CSE 331/503 Computer Organization Homework 2

REPORT

Muhammet Fikret ATAR / 1801042693

Cpp:

CheckSumPossibility function:

The function respectively gets the target number, the array and the size of the array. It returns 1 if a subset of the array can sum up to the target number and otherwise if it is not possible. My CheckSumPossibility function works as recursively.I have some base case like num == 0, arraySize==0 || num<0. If num equal zero,function return 1. Return 1 meaning subset of the array can sum up to the target number. If arraySize == 0 then it means that function didn't find a possibility to obtain target value.

Mips:

$a1 #save arraysize

$a2 #save targetnumber

$t0   #save arrayadress

while1: it's same as for loop in my assembly code. I used it for getting array-inputs from the user in my main function

print: print possible if $v0= 1

print1: print not possible if $v0= 0

ChecknumPossibility: recursive function

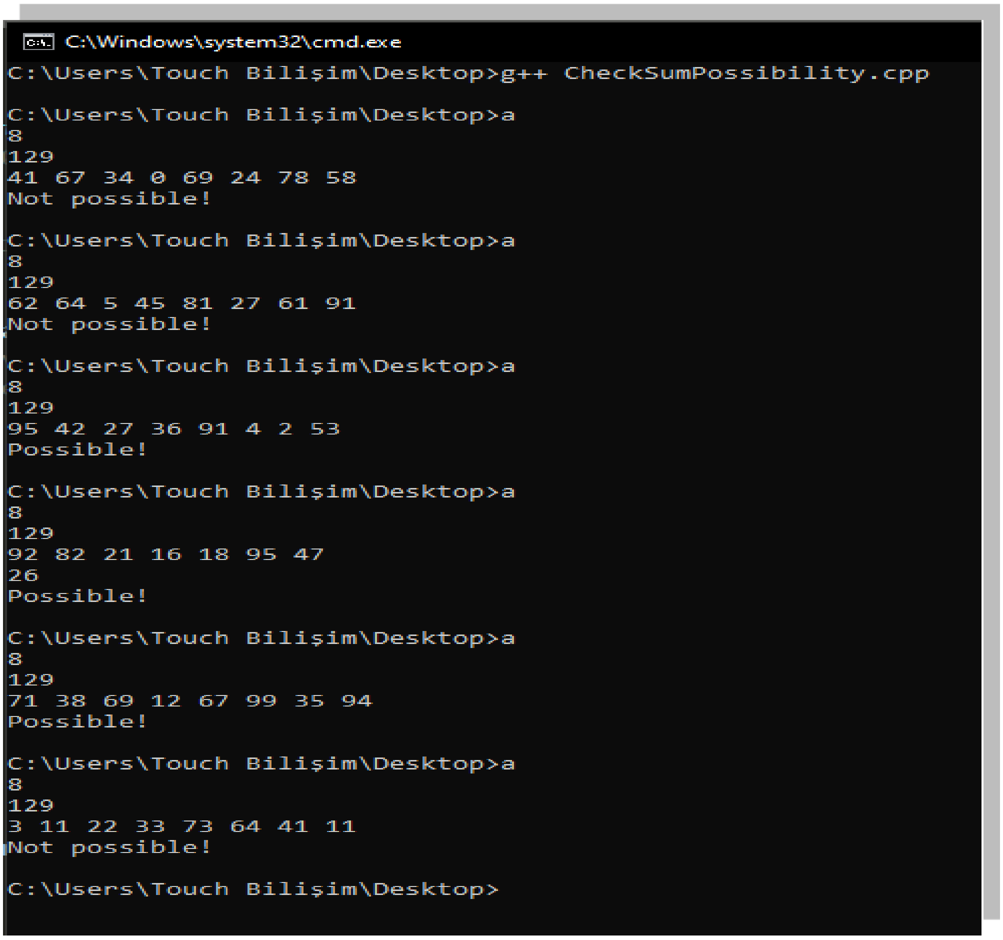
REC1: This label makes num - arr[arraySize] and arraySize - 1

