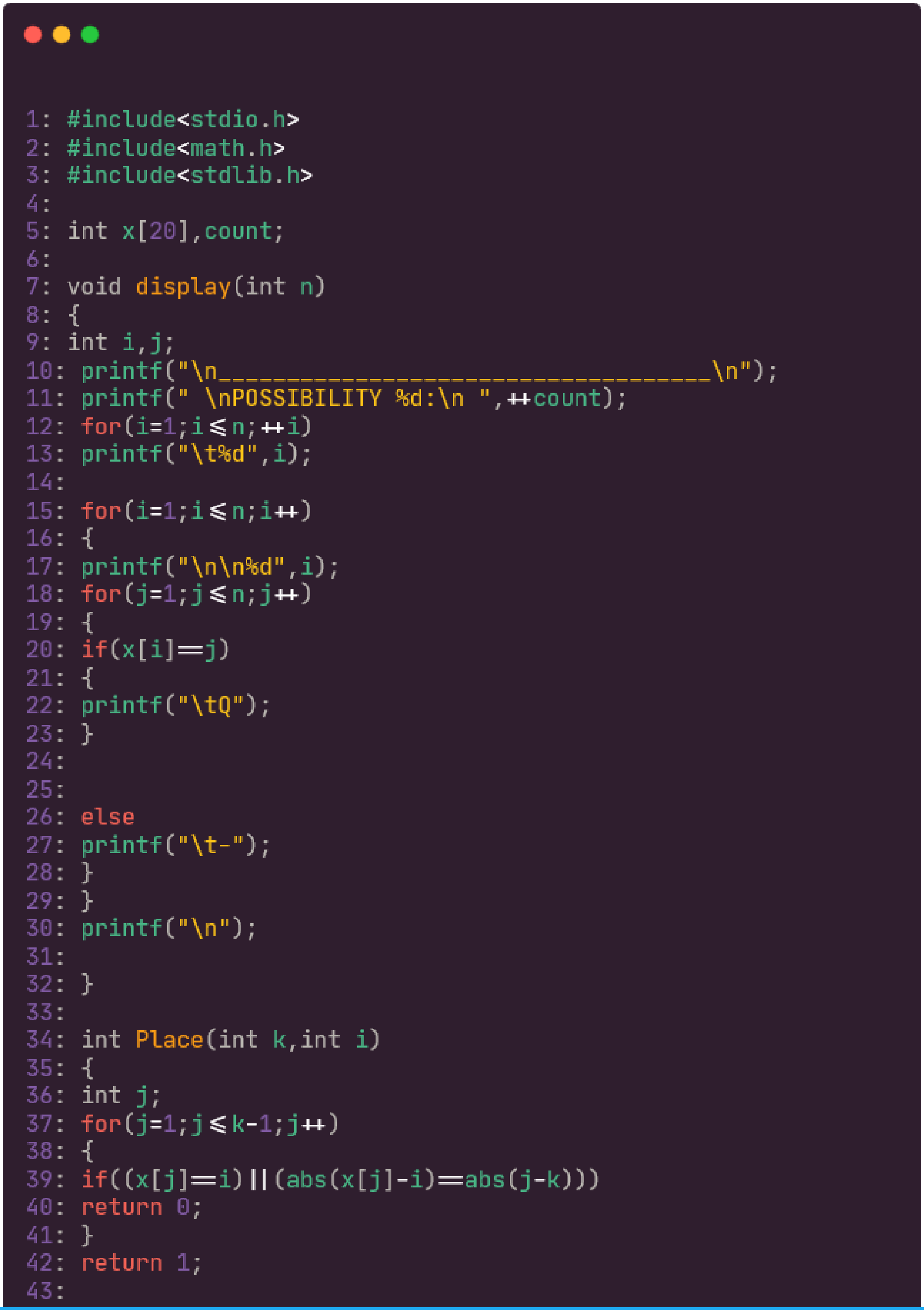
Danyl Fernandes
2020012004 (72)

