Name: Harsh Chandra
UID: 2021700013
CSE(DS) D1

EXP 7

Aim:Backtracking (To implement N Queens problem using backtracking.)

Algorithm:

1.Place(k,i)
2.{
3.Forj←1tok-1
4.doif(x[j]=i)
5.or(Absx[j])-i)=(Abs(j-k))
6.thenreturnfalse;
7.returntrue;
8. }

1.N-Queens(k,n)
2.{
3.Fori←1ton
4.doifPlace(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<stdlib.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--;
        }
}
int main() {
        int i,n;
        printf("Enter the number of Queens\n");
        scanf("%d",&n);
        queen(n);
        printf("\nTotal solutions=%d",count);
   return 0;
}
```

Output:

Conclusion: From this experiment, I understood implementation concept of backtracking in n queens problem.