Bfs code in 2D gird :(Updated)

int knightDir[8][2] = { {-2,1},{-1,2},{1,2},{2,1},{2,-1},{-1,-2},{1,-2},{-2,-1} };

//int dir4[4][2] = {{-1,0},{0,1},{1,0},{0,-1}};

//int dir8[8][2] = { {-1,0},{0,1},{1,0},{0,-1}, {-1,-1},{1,-1},{1,1},{-1,1} };

struct node

{

int x,y;

};

int board[1002][1002] ;

int vis[1002][1002];

int dis[1002][1002] ;

int N ;

bool checkInBoard(int x,int y)

{

return ((x>=1 && x<=N ) && (y>=1 && y<=N )) ;

}

void bfs()

{

memset(board,-1,sizeof(board)) ;

memset(vis,0,sizeof(vis));

queue<node> Q ;

int X = 1 ;

int Y = 1 ;

dis[X][Y] = 0 ;

Q.push({X,Y}) ;

while(!Q.empty())

{

node top = Q.front() ;

Q.pop() ;

int r = top.x ;

int c = top.y ;

for(int i=0 ; i<8; i++)

{

int row = r+ knightDir[i][0] ;

int col = c+ knightDir[i][1] ;

if( vis[row][col] == 0 )

{

if((board[row][col] == -1) && (checkInBoard(row,col) == true))

{

board[row][col]= 1;

vis[row][col] = 1;

dis[row][col] = dis[r][c] + 1 ;

Q.push(node{row,col});

}

}

}

}

}