

## 9) KNIGHT TOUR PROBLEM:

**LINK OF EXPLANATION:** [📺 Knights Tour - Solution | Recursion | Data Structures and Algorithms](#)

### METHOD:

- 1) In this method, we use excess recursion.
- 2) In every function we call 8 functions recursively, because the knight can move to 8 other positions from a particular position.
- 3) So we make an array of this position to have handy access to the positions:  
`int a[8]={-2,-1,1,2,2,1,-1,-2};`  
`int b[8]={1,2,2,1,-1,-2,-2,-1};`
- 4) But we don't call the eight functions if a particular condition is satisfied by any one of the functions so we break.
- 5) If the condition is not satisfied then we continue calling the functions.
- 6) The base condition of this problem is only one.....that is when the count exceeds 63. This means all the 63 numbers are written in some unique positions so we have to return true at that moment.

### THE CODE OF THE PROGRAM:(code and the logic is entirely written by me)

```
#include<iostream>
#include<vector>
using namespace std;

void func(vector<vector<int>> &answer,int i,int j,int count,int n,bool &status){
    if(count==n*n){
        status=true;
    }
    else if(i<0 || i>=n || j<0 || j>=n){
        return;
    }
    else if(answer[i][j]!=-1){
        return;
    }
    else{
        int a[8]={-2,-1,1,2,2,1,-1,-2};
        int b[8]={1,2,2,1,-1,-2,-2,-1};
        for(int index=0;index<8;index++){
            answer[i][j]=count;
            func(answer,i+a[index],j+b[index],count+1,n,status);
            if(status==true){
                return;
            }
            else{
                answer[i][j]=-1;
            }
        }
    }
}
```

```
    }  
}  
  
int main(){  
    int n;  
    cout<<"\n Enter the dimension:";  
    cin>>n;  
    vector<vector<int>> answer(n,vector<int>(n));  
    for(int i=0;i<n;i++){  
        for(int j=0;j<n;j++){  
            answer[i][j]=-1;  
        }  
    }  
    bool status=false;  
    func(answer,0,0,0,n,status);  
    cout<<"\n The answer:";  
    for(int i=0;i<n;i++){  
        cout<<"\n ";  
        for(int j=0;j<n;j++){  
            cout<<answer[i][j]<<" ";  
        }  
    }  
    return 0;  
}
```