CSE 331 Computer Organization

Homework 2 Report

Name: Rahmet Ali Ölmez

Number: 1801042623

Explanation of Functions:

The CheckSumPossibility function takes three arguments: num (the target number), arr (the array to be searched) and size (the size of arr).

This is a recursive function and it searches if there is any subset of the array that when summed up, equals to the target number. Since this is a recursive function, I wrote the base cases for the function. I've thought of three base cases:

- When num is equal to zero: This means that the target sum is found. (See below for details) 1 is returned.
- When num is smaller than zero: This means that the target sum is not found. 0 is returned.
- When size equal to one: This has two sub-cases, when num equals to arr[0] and when they are not equal. I implemented it like this to avoid trying to access any invalid index. In the former case, 1 is returned because the target number will be zero when subracted, in the latter case 0 is returned.

Unless the base cases are reached, the function keeps on calling itself in two ways. The first recursive call takes target number subtracted by the last element of the array, now this becomes the new target number. It also takes the size decremented by one. The other call is the same, but this time the target num is not changed, this helps us search all the possible cases. The basic idea is to solve the problem in a smaller case, and apply it to bigger ones.

Result of Test Cases:

Test 1:

```
Size: 8

Expected: Not possible!

Result: Not possible!
```

```
rahmet@DESKTOP-8T8JQGV:/mnt/c/users/Rahmet/Desktop/cse331-Org/hw/hw2$ make run
./hw2
8
129
41
67
34
0
69
24
78
58
Not possible!
```

Mars Messages | Run I/O

```
Please enter arr size, target num and the elements of the array:
8
129
41
67
34
0
69
24
78
58
Not possible!

-- program is finished running --
```

Test 2:

Size: 8

Expected: Not possible!

Result: Not possible!

```
rahmet@DESKTOP-8T8JQGV:/mnt/c/users/Rahmet/Desktop/cse331-Org/hw/hw2$ make run
./hw2
8
129
62
64
5
45
81
27
61
91
Not possible!
```

Mars Messages | Run I/O

```
Reset: reset completed.

Please enter arr size, target num and the elements of the array:
8
129
62
64
5
45
81
27
61
91
Not possible!

-- program is finished running --
```

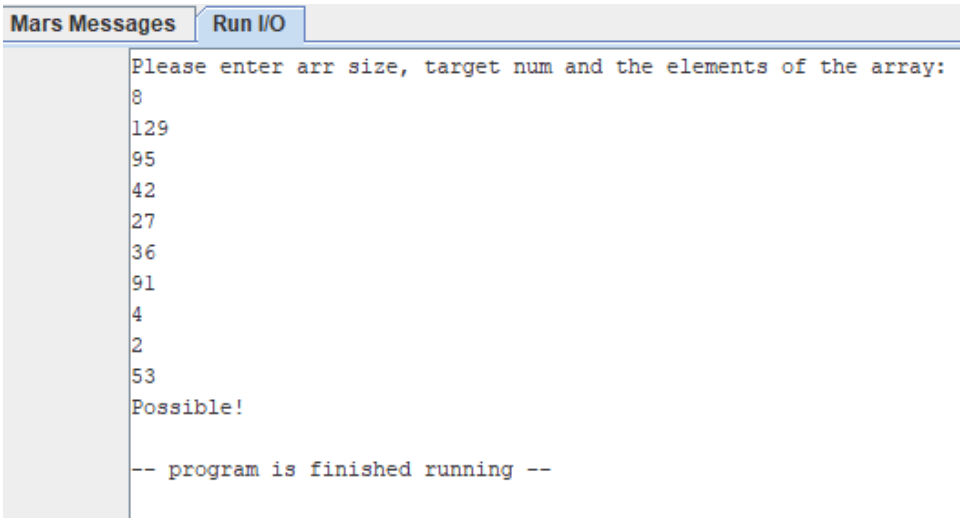
Test 3:

Size: 8

Expected: Possible!

Result: Possible!

```
rahmet@DESKTOP-8T8JQGV:/mnt/c/users/Rahmet/Desktop/cse331-Org/hw/hw2$ make run
./hw2
8
129
95
42
27
36
91
4
2
53
Possible!
```



Test 4:

Size: 8

Expected: Possible!

Result: Possible!

```
rahmet@DESKTOP-8T8JQGV:/mnt/c/users/Rahmet/Desktop/cse331-Org/hw/hw2$ make run
./hw2
8
129
92
82
21
16
18
95
47
26
Possible!
```

Mars Messages | Run I/O

```
Please enter arr size, target num and the elements of the array:
8
129
92
82
21
16
18
95
47
26
Possible!

-- program is finished running --
```

Test 5:

Size: 8

Expected: Possible!

Result: Possible!

```
rahmet@DESKTOP-8T8JQGV:/mnt/c/users/Rahmet/Desktop/cse331-Org/hw/hw2$ make run
./hw2
8
129
71
38
69
12
67
99
35
94
Possible!
```

Mars Messages | Run I/O

```
Please enter arr size, target num and the elements of the array:
8
129
71
38
69
12
67
99
35
94
Possible!

-- program is finished running --
```

Test 6:

Size: 8

Expected: Not possible!

Result: Not possible!

```
rahmet@DESKTOP-8T8JQGV:/mnt/c/users/Rahmet/Desktop/cse331-Org/hw/hw2$ make run
./hw2
8
129
3
11
22
33
73
64
41
11
Not possible!
```

Mars Messages | Run I/O

```
Please enter arr size, target num and the elements of the array:
8
129
3
11
22
33
73
64
41
11
Not possible!

-- program is finished running --
```

Test 7:

Size: 15

Expected: Possible!

Result: Possible!

```
rahmet@DESKTOP-8T8JQGV:/mnt/c/users/Rahmet/Desktop/cse331-Org/hw/hw2$ make run
./hw2
15
100
20
65
42
10
87
70
63
36
11
49
17
39
22
97
18
Possible!
```

Mars Messages | Run I/O

```
Please enter arr size, target num and the elements of the array:
15
100
20
65
42
10
87
70
63
36
11
49
17
39
22
97
18
Possible!

-- program is finished running --
```