Experiment:2.2

Student Name: Virat Samdarshi **UID: 22BCS12648**
Branch: B.E CSE **Section/Group:IOT-627-B**
Semester:5th **DateofPerformance:11-09-24**
Subject Name:DAAlab **SubjectCode:22CSH-311**

1. **Aim:** Develop a program and analyze complexity to implement subset-sum problem using Dynamic Programming.

2. **Objective:** To determine if there exists a subset within a given set of integers that sums up to specified target value. It does by generating possible subsets, calculating their sums, and checking if any subset matches target sum.

3. Implementation/Code:

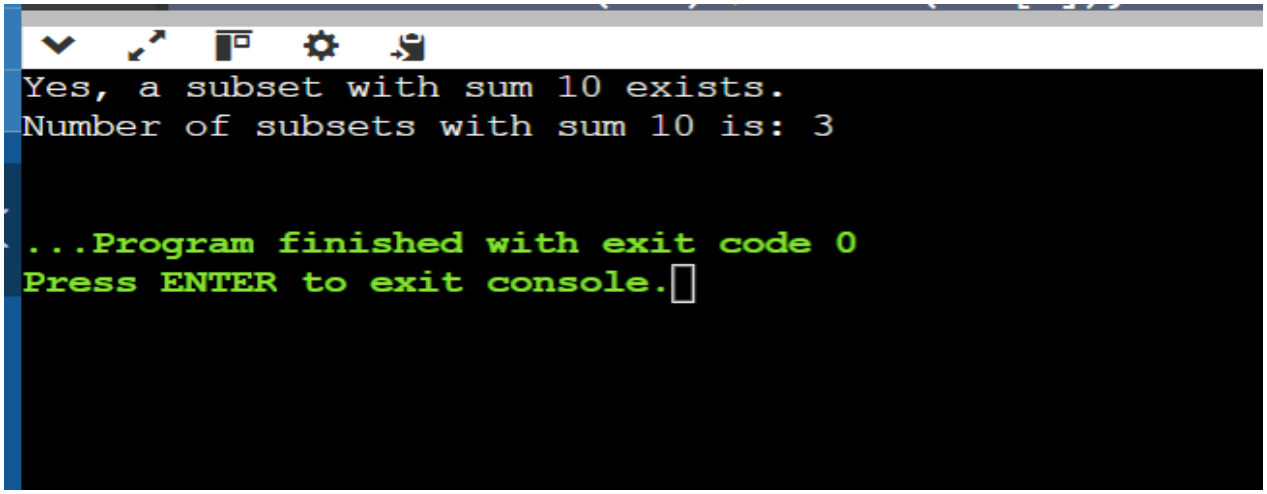
```
#include <iostream>
using namespace std;
bool isSubsetSum(int arr[], int n, int sum, int &subsetCount) {
    int dp[n + 1][sum + 1];
    for (int i = 0; i <= n; i++)
        dp[i][0] = 1;
    for (int j = 1; j <= sum; j++)
        dp[0][j] = 0;
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= sum; j++) {
            if (arr[i - 1] <= j) {

                dp[i][j] = dp[i - 1][j] + dp[i - 1][j - arr[i - 1]];
            } else {
             t
                dp[i][j] = dp[i - 1][j];
            }
        }
    }
    subsetCount = dp[n][sum];
    return (dp[n][sum] > 0);
}
int main() {
    int arr[] = {2, 3, 7, 8, 10};
    int sum = 10;
```

```
    int n = sizeof(arr) / sizeof(arr[0]);
    int subsetCount = 0;
    if (isSubsetSum(arr, n, sum, subsetCount)) {
        cout << "Yes, a subset with sum " << sum << " exists." << endl;
        cout << "Number of subsets with sum " << sum << " is: " << subsetCount <<
endl;
    } else {
        cout << "No subset with sum " << sum << " exists." << endl;
    }

    return 0;
}
```

4. Output:



5. Complexity:

Time Complexity: O(n * sum)

Space Complexity: O(n * sum)