [0,0, 0010] , [1,0, 0110] , [1,-1, 0101] , [1,-2, 0101] , [1,-3, 0101] , [1,-4, 1100] , [0,-4, 1010] , [-1,-4, 1010] , [-2,-4, 1001] , [-2,-3, 0001] , [-2,-2,0001] , [-2,-1, 0101] , [-2,0, 0011] , [-1,0, 1010] , [-1,-2, 1100] , [-1,-1, 0111] , [-1,-3, 1010] , [0,-3, 1100] , [0,-2, 0101] , [0,-1, 0101]

debtArray = [0,0,3] , [0,0,0]
pathArray = [0,0] , [1,0] , [1,-1] , [1,-2] , [1,-3] , [1,-4] , [0,-4], [-1,-4] , [-2,-4] , [-2,-3] , [-1,-3] , [0,-3] , [-2,0] , [-1,0]
openingCounter = 1
debt_coord = [-2,-3,2]

flag = 1

03
Navigation

04
Optimization

Optimization

Data Type	Int (4 bytes) ⟶ Short Int (2 bytes)
Bit Masking	Bit masking used for mapping, only 4 bits are used to set the walls. Hence, instead of 2 bytes of short int, 1 byte of character is used.
Functions	Repeated codes are placed into functions so that they can be reused.
Arrays	Better in terms of performance. Arrays are faster when it comes to element access and searching.
Pre/Post-fix	Pre-increment is used instead of post-increment, both performs the same functionality but it is faster and uses less resource due to the way it is implemented

Thank you:)