|  |  |
| --- | --- |
| **Experiment No.** | **7** |
| **Aim** | **Experiment using backtracking strategy** |
| **Name** | **Rucha Sudhir Kulkarni** |
| **UID No.** | **2021300067** |
| **Class & Division** | **SE Computer Engineering (Div:A)(Batch:D)** |
| **Date of Performance:** | **12.04.2023** |
| **Date of Submission:** | **16.04.2023** |

**Aim: – To implement solution to N queens problem.**

**Algorithm:**

1. Start
2. Input number of queens ,n
3. Call queen(1,n)
4. Stop
5. Function queen(int k,int n):
   1. For all elements from i=1 to n
   2. If(place(k,i))
   3. X[k]=i
   4. If(k==n)
   5. Count++
   6. Print elements of x[] array
   7. Else
   8. Queen(k+1,n)
6. Function place(int k,int i):
   1. For i=1 to k-1
   2. If(x[j]==1) or abs(x[j]-i)==abs(j-k))
   3. Return false
   4. Return true

**Program:**

#include<stdio.h>

#include<math.h>

int x[30],count;

int place(int k,int i) {

    for (int j=1;j<k;j++) {

        if((x[j]==i)||(abs(x[j]-i)==abs(j-k)))

           return 0;

    }

    return 1;

}

void queen(int k,int n) {

    for(int i=1;i<=n;i++){

        //printf("place %d,%d==%d\n",k,i,place(k,i));

        if(place(k,i)==1){

            x[k]=i;

            if(k==n){

                count++;

                printf("solution %d: ",count);

                for(int a=1;a<=n;a++){

                    printf("%d ",x[a]);

                }

                printf("\n");

            }

            else{

                queen(k+1,n);

            }

        }

    }

}

void main() {

    int i,n;

    count=0;

    printf("Enter the number of Queens\n");

    scanf("%d",&n);

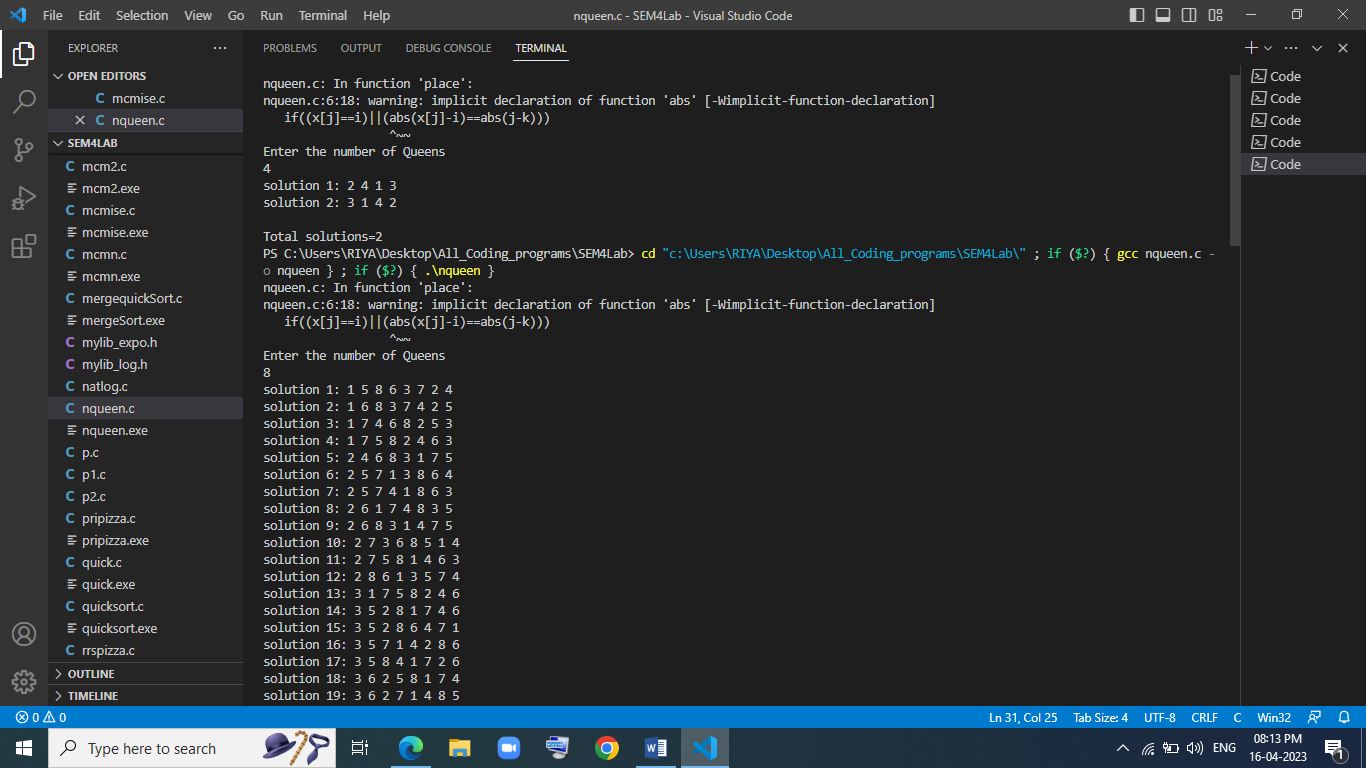
    queen(1,n);

