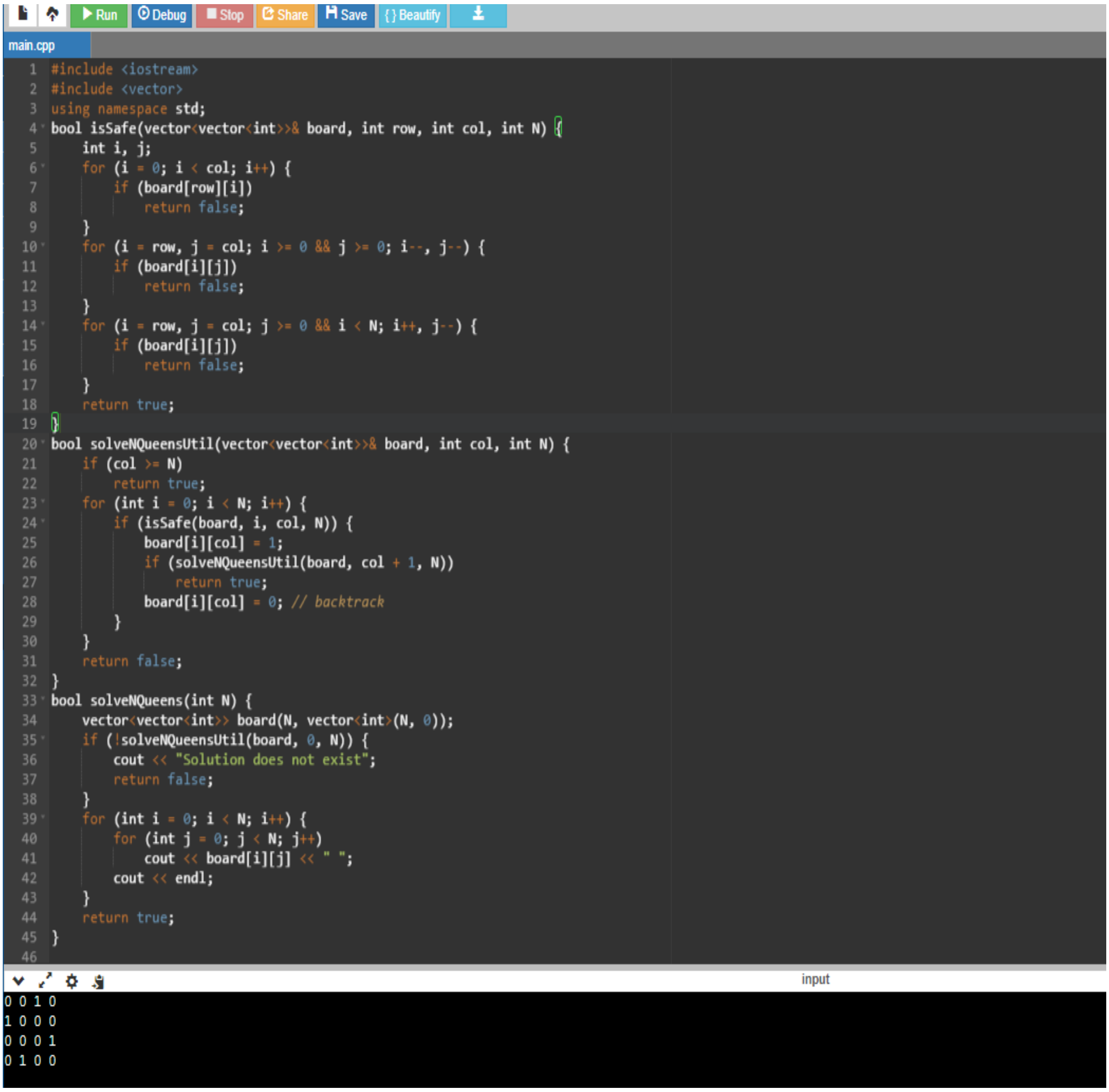
DAA LAB ASSIGNMENT 4

NAME->Chandranshu Bhardwaj

ROLL NO.->102203797

Q1.) N Queen?