21-04-2021

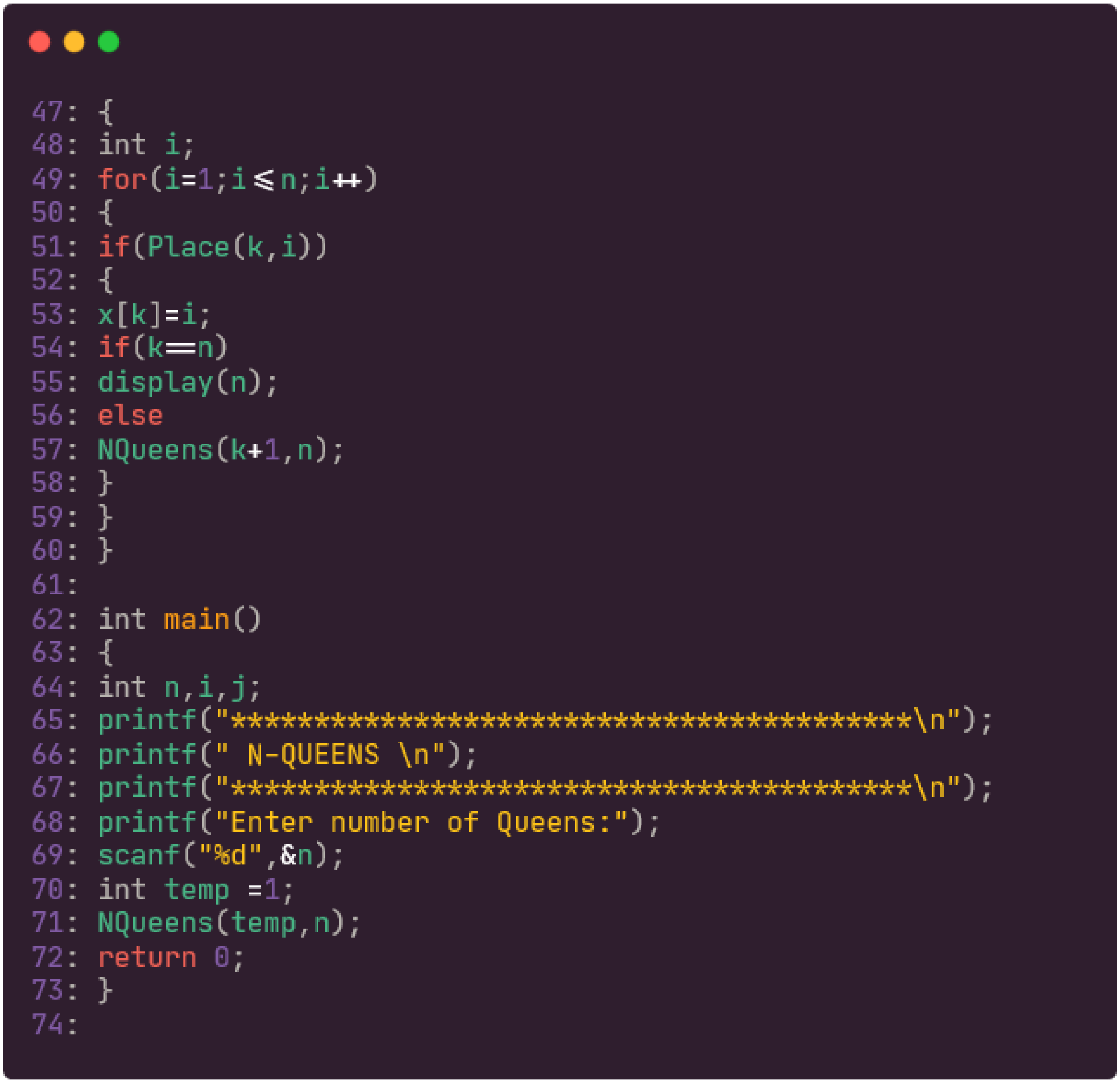
AOA Experiment 7

Aim:

