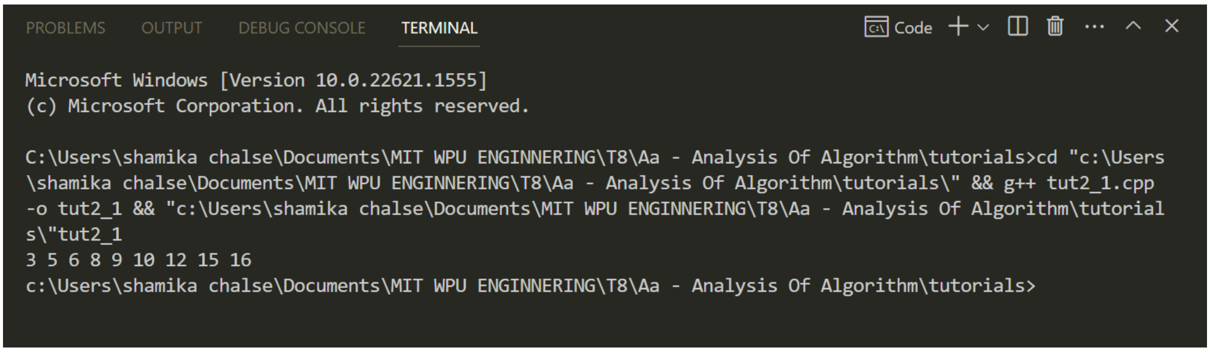
Tut2 AA

Name   :- Shamika Chalse
Roll No :- 53

```
#include<iostream>
#include <vector>
#include <cmath>
using namespace std;

void MQuickSort(vector<int> &arr, int l, int h, int pivot){
    if(l<h){
        int i,z,j,v;
        int pivot1,pivot2,k,countLess,sumLess,countLarger,sumLarger;
        bool N=true;
        i=z=l;
        j=v=h;
        pivot1=pivot2=countLess=countLarger=sumLess=sumLarger=0;
        k=arr[h];
        while(i<=j){
            if(arr[i]<=pivot){
                countLess++;
                sumLess+=arr[i];
                if(N == true && k >= (pivot - arr[i])){
                    k=pivot-arr[i];
                }
                else{
                    N=false;
                }
                i++;
            }
            else{
                countLarger++;
                sumLarger+=arr[i];
                swap(arr[i],arr[j]);
                j--;
            }
        }

        if(countLess!=0){
            pivot1=floor(sumLess/countLess);
            if(N!=true){
                MQuickSort(arr,z,i-1,pivot1);
            }
        }
        if(countLarger!=0){
            pivot2=floor(sumLarger/countLarger);
            MQuickSort(arr,i,v,pivot2);
        }
    }
}


int main(){
    int arr[] = { 10,16,8,12,15,6,3,9,5 };
    vector<int> v(arr, arr + sizeof(arr) / sizeof(int));
    MQuickSort(v,0,v.size()-1,v.size()-1);
    for(int x: v)
        cout<<x<<" ";
    return 0;
}
```

