CSE 331 Computer Organizations Homework 2

Fatih Kaan Salgır - 171044009

1 Algorithm Explanation

Inputs: Array of elements along with its size and number to be checked

Output: 1 if if sub elements of the array can sum up to this number, 0 otherwise.

If there are elements in the array can sum up to the number; an element can either be inclueded or excluded to this group. Therefore elements need to be checked until find a solution or there are no elements left. For any element there are 2 possibilities, inclueded or excluded.

Without optimization algorithm looks like;

```
1. Check rest of the array without the element.
2. Check rest of the array with the element.

- Return 1 if summed values are equal to target number.
- Return 0 if size exceed.
```

I have preferd to decrase the number, and check if it is 0. Instead of summing up.

1.1 Optimzations

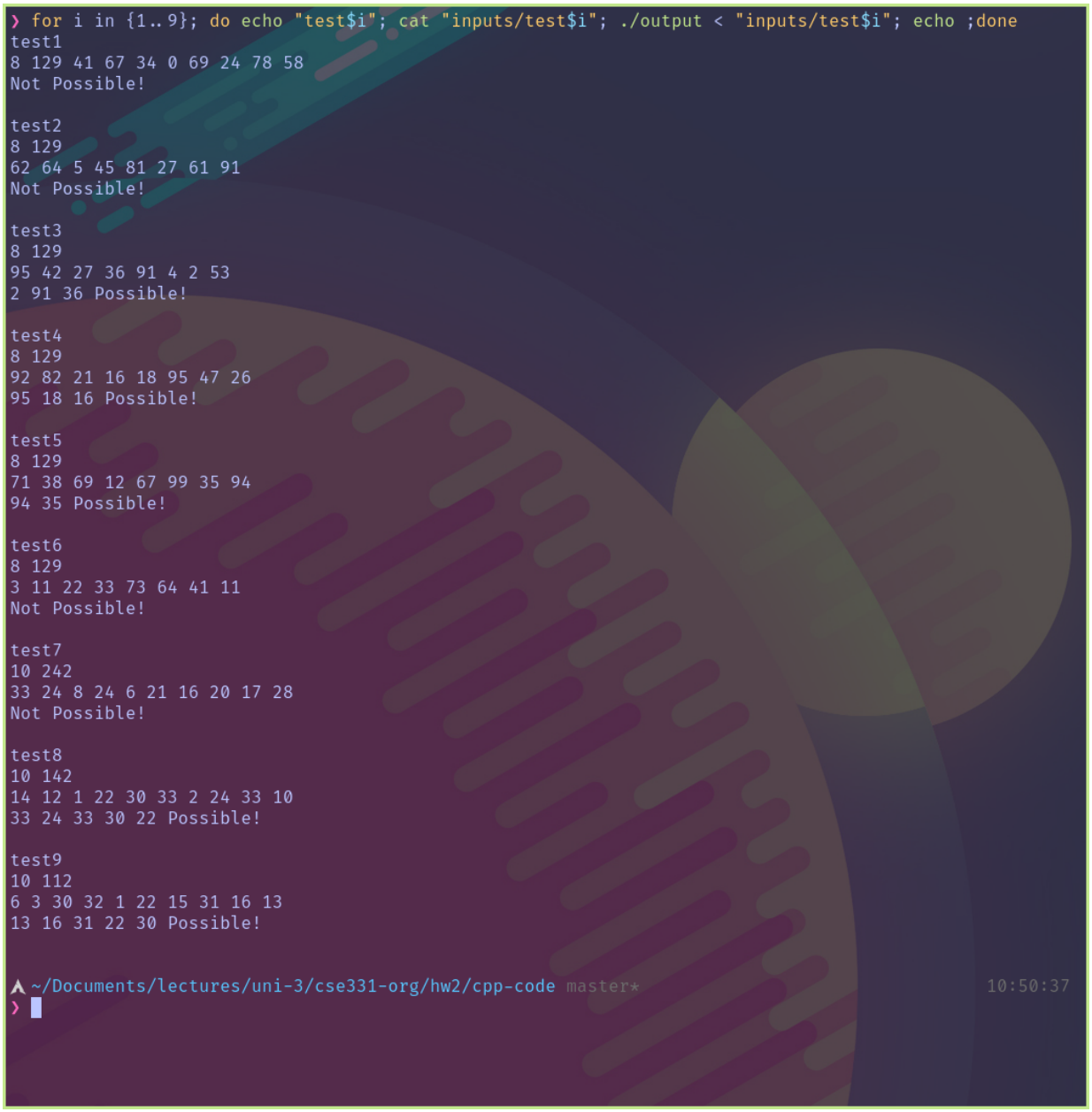
- If summed number exceed to target number, it is pointless to keep adding. (In my case `num < 0`)
- If element is equal to target or remaining value is equal the target then return 1.

I impemented the printing numbers in assembly. I did not store them in a structure, only print the screen when it is found.

Note: Results might not be the same with the pdf, but they are accurate. So, it might be slower or faster according to input data.

2 Test Cases

C++ code output:



MARS output:

```
112  .data
113  test1:   .word   41 67 34 0 69 24 78 58  #Not possible!
114  test2:   .word   62 64 5 45 81 27 61 91  #Not possible!
115  test3:   .word   95 42 27 36 91 4 2 53   #Possible!
116  test4:   .word   92 82 21 16 18 95 47 26 #Possible!
117  test5:   .word   71 38 69 12 67 99 35 94 #Possible!
118  test6:   .word   11 22 33 73 64 41 11    #Possible!
119  test7:   .word   33 24 8 24 6 21 16 20 17 28
120  test8:   .word   14 12 1 22 30 33 2 24 33 10
121  test9:   .word   6 3 30 32 1 22 15 31 16 13
122
```

Line: 111 Column: 1 Show Line Numbers

Mars Messages Run I/O

```
Not Possible!
-- program is finished running --

Not Possible!
-- program is finished running --

2 91 36 Possible!
-- program is finished running --

95 18 16 Possible!
-- program is finished running --

94 35 Possible!
-- program is finished running --

33 41 33 22 Possible!
-- program is finished running --

Not Possible!
-- program is finished running --

33 24 33 30 22 Possible!
-- program is finished running --
```

Clear