ASSIGNMENT NO :- 4

ROLL NO :- 33252

```
#include <bits/stdc++.h>
using namespace std;
bool backtracking(int sum, vector<int>& arr, int i, int fsum){
  if (fsum == sum) {
    return true;
  }
  if (i >= arr.size()) {
    return false;
  }
  bool take = false;
  if (arr[i] + sum <= fsum) {
    take = backtracking(sum + arr[i], arr, i + 1, fsum);
  }

  bool non_take = backtracking(sum,arr,i+1,fsum);
  return non_take || take;
}
vector<int> answer(int sum, vector<int>& arr, int i, int fsum, vector<int>& ans) {
  if (sum == fsum) {
    return ans;
  }

  if (i >= arr.size() || sum > fsum) {
    return {};
  }

  ans.push_back(arr[i]);
  vector<int> res1 = answer(sum + arr[i], arr, i + 1, fsum, ans);
```

```
    if (!res1.empty()) {
        return res1;
    }

    ans.pop_back();
    return answer(sum, arr, i + 1, fsum, ans);
}
int main() {
    int fsum;
    cout << "Enter target sum: ";
    cin >> fsum;

    int n;
    cout << "Enter number of elements: ";
    cin >> n;

    vector<int> arr(n);
    vector<vector<int>>dp(n+1,vector<int>(fsum+1,-1));
    cout << "Enter elements: ";
    for (int i = 0; i < n; i++) {
        cin >> arr[i];
    }

    if (backtracking(0, arr, 0, fsum)) {
        cout << "A subset with the given sum exists." << endl;
    } else {
        cout << "No subset with the given sum exists." << endl;
    }

    vector<int>ans;
    vector<int>fans = answer(0, arr, 0, fsum,ans);
    for(auto it:fans){
        cout<<it<<" ";
    }
    return 0;
}
```

OUTPUT

```
Enter target sum: 30

Enter number of elements: 6

Enter elements: 5

10

12

13

15

18

A subset with the given sum exists.

5 10 15

Enter target sum: 35

Enter number of elements: 6

Enter elements: 5 7 10 12 15 18

A subset with the given sum exists.

5 12 18
```