### Implement N-Queen's problem using Back Tracking.

#### Code:

#include<iostream>

#include<cstdlib>

using namespace std;

int \*a;

int count=0;

int right\_place(int pos){

    for(int i =1;i<pos;i++){

        if(a[i]==a[pos])

         return 0;

        if(labs((a[i]-a[pos]))==labs(i-pos))

         return 0;

    }

    return 1;

}

void print(int n){

    int i,j;

    count++;

    cout<<"\nSolution : "<<count<<endl;

    for(i=1;i<=n;i++){

        for (j=1;j<=n;j++){

            if(a[i]==j)

             cout<<"Q ";

            else

             cout<<"\* ";

        }

        cout<<endl;

    }

}

void Nqueens(int n){

    int k = 1;

    a[k] = 0;

    while(k!=0){

       a[k] = a[k]+1;

       while((a[k]<=n) && !right\_place(k))

        a[k]++;

       if(a[k]<=n){

           if(k==n)

             print(n);

           else{

               k++;

               a[k]=0;

           }

       }

       else

        k--;

    }

}

int main(){

    int n;

    cout<<"Enter number of Queens\n";

    cin>>n;

    a = new int[n];

    Nqueens(n);

    cout<<"Total number of solutions : "<<count<<endl;

    return 0;

}

#### Output:



