

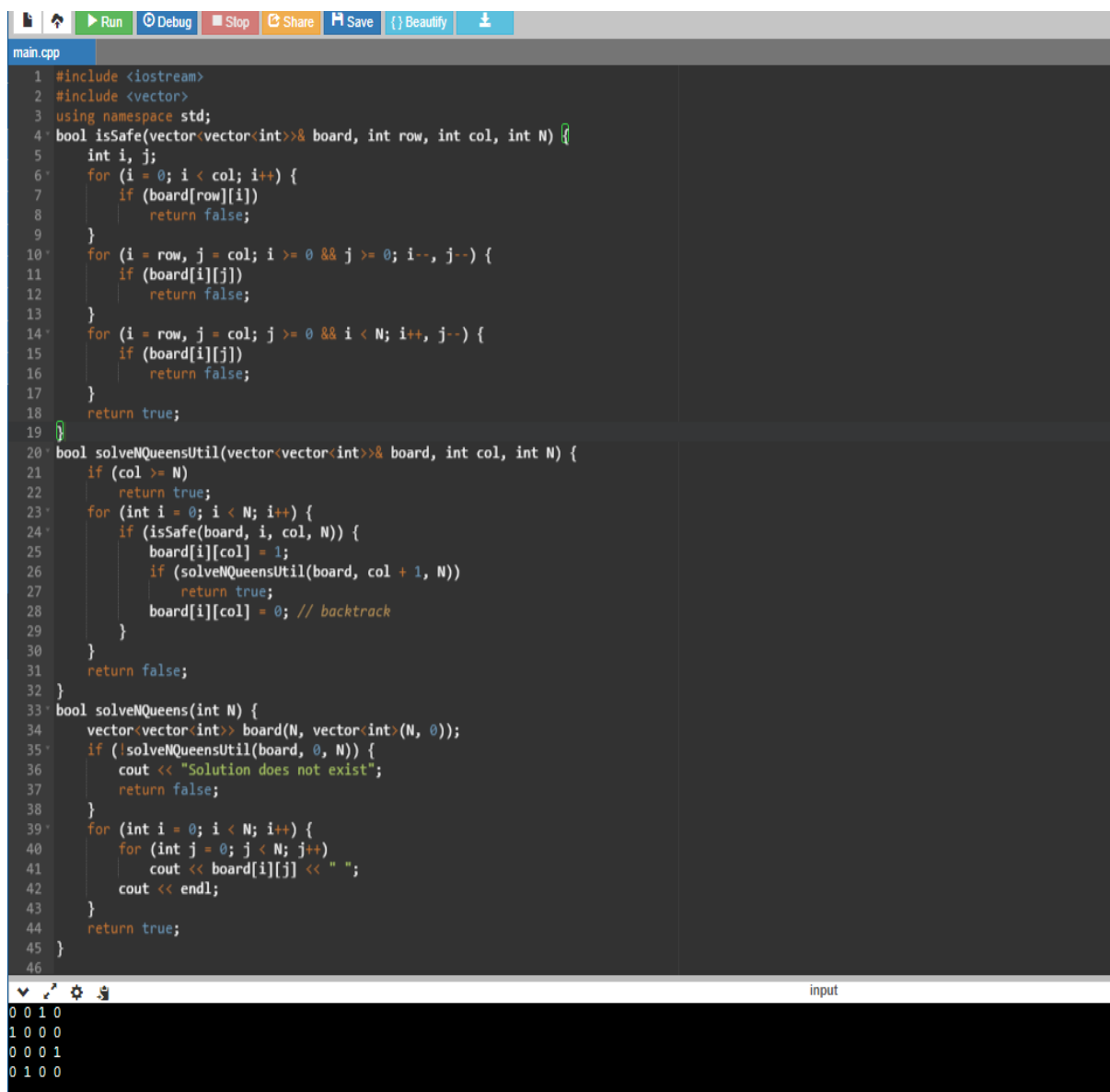
DAA LAB ASSIGNMENT 4

NAME->Chandranshu Bhardwaj

ROLL NO.->102203797

Q1.) N Queen?

