Program 12:

Design and implement C/C++ program for N Queen's problem using Backtracking.

Algorithm:

```
Algorithm NQueens (k, n)
//Using backtracking, this procedure prints all possible placements of n queens
//on an n x n chessboard so that they are non-attacking
{
        for i ← 1 to n do
        {
                if(Place(k,i) )
                {
                        x[k] ← i
                        if (k=n)
                                write ( x[1...n])
                        else
                                Nqueens (k+1, n)
                }
        }
}

Algorithm Place( k, i)
//Returns true if a queen can be placed in kth row and ith column. Otherwise it
//returns false. x[] is a global array whose first (k-1) values have been set. Abs(r)
//returns the absolute value of r.
{
        for j ← 1 to k-1 do
        {
                if (  (x[j]=i or Abs(x[j]-i) = Abs(j-k) )
                {
                        return false
                }
        }
}
```

Code:

```
#include<stdio.h>

#include<math.h>

#include<stdlib.h>

int place(int x[], int k)

{

for(int i=1;i<k;i++)
```

```
{
if((x[i] == x[k]) || (abs(x[i]-x[k]) == abs(i-k)))
return 0;
}
return 1;
}
int nqueens(int n)
{
int x[10],k,count=0;
k=1;
x[k]=0;
while(k!=0)
{
x[k]++;
while((x[k]<=n) && (!place(x,k)))
x[k]++;
if(x[k]<=n)
{
if(k==n)
{
printf("\nSolution %d\n", ++count);
for(int i=1;i<=n;i++)
{
for(int j=1;j<=n;j++)
```

```
printf("%c", j==x[i]?'Q':'X');
printf("\n");
}
}
else
{
++k;
x[k]=0;
}
}
else
k--;
}
return count;
}
void main()
{
int n;
printf("Enter the size of chessboard: ");
scanf("%d",&n);
printf("\n The number of possibilities are %d\n", nqueens(n));
}
```

Output:

```
sru-ubuntu@srujani-Ubuntu-VirtualBox:~$ gcc p12.c
sru-ubuntu@srujani-Ubuntu-VirtualBox:~$ ./a.out
Enter the size of chessboard: 4

Solution 1
XQXX
XXXQ
QXXX
XXQX

Solution 2
XXQX
QXXX
XXXQ
XQXX

 The number of possibilities are 2
sru-ubuntu@srujani-Ubuntu-VirtualBox:~$ ./a.out
Enter the size of chessboard: 3

 The number of possibilities are 0
sru-ubuntu@srujani-Ubuntu-VirtualBox:~$ ./a.out
Enter the size of chessboard: 1

Solution 1
Q

 The number of possibilities are 1
sru-ubuntu@srujani-Ubuntu-VirtualBox:~$
```