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

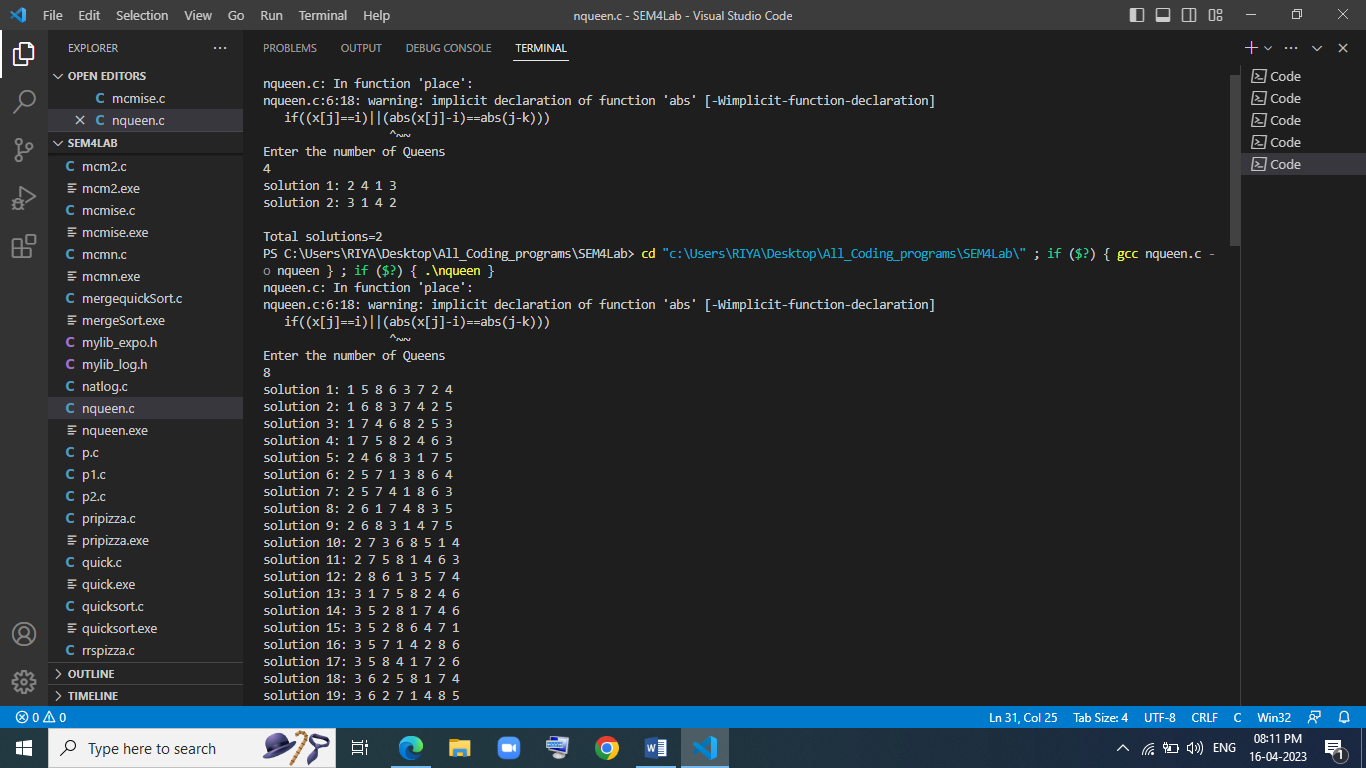
}

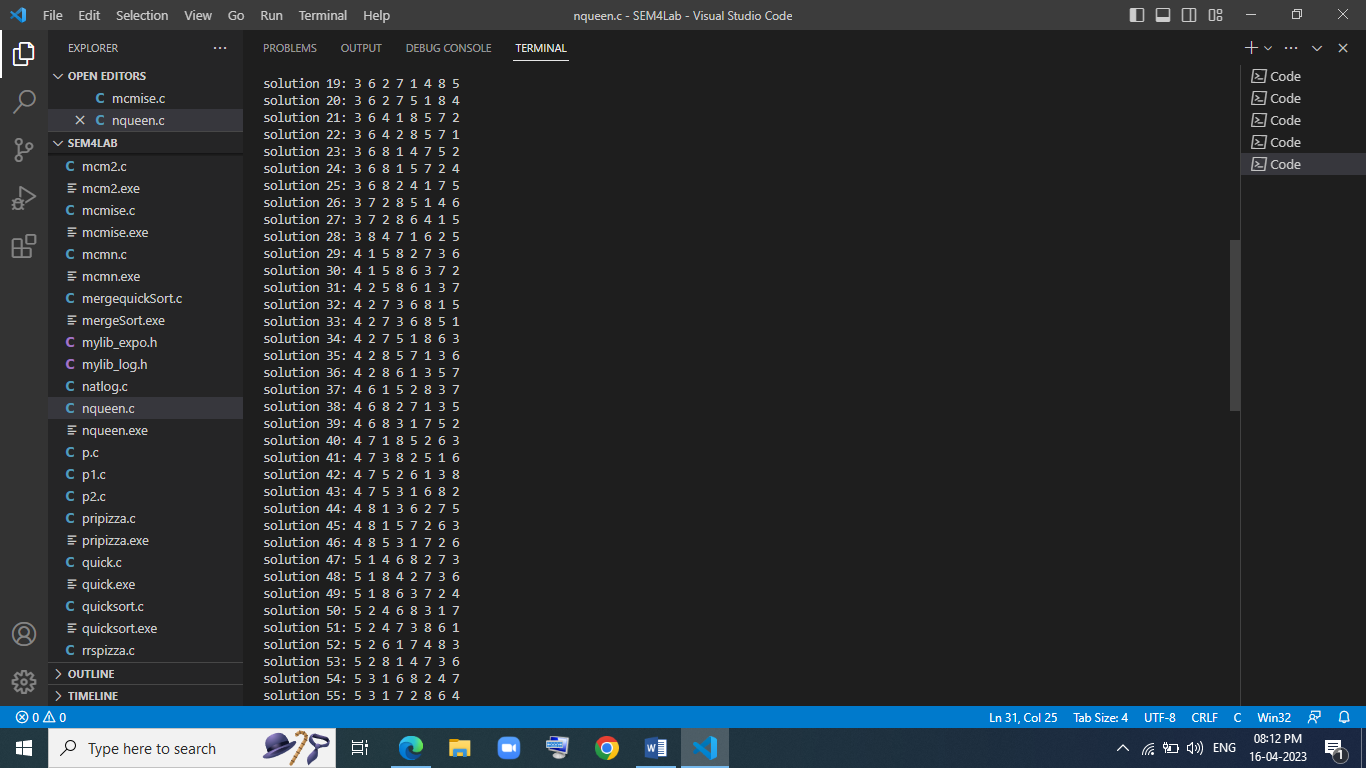
