**C++ Code for N-Queens Problem:**

**#include<iostream>**

**using namespace std;**

**bool isSafe(int \*\* arr , int x ,int y , int n){**

**for(int row = 0;row<x;row++)**

**{**

**if(arr[row][y]==1)**

**{**

**return false;**

**}**

**}**

**int row=x;**

**int col=y;**

**while(row>=0 && col>=0)**

**{**

**if(arr[row][col]==1){**

**return false;**

**}**

**row--;**

**col--;**

**}**

**row=x;**

**col=y;**

**while(row>=0 && col<n)**

**{**

**if(arr[row][col]==1){**

**return false;**

**}**

**row--;**

**col++;**

**}**

**return true;**

**}**

**bool nQueen(int \*\*arr , int x, int n)**

**{**

**if(x>=n){**

**return true;**

**}**

**for(int col=0;col<n;col++)**

**{**

**if(isSafe(arr,x,col,n))**

**{**

**arr[x][col]=1;**

**if(nQueen(arr,x+1,n)){**

**return true;**

**}**

**arr[x][col]=0; // Here we are performing backtracking**

**}**

**}**

**return false;**

**}**

**int main()**

**{**

**int n;**

**cout<<"Enter the value of n for n x n board : ";**

**cin>>n;**

**int \*\* arr = new int \*[n];**

**for(int i=0;i<n;i++)**

**{**

**arr[i]= new int[n];**

**for(int j=0;j<n;j++){**

**arr[i][j]=0;**

**}**

**}**

**cout<<"\nThe solution for "<<n<<" Queens Problem is : \n";**

**if(nQueen(arr,0,n)){**

**for(int i=0;i<n;i++)**

**{**

**for(int j=0;j<n;j++){**

**cout<<arr[i][j]<<" ";**

**}cout<<endl;**

**}**

**cout<<"\nNote: Here 1 represents that Queen is placed at that position.\n";**

**}**

**return 0;**

**}**

**OUTPUT :**

**Enter the value of n for n x n board : 8**

**The solution for 8 Queens Problem is :**

**1 0 0 0 0 0 0 0**

**0 0 0 0 1 0 0 0**

**0 0 0 0 0 0 0 1**

**0 0 0 0 0 1 0 0**

**0 0 1 0 0 0 0 0**

**0 0 0 0 0 0 1 0**

**0 1 0 0 0 0 0 0**

**0 0 0 1 0 0 0 0**

**Note: Here 1 represents that Queen is placed at that position.**

**PS C:\Users\vinayak jalan\Desktop\Programs\Artificial Intelligence\Ass-4>**