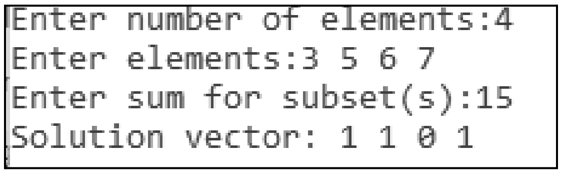
Sum of Subsets using Backtracking
Source Code:

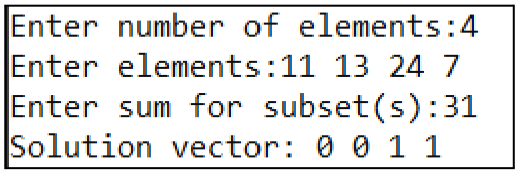
```
#include <iostream>
#include<conio.h>
using namespace std;

int *arr, sum, *trackarr;

void printArr(int k) {
	for (int i = 0; i <= k; i++)
		cout << trackarr[i] << " ";
	cout << endl;
}

void sumofSub(int lowBound, int k, int upBound) {
	trackarr[k] = 0;
	if (lowBound + arr[k] == sum) {
		trackarr[k] = 1;
		printArr(k);
	}
	else if (lowBound + arr[k] + arr[k + 1] <= sum) {
		trackarr[k] = 1;
		sumofSub(lowBound + arr[k], k + 1, upBound - arr[k]);
	}
	if ((lowBound + upBound - arr[k] >= sum) && (lowBound + arr[k + 1] <= sum)) {
		trackarr[k] = 0;
		sumofSub(lowBound, k + 1, upBound - arr[k]);
	}
}
int main() {
	int size, upBound = 0;
	// Write C++ code here
	cout << "Enter number of elements:";
	cin >> size;
	arr = new int[size];
	trackarr = new int[size];
	cout << "Enter elements:";
	for (int i = 0; i < size; i++) {
		cin >> arr[i];
		upBound += arr[i];
```

```
        }
        cout << "Enter sum for subset(s):";
        cin >> sum;
        cout << "Solution vector: ";
        sumofSub(0, 0, upBound);
        _getch();
        return 0;
}
```

Output:



```
Enter number of elements:4
Enter elements:3 5 6 7
Enter sum for subset(s):15
Solution vector: 1 1 0 1
```



```
Enter number of elements:4
Enter elements:11 13 24 7
Enter sum for subset(s):31
Solution vector: 0 0 1 1
```