Ans->

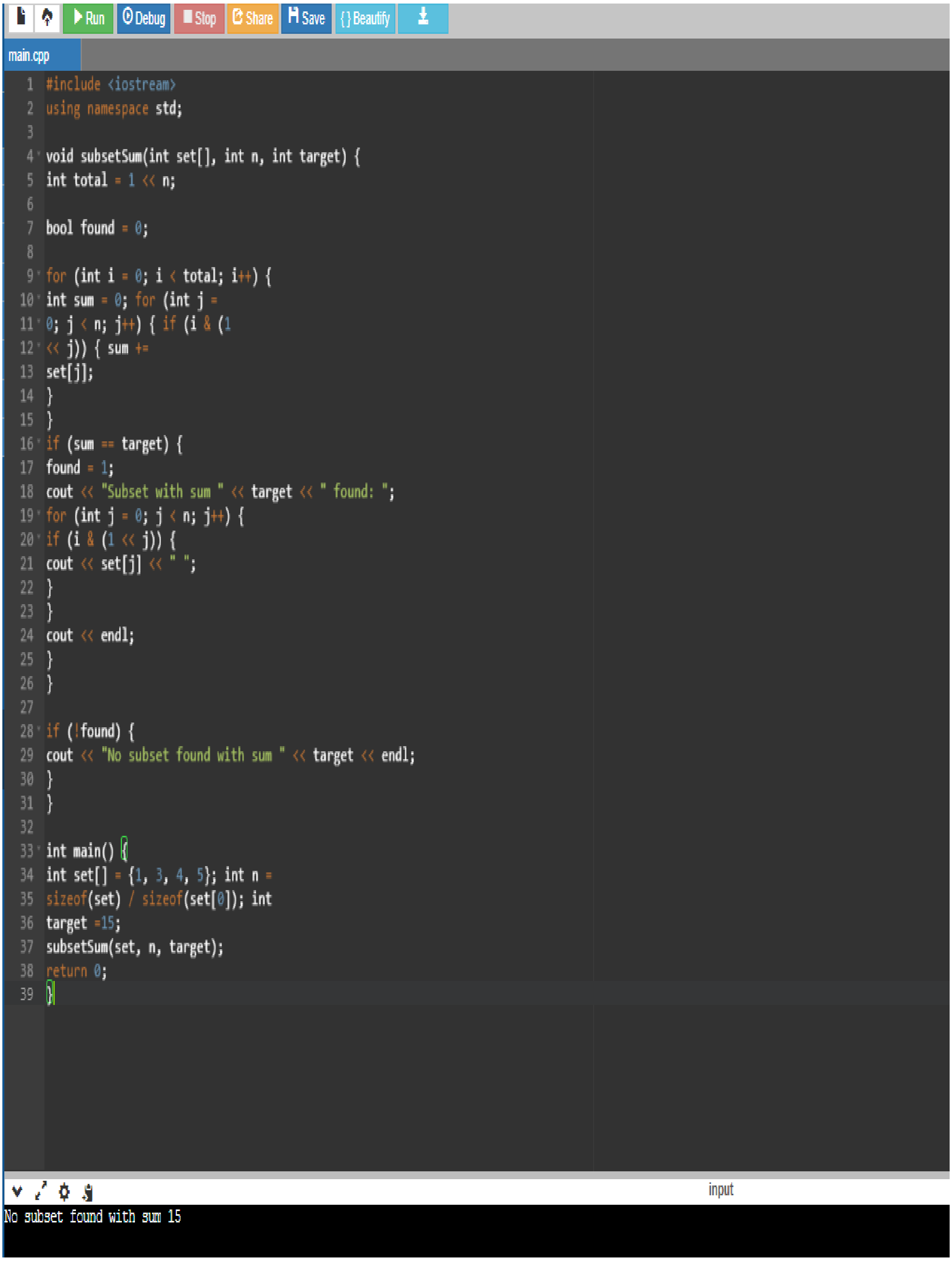


```cpp
#include <iostream>
#include <vector>
using namespace std;
bool isSafe(vector<vector<int>>& board, int row, int col, int N) {
    int i, j;
    for (i = 0; i < col; i++) {
        if (board[row][i])
            return false;
    }
    for (i = row, j = col; i >= 0 && j >= 0; i--, j--) {
        if (board[i][j])
            return false;
    }
    for (i = row, j = col; j >= 0 && i < N; i++, j--) {
        if (board[i][j])
            return false;
    }
    return true;
}
bool solveNQueensUtil(vector<vector<int>>& board, int col, int N) {
    if (col >= N)
        return true;
    for (int i = 0; i < N; i++) {
        if (isSafe(board, i, col, N)) {
            board[i][col] = 1;
            if (solveNQueensUtil(board, col + 1, N))
                return true;
            board[i][col] = 0; // backtrack
        }
    }
    return false;
}
bool solveNQueens(int N) {
    vector<vector<int>> board(N, vector<int>(N, 0));
    if (!solveNQueensUtil(board, 0, N)) {
        cout << "Solution does not exist";
        return false;
    }
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++)
            cout << board[i][j] << " ";
        cout << endl;
    }
    return true;
}

```

```
0 0 1 0
1 0 0 0
0 0 0 1
0 1 0 0
```

Q2.) Sum of Subset?