To implement & analyze N-Queen Problem using Backtracking approach:

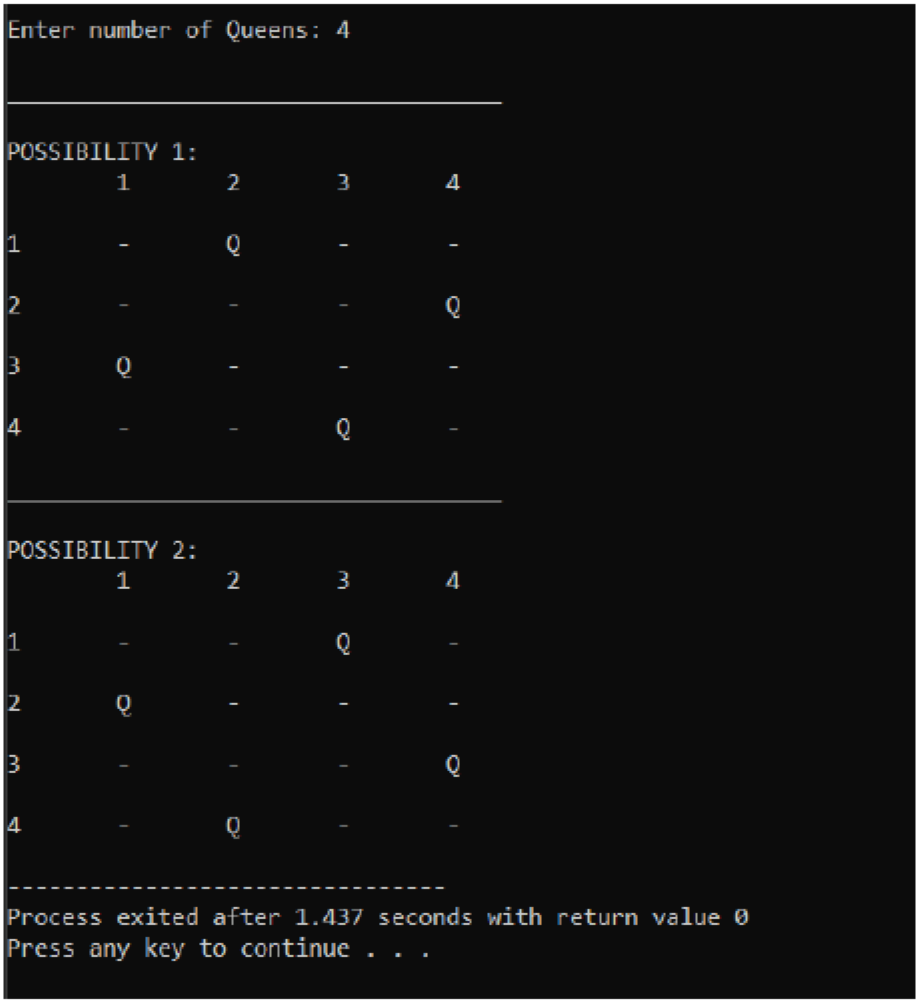


```c
#include<stdio.h>
#include<math.h>
#include<stdlib.h>

int x[20],count;

void display(int n)
{
int i,j;
printf("\n____________________________________\n");
printf(" \nPOSSIBILITY %d:\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++)
{
if(x[i]==j)
{
printf("\tQ");
}


else
printf("\t-");
}
}
printf("\n");

}

int Place(int k,int i)
{
int j;
for(j=1;j<=k-1;j++)
{
if((x[j]==i)||(abs(x[j]-i)==abs(j-k)))
return 0;
}
return 1;

