DAA – CSE2012

Lab-07

N QUEEN PROBLEM

(PPS N Queen Back-tracking)

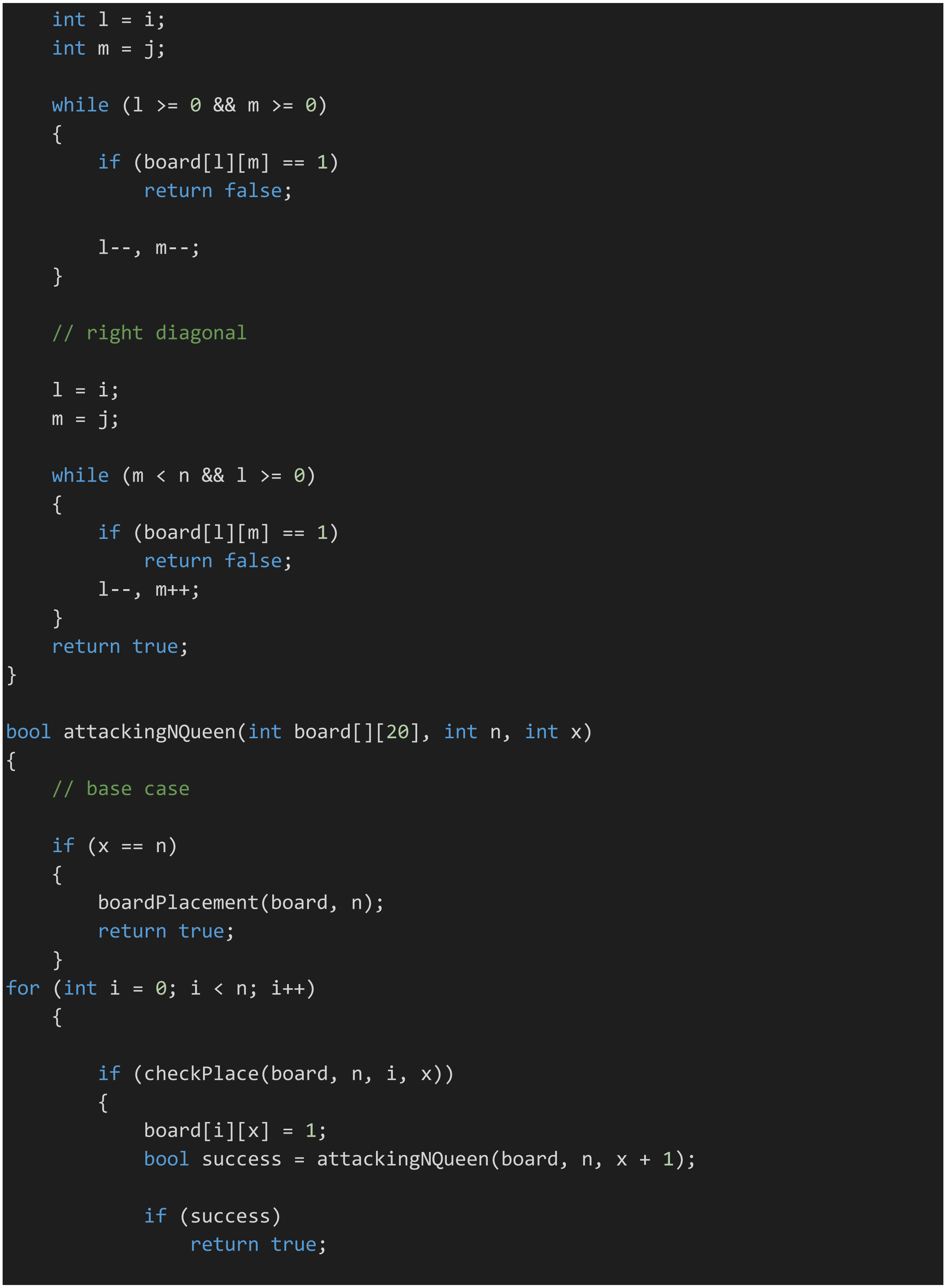
Name: Preyash Registration Number: 20BPS1022

Algorithm:

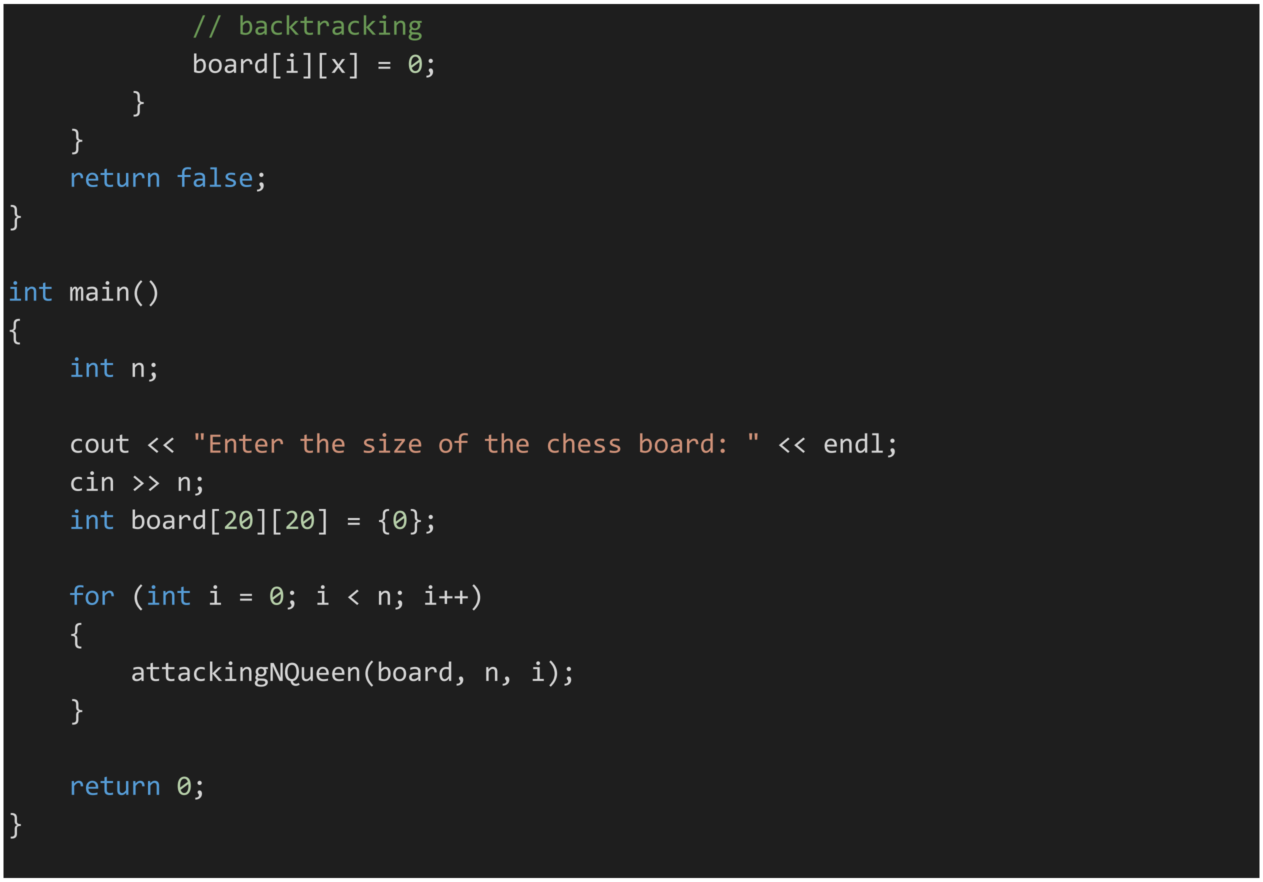
1. Start
2. First we are checking the row match
3. Then we are checking for the left diagonal match
4. Then we are checking for the right diagonal match
5. Stop

Code:

```cpp
#include <iostream>
using namespace std;

void boardPlacement(int board[][20], int n)
{
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            cout << board[i][j] << " ";
        }
        cout << endl;
    }
}

bool checkPlace(int board[][20], int n, int i, int j)
{
    // row check

    for (int k = 0; k < j; k++)
    {
        if (board[i][k] == 1)
            return false;
    }

    // left diagonal
```

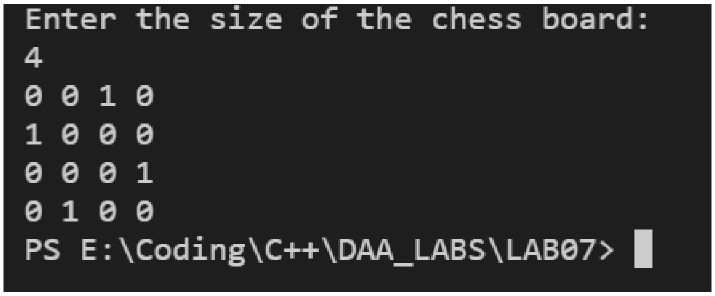


```
    int l = i;
    int m = j;

    while (l >= 0 && m >= 0)
    {
        if (board[l][m] == 1)
            return false;

        l--, m--;
    }

    // right diagonal

    l = i;
    m = j;

    while (m < n && l >= 0)
    {
        if (board[l][m] == 1)
            return false;
        l--, m++;
    }
    return true;
}

bool attackingNQueen(int board[][20], int n, int x)
{
    // base case

    if (x == n)
    {
        boardPlacement(board, n);
        return true;
    }
for (int i = 0; i < n; i++)
    {

        if (checkPlace(board, n, i, x))
        {
            board[i][x] = 1;
            bool success = attackingNQueen(board, n, x + 1);

            if (success)
                return true;
```



```
            // backtracking
            board[i][x] = 0;
        }
    }
    return false;
}

int main()
{
    int n;

    cout << "Enter the size of the chess board: " << endl;
    cin >> n;
    int board[20][20] = {0};

    for (int i = 0; i < n; i++)
    {
        attackingNQueen(board, n, i);
    }

    return 0;
}
```

Output:



```
Enter the size of the chess board:
4
0 0 1 0
1 0 0 0
0 0 0 1
0 1 0 0
PS E:\Coding\C++\DAA_LABS\LAB07>
```