Ans->



```
main.cpp
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4 bool isSafe(vector<vector<int>>& board, int row, int col, int N) {
5     int i, j;
6     for (i = 0; i < col; i++) {
7         if (board[row][i])
8             return false;
9     }
10    for (i = row, j = col; i >= 0 && j >= 0; i--, j--) {
11        if (board[i][j])
12            return false;
13    }
14    for (i = row, j = col; j >= 0 && i < N; i++, j--) {
15        if (board[i][j])
16            return false;
17    }
18    return true;
19 }
20 bool solveNQueensUtil(vector<vector<int>>& board, int col, int N) {
21     if (col >= N)
22         return true;
23     for (int i = 0; i < N; i++) {
24         if (isSafe(board, i, col, N)) {
25             board[i][col] = 1;
26             if (solveNQueensUtil(board, col + 1, N))
27                 return true;
28             board[i][col] = 0; // backtrack
29         }
30     }
31     return false;
32 }
33 bool solveNQueens(int N) {
34     vector<vector<int>> board(N, vector<int>(N, 0));
35     if (!solveNQueensUtil(board, 0, N)) {
36         cout << "Solution does not exist";
37         return false;
38     }
39     for (int i = 0; i < N; i++) {
40         for (int j = 0; j < N; j++)
41             cout << board[i][j] << " ";
42         cout << endl;
43     }
44     return true;
45 }
46
```

input

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

Q2.) Sum of Subset?

Ans->



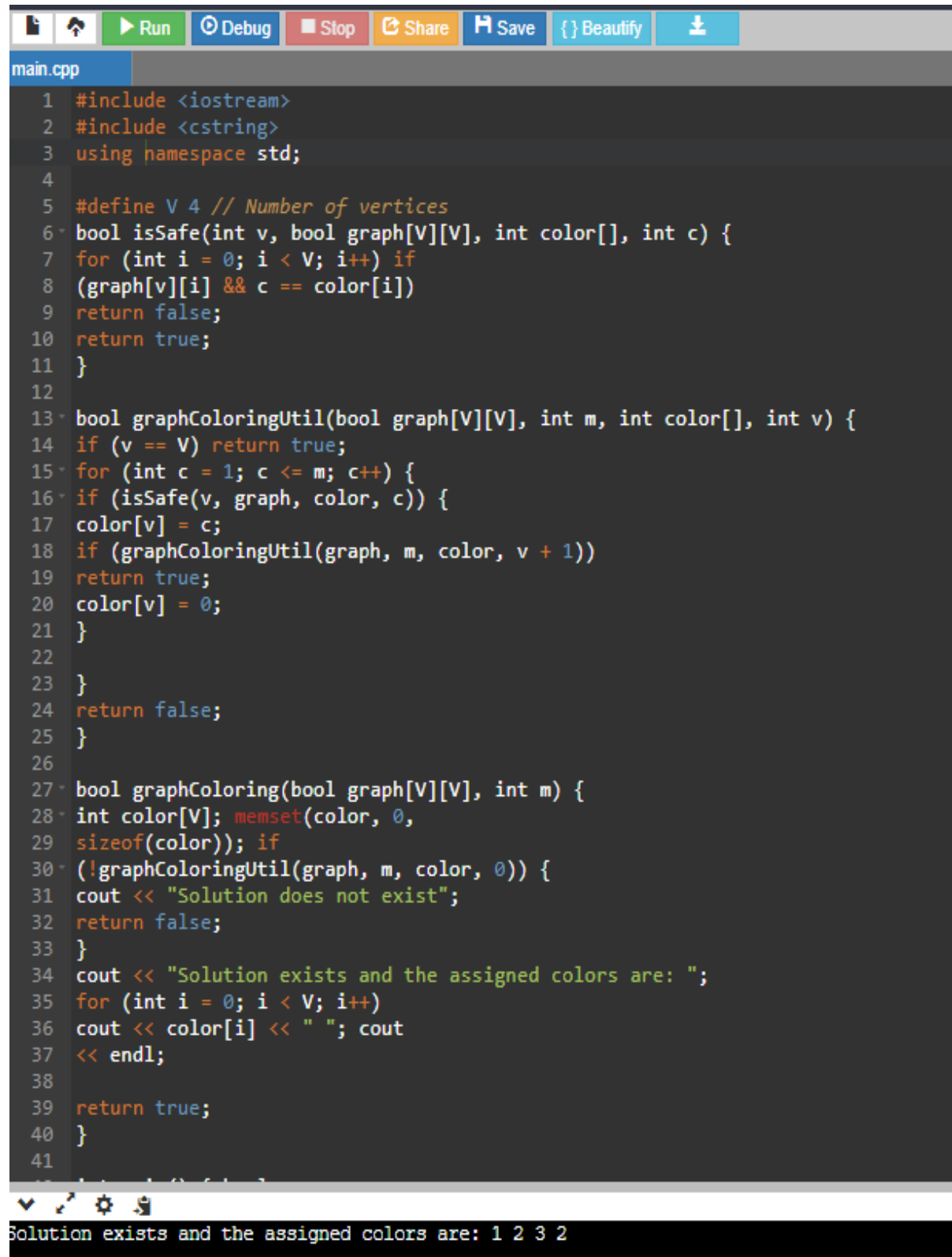
```
1 #include <iostream>
2 using namespace std;
3
4 void subsetSum(int set[], int n, int target) {
5     int total = 1 << n;
6
7     bool found = 0;
8
9     for (int i = 0; i < total; i++) {
10         int sum = 0; for (int j =
11         0; j < n; j++) { if (i & (1
12         << j)) { sum +=
13         set[j];
14     }
15 }
16 if (sum == target) {
17     found = 1;
18     cout << "Subset with sum " << target << " found: ";
19     for (int j = 0; j < n; j++) {
20         if (i & (1 << j)) {
21             cout << set[j] << " ";
22         }
23     }
24     cout << endl;
25 }
26 }
27
28 if (!found) {
29     cout << "No subset found with sum " << target << endl;
30 }
31 }
32
33 int main() {
34     int set[] = {1, 3, 4, 5}; int n =
35     sizeof(set) / sizeof(set[0]); int
36     target = 15;
37     subsetSum(set, n, target);
38     return 0;
39 }
```

input

No subset found with sum 15

Q3.)Graph Coloring?

Ans->



The image shows a screenshot of a C++ IDE with a dark theme. The editor window displays a C++ program for graph coloring. The code defines a constant V for the number of vertices (4) and implements three functions: isSafe, graphColoringUtil, and graphColoring. The main function calls graphColoring, which prints the solution if it exists. The output at the bottom of the IDE shows the message: "Solution exists and the assigned colors are: 1 2 3 2".

```
main.cpp
1  #include <iostream>
2  #include <cstring>
3  using namespace std;
4
5  #define V 4 // Number of vertices
6  bool isSafe(int v, bool graph[V][V], int color[], int c) {
7      for (int i = 0; i < V; i++) if
8      (graph[v][i] && c == color[i])
9          return false;
10     return true;
11 }
12
13 bool graphColoringUtil(bool graph[V][V], int m, int color[], int v) {
14     if (v == V) return true;
15     for (int c = 1; c <= m; c++) {
16         if (isSafe(v, graph, color, c)) {
17             color[v] = c;
18             if (graphColoringUtil(graph, m, color, v + 1))
19                 return true;
20             color[v] = 0;
21         }
22     }
23     return false;
24 }
25
26 bool graphColoring(bool graph[V][V], int m) {
27     int color[V]; memset(color, 0,
28     sizeof(color)); if
29     (!graphColoringUtil(graph, m, color, 0)) {
30         cout << "Solution does not exist";
31         return false;
32     }
33     cout << "Solution exists and the assigned colors are: ";
34     for (int i = 0; i < V; i++)
35         cout << color[i] << " "; cout
36         << endl;
37     return true;
38 }
39
40
41
```

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