```



```
{
int i;
for(i=1;i<=n;i++)
{
if(Place(k,i))
{
x[k]=i;
if(k==n)
display(n);
else
NQueens(k+1,n);
}
}
}

int main()
{
int n,i,j;
printf("*****************************************\n");
printf(" N-QUEENS \n");
printf("*****************************************\n");
printf("Enter number of Queens:");
scanf("%d",&n);
int temp =1;
NQueens(temp,n);
return 0;
}
```

Output:



```
Enter number of Queens: 4

____________________________________

POSSIBILITY 1:
        1       2       3       4

1       -       Q       -       -

2       -       -       -       Q

3       Q       -       -       -

4       -       -       Q       -

____________________________________

POSSIBILITY 2:
        1       2       3       4

1       -       -       Q       -

2       Q       -       -       -

3       -       -       -       Q

4       -       Q       -       -

--------------------------------
Process exited after 1.437 seconds with return value 0
Press any key to continue . . .
```

Darryl Fernandes
2020012004 (72)

Exp 07

Theory :

- It is a famous chess puzzle based on Combinational logic. The efficient solution to this problem is given by the backtracking Strategy.

- It is a classical example of backtracking algorithm

Problem description!

- Place n-queens on an $n \times n$ chessboard such that none of them con attach any other using standard chess queen's moves.

- This implies that no two 2 queens placed at position (i,j) & (k,l) where i & k are the rows indides & j & k ale the coloumn indices, then

  $i \neq k$ (not same low)
  $j \neq k$ (not same coloumn) &
  $|i-k| \neq |j-l|$ { No same diagonal
  where $i, j, k, l \in \{1, 2, 3 \dots N\}$ }

Algorithm :

```
Place (k,i) {
    for j=1 to k-1 do
        if (x[j] = i)
```

Daryl Fernandes
2020012004(72)

```
        (Abs (x [j] - i) = Abs (j - k)))
        then return false,
    return true;
    }

    NQueens (k, n)  {
      for i = 1  to  n {
           if  Place (k, i) then {
               x[k] = i ;
               if (k = n) the write (x[1:n]);
               else NQueens (k + 1, n);
           }
      }
    }
```

Analysis :

- This algorithm takes O(N) time as it iterates through our array everytime, for each invocation of Place method, the loop runs for O(N) time.

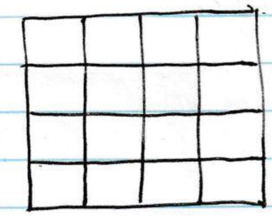
- In each iteration of this loop, there is a recursive call O(N)

- Therefore the run time
  $T(N) = O(N^2) + N * T(N-1)$

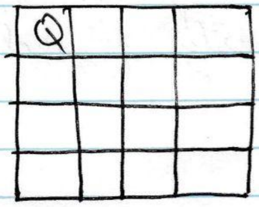
After solving this, it can be reduced to O(N!). The best case occurs if you find your solution before finding all possibilities.

Danyl Fernandes
2020012004 (72)

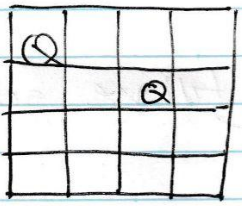
Example :



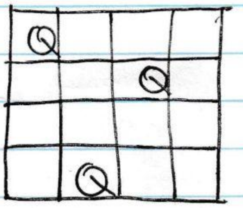
We could start by placing 1st queen in the first row



Now, the second step is to place the second queen in a safe position. We would place the queen in 2nd row, as we cant in frist



We would place the third queen in some safe position in the third row



There is no safe position to place the last queen, so we will change the positio of previous queen ie backtracking & changing previous decision.

Also, there is no other position, where, can place the 3rd queen, so we will go back 1 more step & change position of 2nd queen

Danyl Fernandes
2020012004 (72)

Q			
			Q

→ position changed

removed

We would place the third queen in a saf position other than previously placed positi in 3rd row

Q			
			Q
	Q		

The process continues , till we get the final solution

	Q		
			Q
Q			
		Q	

Conclusion : We successfully able to analyse and implement N queens algorithm using backtracking approach.