Name:	**Omkar Deshmukh**
UID:	**2021700018**
Div:	**CSE(DS)D1**
Exp:	**07**
Aim:	**Backtracking (To implement N Queens problem using backtracking.)**

Algorithm:

```
1.  Place (k, i)
2.  {
3.  For j ← 1 to k - 1
4.  do if (x [j] = i)
5.  or (Abs x [j]) - i) = (Abs (j - k))
6.  then return false;
7.  return true;
8.  }
```

```
1.  N - Queens (k, n)
2.  {
3.  For i ← 1 to n
4.  do if Place (k, i) then
5.  {
6.  x [k] ← i
7.  if (k ==n) then
8.  write (x [1....n));
9.  else
10. N - Queens (k + 1, n);
```

	11. } 12. }

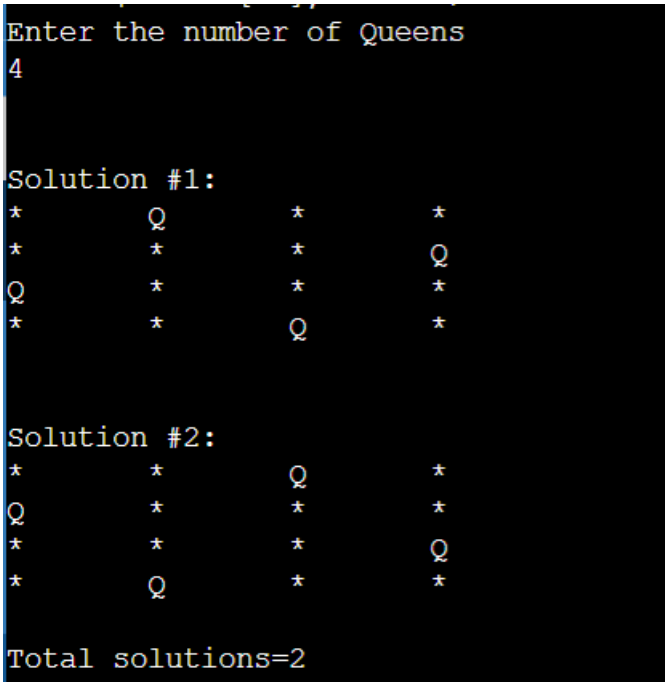
Code:-

```
#include<stdio.h>
#include<math.h>
int a[30],count=0;
int place(int pos) {
int i;

for (i = 1; i < pos; i++)
  {
if ((a[i] == a[pos]) || ((abs (a[i] - a[pos]) == abs (i - pos))))
     return 0;
}
return 1;
}
Void print_sol (int n)
{

int i, j;
  count++;

printf ("\n\nSolution #%d:\n", count);
  for (i = 1; i <= n; i++)
    {
```

```
      for (j = 1; j <= n; j++)
         {

if (a[i] == j)
            printf ("Q\t");
          else
            printf ("*\t");
         }

printf ("\n");
   }
}
Void queen (int n)
{
  int k = 1;
  a[k] = 0;
  while (k != 0)
    {
      a[k] = a[k] + 1;
      while ((a[k] <= n) && !place (k))
         a[k]++;
      if (a[k] <= n)
         {
           if (k == n)
             print_sol (n);
           else
             {
```

```
k++;
a[k] = 0;
            }
        }
     else
        k--;
  }
}


void main ()
{
  int i, n;

//      clrscr();
    printf ("Enter the number of Queens\n");
  scanf ("%d", &n);
  queen (n);

printf ("\nTotal solutions=%d", count);

//      getch();
}
```

Output:



Conclusion: In this experiment , I Have Understood the concept of backtracking and understood the Queens problem.