Ans->

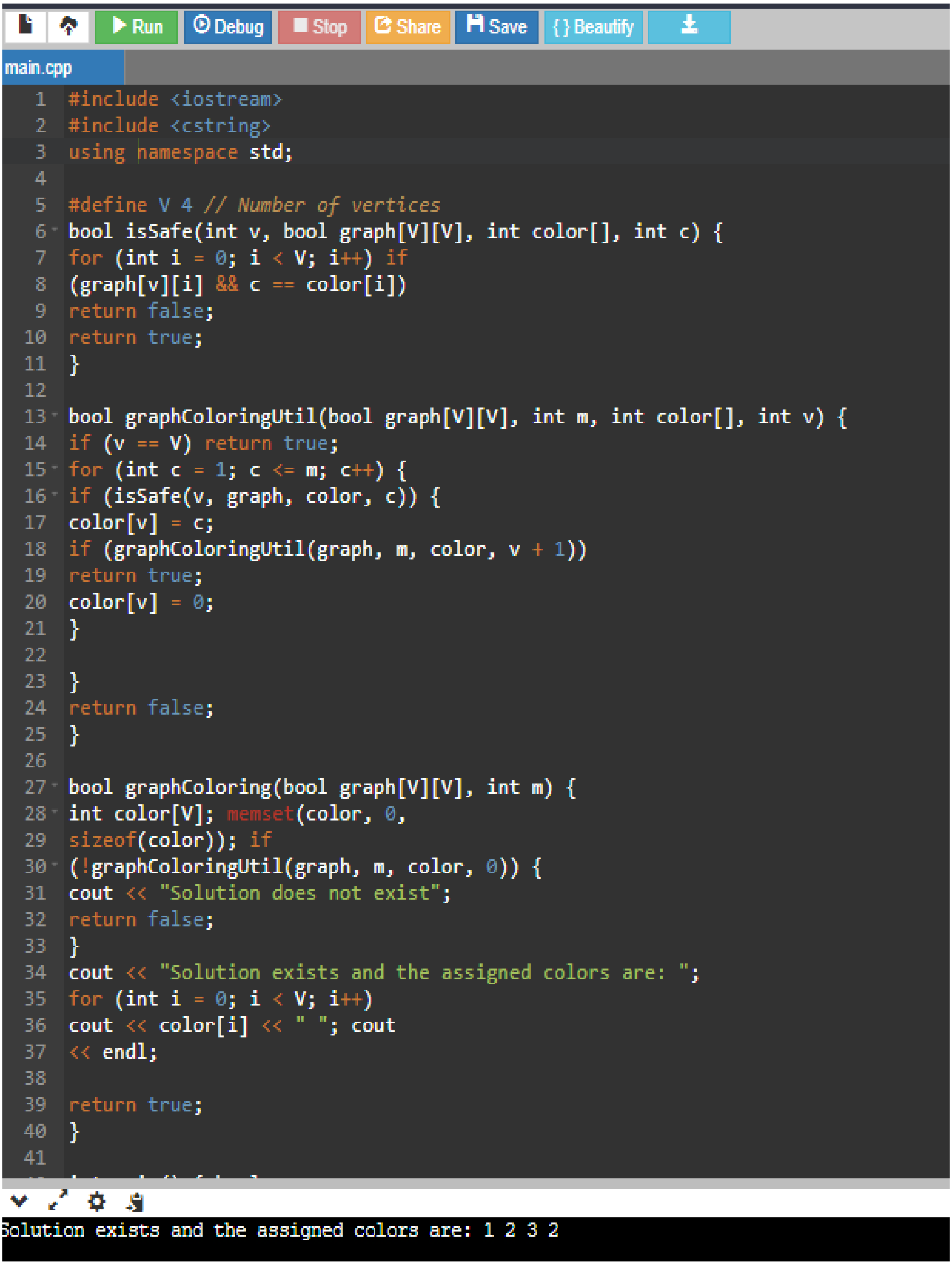


```
#include <iostream>
using namespace std;

void subsetSum(int set[], int n, int target) {
int total = 1 << n;

bool found = 0;

for (int i = 0; i < total; i++) {
int sum = 0; for (int j =
0; j < n; j++) { if (i & (1
<< j)) { sum +=
set[j];
}
}
if (sum == target) {
found = 1;
cout << "Subset with sum " << target << " found: ";
for (int j = 0; j < n; j++) {
if (i & (1 << j)) {
cout << set[j] << " ";
}
}
cout << endl;
}
}

if (!found) {
cout << "No subset found with sum " << target << endl;
}
}

int main() {
int set[] = {1, 3, 4, 5}; int n =
sizeof(set) / sizeof(set[0]); int
target =15;
subsetSum(set, n, target);
return 0;
}
```

```
No subset found with sum 15
```

Q3.)Graph Coloring?

Ans->



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

#define V 4 // Number of vertices
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))
return true;
color[v] = 0;
}

}
return false;
}

bool graphColoring(bool graph[V][V], int m) {
int color[V]; memset(color, 0,
sizeof(color)); if
(!graphColoringUtil(graph, m, color, 0)) {
cout << "Solution does not exist";
return false;
}
cout << "Solution exists and the assigned colors are: ";
for (int i = 0; i < V; i++)
cout << color[i] << " "; cout
<< endl;

return true;
}

```

```
Solution exists and the assigned colors are: 1 2 3 2
```