**Output and Observation:**

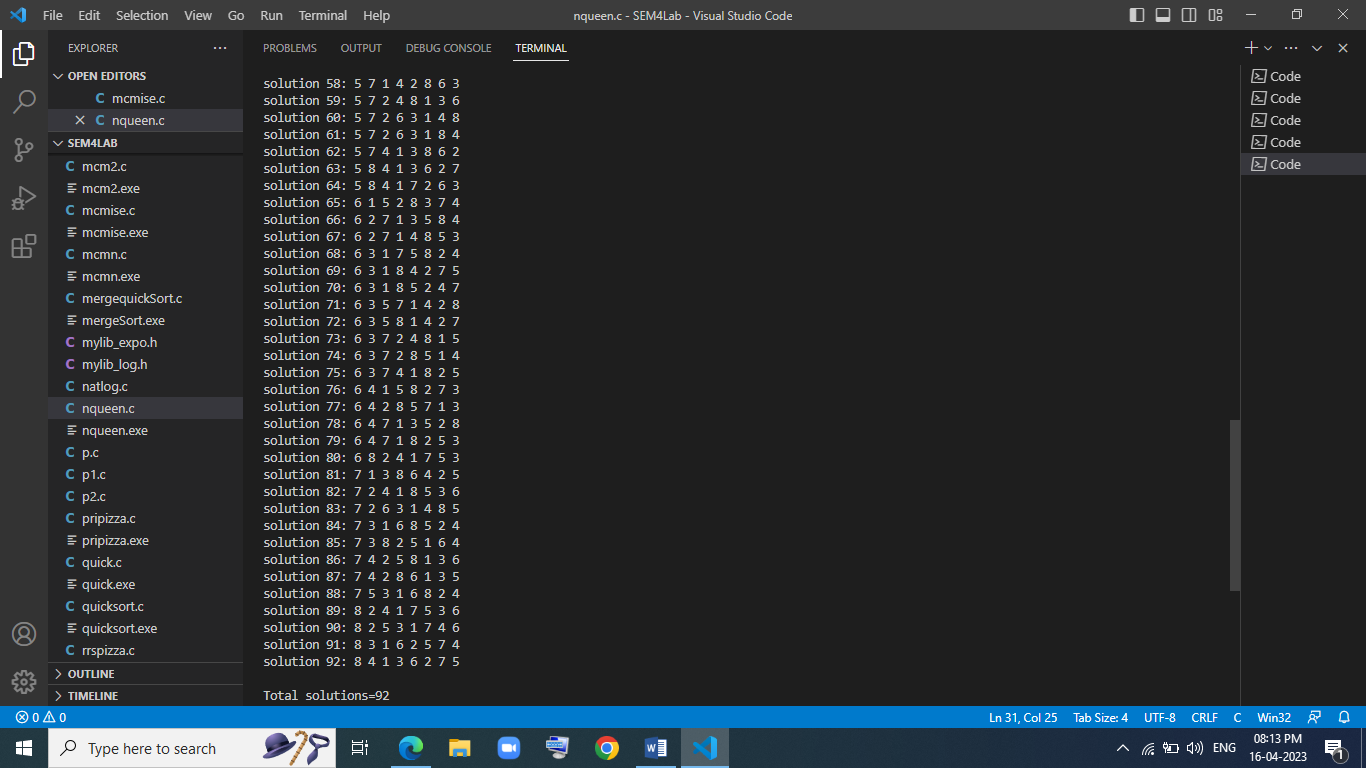
**4 queens problem has 2 possible solutions:**



**8 queens problem has 92 possible solutions:**







**Conclusion:**

After performing the above experiment, I got to know what is N queen’s problem and how to solve it using backtracking strategy.