Design and implement C/C++ Program to find a subset of a given set S = {sl , s2,.....,sn} of n positive integers whose sum is equal to a given positive integer d.

```
#include <stdio.h>

#include <stdlib.h>


#define MAX 20


int x[MAX];

int s[MAX];

int d, flag = 0;


void SumofSub(int m, int k, int r);

void inputArray(int arr[], int n);


int main() {

   int n, sum = 0;


   printf("Enter the number of elements: ");

   scanf("%d", &n);


   printf("Enter the elements:\n");

   inputArray(s, n);
```

```
    printf("Enter the value of d: ");
    scanf("%d", &d);

    for (int i = 1; i <= n; i++) {
        sum += s[i];
    }

    if (sum < d || s[1] > d) {
        printf("The given problem instance does not have a solution\n");
        return 0;
    } else {
        printf("Subsets are:\n");
        SumofSub(0, 1, sum);
    }

    if (flag == 0) {
        printf("No subset possible\n");
    }

    return 0;
}
```

```
// Method to Calculate the Subset Sum
void SumofSub(int m, int k, int r) {
    int i;
    x[k] = 1;
    if (m + s[k] == d) {
        flag = 1;


        printf("{");
        for (i = 1; i <= k; i++) {
            if (x[i] == 1) {
                printf("%d ", s[i]);
            }
        }
        printf("}\n");
    } else if ((m + s[k] + s[k + 1]) <= d) {
        SumofSub(m + s[k], k + 1, r - s[k]);
    }


    if ((m + r - s[k] >= d) && (m + s[k + 1] <= d)) {
        x[k] = 0;
        SumofSub(m, k + 1, r - s[k]);
```

```
    }
}

// Function to input array elements
void inputArray(int arr[], int n) {
    for (int i = 1; i <= n; i++) {
        scanf("%d", &arr[i]);
    }
}
```