Name: Ashmit Thawait

Roll No: 102203790

Group: 2CO17

Lab Assignment 4

Q1. N-queen Problem

```cpp
#include <bits/stdc++.h>
using namespace std;
#define N 4

int ld[30] = { 0 };
int rd[30] = { 0 };
int cl[30] = { 0 };

void printSolution(int board[N][N])
{
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++)
            cout << " " << (board[i][j]==1?"Q":".") << " ";
        cout << endl;
    }
}

bool solveNQUtil(int board[N][N], int col)
{
    if (col >= N)
        return true;

    for (int i = 0; i < N; i++) {

        if ((ld[i - col + N - 1] != 1 && rd[i + col] != 1) && cl[i] != 1) {

            board[i][col] = 1;
            ld[i - col + N - 1] = rd[i + col] = cl[i] = 1;

            if (solveNQUtil(board, col + 1))
                return true;

            board[i][col] = 0;
            ld[i - col + N - 1] = rd[i + col] = cl[i] = 0;
        }
    }
    return false;
}
```

```cpp
bool solveNQ()
{
    int board[N][N] = { { 0, 0, 0, 0 },
                        { 0, 0, 0, 0 },
                        { 0, 0, 0, 0 },
                        { 0, 0, 0, 0 } };

    if (solveNQUtil(board, 0) == false) {
        cout << "Solution does not exist";
        return false;
    }

    printSolution(board);
    return true;
}

int main()
{
    solveNQ();
    return 0;
}
```

```
PS D:\DAA Assignments\Assignment 4> g++ n_queen.cpp
PS D:\DAA Assignments\Assignment 4> ./a.exe
 .  .  Q  .
 Q  .  .  .
 .  .  .  Q
 .  Q  .  .
```

Q2. Sum of subsets

```cpp
#include <bits/stdc++.h>
using namespace std;

bool flag = 0;
void PrintSubsetSum(int i, int n, int set[], int targetSum, vector<int>& subset)
{
    if (targetSum == 0) {

        flag = 1;
        cout << "[ ";
        for (int i = 0; i < subset.size(); i++) {
            cout << subset[i] << " ";
        }
        cout << "]";
        return;
    }

    if (i == n) {
        return;
    }

    PrintSubsetSum(i + 1, n, set, targetSum, subset);

    if (set[i] <= targetSum) {

        subset.push_back(set[i]);

        PrintSubsetSum(i + 1, n, set, targetSum - set[i],
                       subset);

        subset.pop_back();
    }
}
```

```cpp
int main()
{
    int set[] = { 1, 2, 1 };
    int sum = 3;
    int n = sizeof(set) / sizeof(set[0]);
    vector<int> subset;
    cout << "Output 1:" << endl;
    PrintSubsetSum(0, n, set, sum, subset);
    cout << endl;
    flag = 0;

    int set2[] = { 3, 34, 4, 12, 5, 2 };
    int sum2 = 30;
    int n2 = sizeof(set) / sizeof(set[0]);
    vector<int> subset2;
    cout << "Output 2:" << endl;
    PrintSubsetSum(0, n2, set2, sum2, subset2);
    if (!flag) {
        cout << "There is no such subset";
    }

    return 0;
}
```

```
PS D:\DAA Assignments\Assignment 4> g++ SumOfSubsets.cpp
PS D:\DAA Assignments\Assignment 4> ./a.exe
Output 1:
[ 2 1 ][ 1 2 ]
Output 2:
There is no such subset
PS D:\DAA Assignments\Assignment 4>
```

Q3. Graph coloring

```
#include <bits/stdc++.h>
using namespace std;
#define V 4

void printSolution(int color[]);

bool isSafe(int v, bool graph[V][V], int color[], int c)
{
    for (int i = 0; i < V; i++)
        if (graph[v][i] && c == color[i])
            return false;

    return true;
}

bool graphColoringUtil(bool graph[V][V], int m, int color[],
                       int v)
{
    if (v == V)
        return true;

    for (int c = 1; c <= m; c++) {

        if (isSafe(v, graph, color, c)) {
            color[v] = c;

            if (graphColoringUtil(graph, m, color, v + 1)
                == true)
                return true;

            color[v] = 0;
        }
    }
    return false;
}
```

```cpp
bool graphColoring(bool graph[V][V], int m)
{
    int color[V];
    for (int i = 0; i < V; i++)
        color[i] = 0;

    if (graphColoringUtil(graph, m, color, 0) == false) {
        cout << "Solution does not exist";
        return false;
    }

    printSolution(color);
    return true;
}

void printSolution(int color[])
{
    cout << "Solution Exists:"
        << " Following are the assigned colors"
        << "\n";
    for (int i = 0; i < V; i++)
        cout << " " << color[i] << " ";

    cout << "\n";
}

int main()
{
    bool graph[V][V] = {
        { 0, 1, 1, 1 },
        { 1, 0, 1, 0 },
        { 1, 1, 0, 1 },
        { 1, 0, 1, 0 },
    };

    int m = 3;
    graphColoring(graph, m);
    return 0;
}
```

```
PS D:\DAA Assignments\Assignment 4> g++ .\GraphColouring.cpp
PS D:\DAA Assignments\Assignment 4> ./a.exe
Solution Exists: Following are the assigned colors
 1  2  3  2
PS D:\DAA Assignments\Assignment 4>
```