EXP11

23UAI313

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

void subsets(int arr[], int n, int subset_sum, int subset[], int index, bool &is_present)
{
    if (index == n)
    {
        // Base case: when all elements have been considered
        if (subset_sum == 0)
        {
            is_present = true;
        }
        return;
    }
    // Include current element in the subset and recurse
    subset[index] = arr[index];
    subsets(arr, n, subset_sum - arr[index], subset, index + 1, is_present);
    // Exclude current element from the subset and recurse
    subset[index] = 0;
    subsets(arr, n, subset_sum, subset, index + 1, is_present);
}

int main()
{
    int n;
    cout << "Enter the size of the input array: ";
    cin >> n;
    int arr[n];
    cout << "Enter the elements of the array: ";
    for (int i = 0; i < n; i++)
    {
        cin >> arr[i];
    }
    int subset_sum;
    cout << "Enter the desired subset sum: ";
    cin >> subset_sum;
    int subset[n] = {0};
    bool is_present = false;
    subsets(arr, n, subset_sum, subset, 0, is_present);
    if (is_present)
    {
        cout << "Subset with sum " << subset_sum << " is present in the array" << endl;
    }
    else
    {
        cout << "Subset with sum " << subset_sum << " is not present in the array" << endl;
    }
    return 0;
}
```