REC2: This label makes arraySize - 1

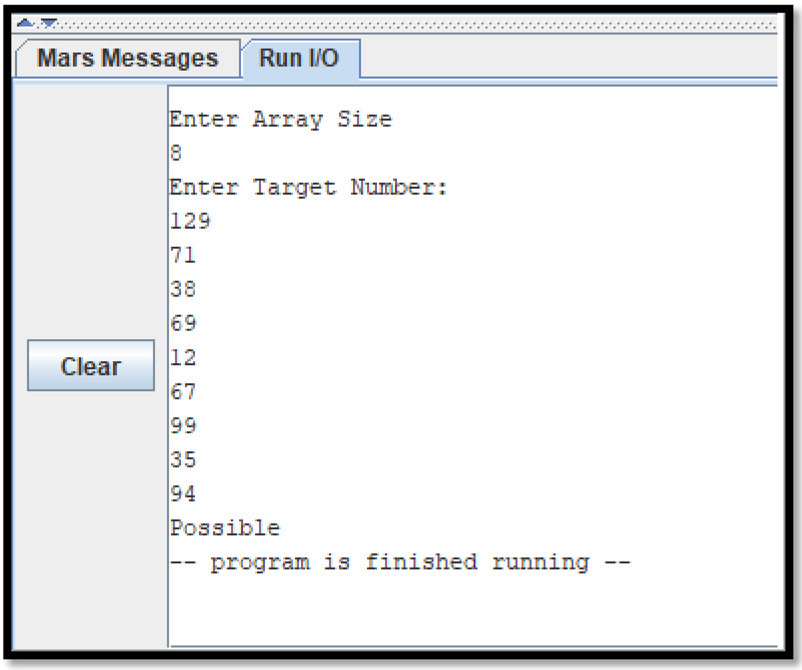
ifreturn1:In this label #vo = 1 and return 1 . this mean sub set of array can sum up to target number .and jump with ra to start of function

ifreturn0: In this label #vo = 0 and return 0 . this mean sub set of array can not sum up to target number .And jump with ra to one upper

TEST CPP AND MİSP:



```
C:\Users\Touch Bilişim\Desktop>g++ CheckSumPossibility.cpp

C:\Users\Touch Bilişim\Desktop>a
8
129
41 67 34 0 69 24 78 58
Not possible!

C:\Users\Touch Bilişim\Desktop>a
8
129
62 64 5 45 81 27 61 91
Not possible!

C:\Users\Touch Bilişim\Desktop>a
8
129
95 42 27 36 91 4 2 53
Possible!

C:\Users\Touch Bilişim\Desktop>a
8
129
92 82 21 16 18 95 47
26
Possible!

C:\Users\Touch Bilişim\Desktop>a
8
129
71 38 69 12 67 99 35 94
Possible!

C:\Users\Touch Bilişim\Desktop>a
8
129
3 11 22 33 73 64 41 11
Not possible!

C:\Users\Touch Bilişim\Desktop>
```

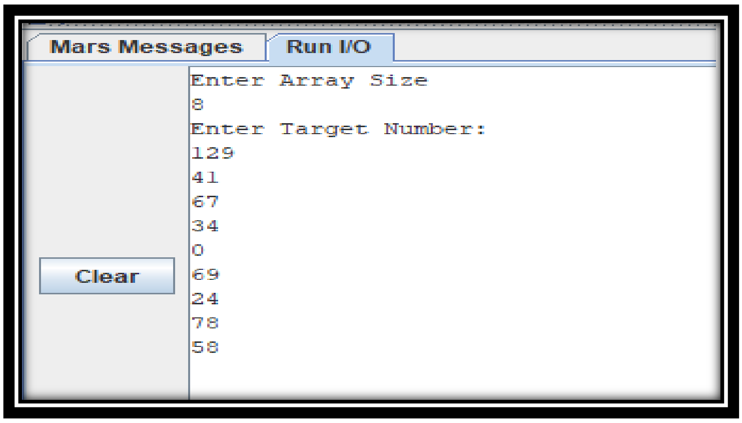


```
Enter Array Size
8
Enter Target Number:
129
71
38
69
12
67
99
35
94
Possible
-- program is finished running --
```

There are some problem except for the above test;

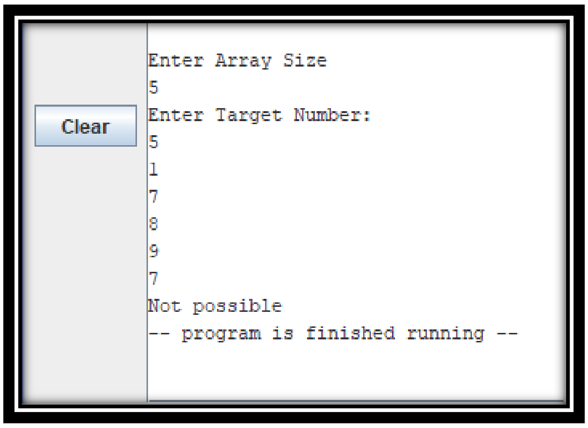
In my assembly code, I think I am progressing respectively, but after base case checks, sometimes i have trouble about returning. So my function stays in an infinite loop.

Such as:



```
Enter Array Size
8
Enter Target Number:
129
41
67
34
0
69
24
78
58
```

My own sample:



```
Enter Array Size
5
Enter Target Number:
5
1
7
8
9
7
Not possible
-- program is finished running --
```