Q.) The "*4-Queens Problem*" consists of placing four queens on a 4 x 4 chessboard so that no two queens can capture each other. That is, no two queens are allowed to be placed on the same row, the same column or the same diagonal (both primary and secondary diagonals). Write a C program to simulate the given problem and perform the algorithmic complexity analysis for the solution you propose.

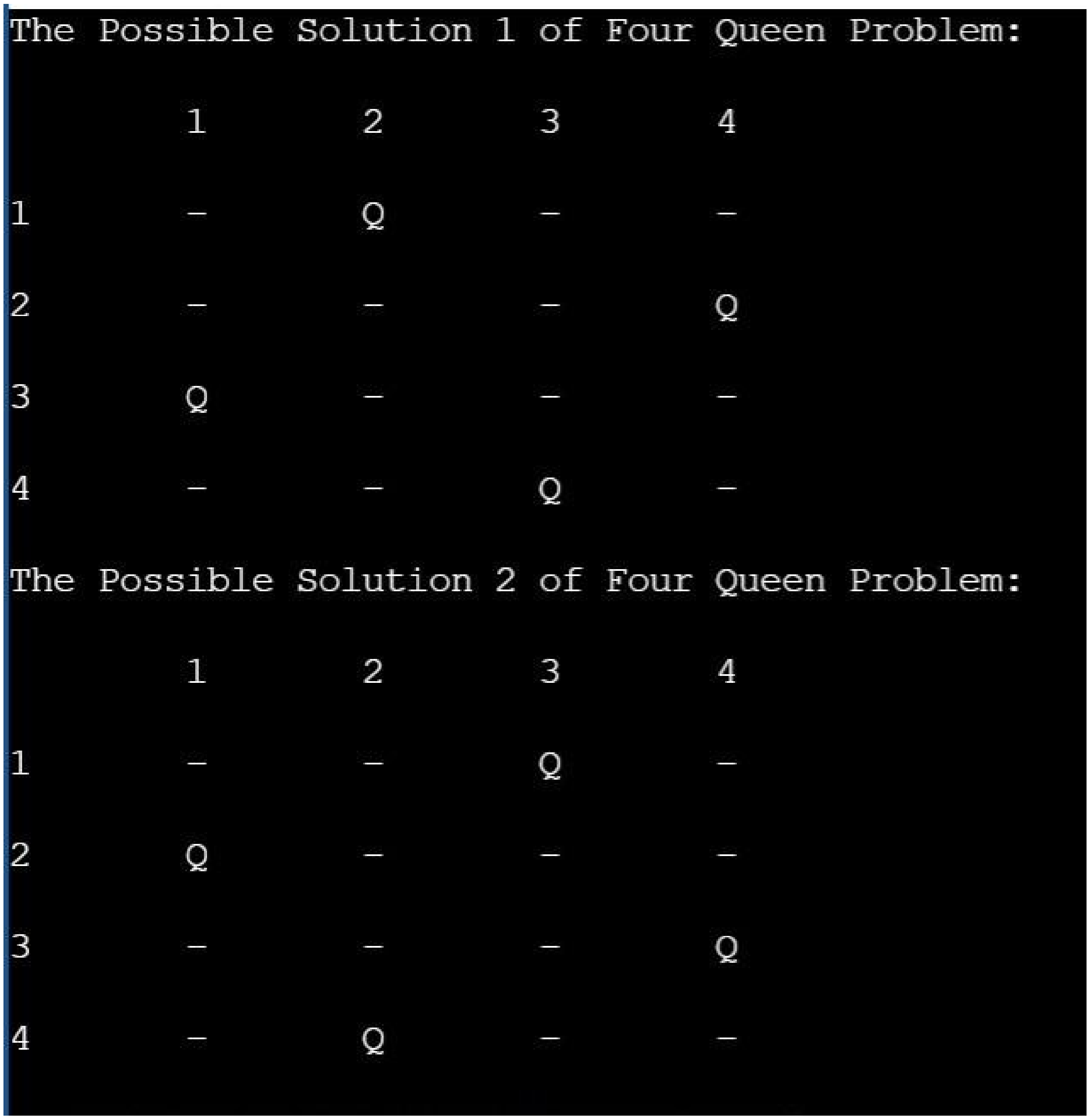
A.) (i)<u>CODE</u>:-

```
#include<stdio.h>

#include<math.h>

#include<stdlib.h>

 int a[20],count;

 int main()
 {

 int n=4,i,j;

 void fourqueen(int row,int n);

 printf(" Four Queens Problem Using Backtracking -");

 fourqueen(1,n);

 return 0;

}

//function for printing the solution

void printfourqueensolution(int n)

{

 int i,j;
 printf("\n\nThe Possible Solution %d of Four Queen Problem:\n\n",++count);

 for(i=1;i<=n;++i)

  printf("\t%d",i);

 for(i=1;i<=n;++i)

 {

  printf("\n\n%d",i);

  for(j=1;j<=n;++j) //for nxn board

  {

   if(a[i]==j)

    printf("\tQ"); //queen at i,j position

   else

    printf("\t-"); //empty slot
    }
    }

}

//function to check conflicts If no conflict for desired position returns 1 otherwise returns 0/

int placequeen(int row,int column)

{

 int i;

 for(i=1;i<=row-1;++i) {

 //checking column and diagonal conflicts
```

```
  if(a[i]==column)

   return 0;

  else

   if(abs(a[i]-column)==abs(i-row))

    return 0;

 }
 return 1; //no conflicts

}

//function to check for proper positioning of queen

void fourqueen(int row,int n)
{

 int column;

 for(column=1;column<=n;++column)
 {

  if(placequeen(row,column))
  {

   a[row]=column; //no conflicts so place queen

   if(row==n) //dead end

    printfourqueensolution(n); //printing the board configuration

   else //try queen with next position

    fourqueen(row+1,n);

  }
  }
```

}

(ii) <u>SAMPLE OUTPUT</u>:-



```
The Possible Solution 1 of Four Queen Problem:

        1       2       3       4

1       -       Q       -       -

2       -       -       -       Q

3       Q       -       -       -

4       -       -       Q       -

The Possible Solution 2 of Four Queen Problem:

        1       2       3       4

1       -       -       Q       -

2       Q       -       -       -

3       -       -       -       Q

4       -       Q       -       -
```