Output:-



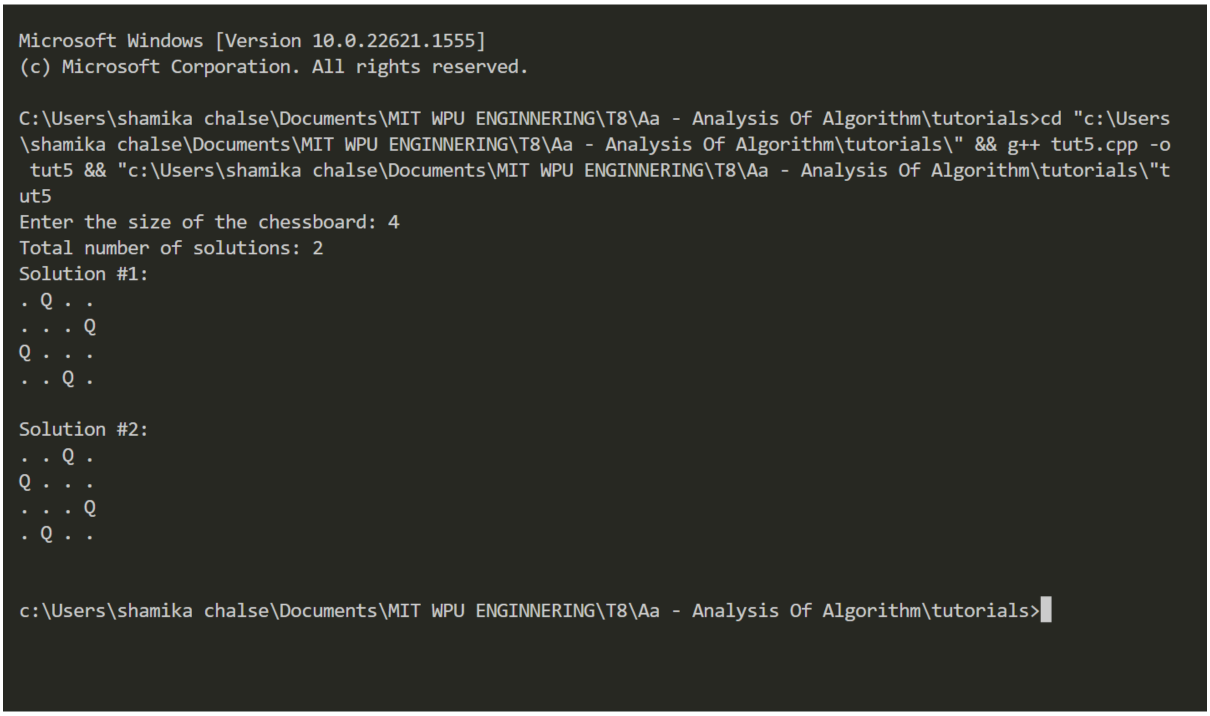
```
Microsoft Windows [Version 10.0.22621.1555]
(c) Microsoft Corporation. All rights reserved.

C:\Users\shamika chalse\Documents\MIT WPU ENGINNERING\T8\Aa - Analysis Of Algorithm\tutorials>cd "c:\Users
\shamika chalse\Documents\MIT WPU ENGINNERING\T8\Aa - Analysis Of Algorithm\tutorials\" && g++ tut2_1.cpp
-o tut2_1 && "c:\Users\shamika chalse\Documents\MIT WPU ENGINNERING\T8\Aa - Analysis Of Algorithm\tutorial
s\"tut2_1
3 5 6 8 9 10 12 15 16
c:\Users\shamika chalse\Documents\MIT WPU ENGINNERING\T8\Aa - Analysis Of Algorithm\tutorials>
```

Tut5 AA

Name :- Shamika Chalse
Roll No :- 53

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

bool isSafe(vector<int>& board, int row, int col) {
    // Check if the current queen can be placed in this position
    for (int i = 0; i < row; i++) {
        // Check if there is a queen in the same column or diagonal as the current queen
        if (board[i] == col || board[i] - i == col - row || board[i] + i == col + row) {
            return false;
        }
    }
    return true;
}

void solveNQueens(vector<int>& board, int row, vector<vector<int>>& solutions) {
    int n = board.size();
    if (row == n) {
        // All queens have been placed, so add the solution to the list of solutions
        solutions.push_back(board);
        return;
    }
    for (int col = 0; col < n; col++) {
        if (isSafe(board, row, col)) {
            // Place the queen in this position and move on to the next row
            board[row] = col;
            solveNQueens(board, row + 1, solutions);
            // Remove the queen from this position and backtrack to try the next position in the same row
            board[row] = -1;
        }
    }
}

vector<vector<int>> solveNQueens(int n) {
    vector<vector<int>> solutions;
    vector<int> board(n, -1);
    solveNQueens(board, 0, solutions);
    return solutions;
}

int main() {
    int n;
    cout << "Enter the size of the chessboard: ";
    cin >> n;
    vector<vector<int>> solutions = solveNQueens(n);
    cout << "Total number of solutions: " << solutions.size() << endl;
    for (int i = 0; i < solutions.size(); i++) {
        cout << "Solution #" << i + 1 << ":" << endl;
        for (int j = 0; j < n; j++) {
            for (int k = 0; k < n; k++) {
                cout << (solutions[i][j] == k ? "Q " : ". ");
            }
            cout << endl;
        }
        cout << endl;
    }
    return 0;
}
```

Output:-



```
Microsoft Windows [Version 10.0.22621.1555]
(c) Microsoft Corporation. All rights reserved.

C:\Users\shamika chalse\Documents\MIT WPU ENGINNERING\T8\Aa - Analysis Of Algorithm\tutorials>cd "c:\Users
\shamika chalse\Documents\MIT WPU ENGINNERING\T8\Aa - Analysis Of Algorithm\tutorials\" && g++ tut5.cpp -o
 tut5 && "c:\Users\shamika chalse\Documents\MIT WPU ENGINNERING\T8\Aa - Analysis Of Algorithm\tutorials\"t
ut5
Enter the size of the chessboard: 4
Total number of solutions: 2
Solution #1:
. Q . .
. . . Q
Q . . .
. . Q .

Solution #2:
. . Q .
Q . . .
. . . Q
. Q . .


c:\Users\shamika chalse\Documents\MIT WPU ENGINNERING\T8\Aa - Analysis Of Algorithm\tutorials>
```