**/\*INPUT:**

**M[0..N-1][0..N-1] <- A 11x11 matrix which is basically the hex board**

**row <- the row selected**

**col <- the column selected**

**visited[0..N-1][0..N-1] <- A visited 11x11 matrix to keep track of the visited cells on the hex board**

**player <- the player currently playing (Player 1 or Player 2)**

**GLOBAL VARIABLES:**

**player1[0…N-1] <- store 1 if a connection to a new row is encountered**

**player2[0…N-1] <- store 1 if a connection to a new column is encountered**

**\*/**

**DFS(M[0…N-1][0…N-1],row,col,visited[0…N-1][0…N-1],player)**

**/\*Depth First Search to check if the surrounding hex cells have been marked\*/**

rowNbr[]<- {-1,-1,0,0,1,1}

colNbr[] <- {0,1,-1,1,-1,,0}

visited[row][col] <- TRUE **//Mark this cell as visited and recur for all it’s surrounding cells**

for k<-0 to 6 do

if (isSafe(M,row+rowNbr[k],col+colNbr[k],visited,player)

if (player=1)

player1[row+rowNbr[k]]<-1

else

player2[col+colNbr[k]]<-1

return DFS(M,row,col,visited,player)

**boolean isSafe(M[0…N-1][0…N-1],row,col,visited[0…N-1][0…N-1],player)**

return (row>=0) and (row<11) and (col>=0) and (col<11) and (M[row][col]=player and !visited[row][col])

**/\*INPUT:**

**M[0..N-1][0..N-1] <- A 11x11 matrix which is basically the hex board**

**player <- the player currently playing (Player 1 or Player 2)**

**OUTPUT:**

**TRUE , If the game has ended else FALSE \*/**

**boolean countConnected(M[0…N-1[0..N-1],player)**

Create a visited array to keep track of the cells visited

count1 <- 0 **// (Player 1 count)**

count2 <- 0 **// (Player2 count)**

s <-false **//Check if a player has won**

for i<-0 to N do

for j<-0 to N do

**// If a cell with the player number has not been visited yet**

if (M[i][j]=player and !visited[i][j])

DFS(M,i,j,visited,player)

for i<-1 to N do

if player2[i] =1

count2<-count2+1;

if player1[i]=1

count1<-count1+1;

if(count1=10 or count2=10)

s<-TRUE; **//Game has reached an end**

return s