


# CSE-331 Homework 2 Report

|                         |  |
|-------------------------|--|
| ☰ Computer Organization | HomeWork Lecture   |
| 📅 Created               | @Nov 26, 2020  |
| 👤 Property              |  Baran Hasan Bozduman |

## PART-1

I did not change the signature of the function which is recursive function which is given below

```
int CheckSumPossibility(int num, int arr[], int size){
```

Basically I get the parameters above and make two recursive calls for them and each time I subtract the array elements from given target number if it provides zero that means we can return that value as true.

I make two calls one of them for the current index of array the other one for the other elements of the array and we check all array elements like this.

```
if(CheckSumPossibility( num-arr[size-1], arr, size-1)){
```

```
if(CheckSumPossibility(num, arr, size-1)){
```

and we make that until target element reaches to zero or array of index goes below zero, For detail you can check the comments of the cpp file.

Also I can print the set of the numbers even there is a more than one subset you can see them also, to print all sets i just need to if condition without else so it can check each numbers and put a comma between them the outputs for the array given below

The screenshot shows the Kate text editor with a file named `hw2.cpp`. The code is a C++ program that checks if a subset of an array can sum up to a given number. The code is as follows:

```
1 #include <iostream>
2 using namespace std;
3 #define MAX_SIZE 20
4
5 int CheckSumPossibility(int num, int arr[], int size){
6     int returnVal1=0, returnVal2=0;
7     if (num == 0){//If sum becomes 0 that means it find a subset so it
8         can return 1
9         cout<<" ";//to split numbers like if there is more then one pairs
10        which gives the target number to split the i use comma
11        return 1;
12    }
13    //if number becomes smaller than 0 that means the number which is
14    subtracted bigger than number so it returns 0
15    //also if the index smaller than zero that means it tries to out of
16    index bound
17    if(size <= 0 || num <0){
18        return 0;
19    }
20    //for each call it subtracts given indis from number and return next
```

The terminal output shows the execution of the program for six different inputs:

```
[anonxx@eXmachina computerOrganization]$ g++ hw2.cpp
[anonxx@eXmachina computerOrganization]$ ./a.out
8 129
41 67 34 0 69 24 78 58
Not possible
[anonxx@eXmachina computerOrganization]$ ./a.out
8 129
62 64 5 45 81 27 61 91
Not possible
[anonxx@eXmachina computerOrganization]$ ./a.out
8 129
95 42 27 36 91 4 2 53
(,36 91 2 )
Possible!
[anonxx@eXmachina computerOrganization]$ ./a.out
8 129
92 82 21 16 18 95 47 26
(,82 21 26 ,82 47 ,16 18 95 ,92 21 16 )
Possible!
[anonxx@eXmachina computerOrganization]$ ./a.out
8 129
71 38 69 12 67 99 35 94
(,35 94 )
Possible!
[anonxx@eXmachina computerOrganization]$ ./a.out
8 129
3 11 22 33 73 64 41 11
Not possible
```

For the inputs I tried 6 inputs which is given assignment file so it works for all of them

**Usage:**First you need to enter array size then you enter target number and then you can enter the elements of array. My Max array size is 20 but you can change it if you want to enter more than 20 elements.



Both of the program wont ask anything the input order given above green box for mips and cpp part

The screenshot shows the Kate text editor with a file named `hw2.cpp`. The code is a C++ program that reads an array size and a number, then checks if the sum of the array elements is less than, equal to, or greater than the number. The code is as follows:

```
34     int returnVal;
35     int sumarray=0;
36
37     cin >> arraySize;
38     cin >> num;
39
40     for (int i = 0; i < arraySize; i++){
41         cin >> arr[i];
42         sumarray+=arr[i]; //when it adds it also sum all of the array
                               elements
43     }
44     if(sumarray < num){
45         returnVal=0; //if sum of array smaller than number it assigns
                               0
46     }
47     else if(sumarray == num){
48         returnVal=1; //if sum of all elements gives the target value it
                               returns 1
49         cout<<"All elements is sum subset of given value "<<endl;
50     }
51     else{
52         cout<<"("; // I have to put them here to print paranthesies
53         returnVal = CheckSumPossibility(num, arr, arraySize);
54     }
```

The terminal output shows the program being run twice. The first run shows the input `3 4 1 1 2` and the output `All elements is sum subset of given value ) Possible!`. The second run shows the input `3 5 2 1 1` and the output `Not possible`.

```
[anonxx@eXmachina computerOrganization]$ ./a.out
3
4
1
1
2
All elements is sum subset of given value
)
Possible!
[anonxx@eXmachina computerOrganization]$ ./a.out
3
5
2
1
1
Not possible
[anonxx@eXmachina computerOrganization]$
```

As you see above I also optimize the code I had to change main to do that so I hope it does not give a trouble to me. I also if sum the array elements it wont

[call recursive](#) for detail you can read the comments above of picture and you can check the outputs.

## PART-2

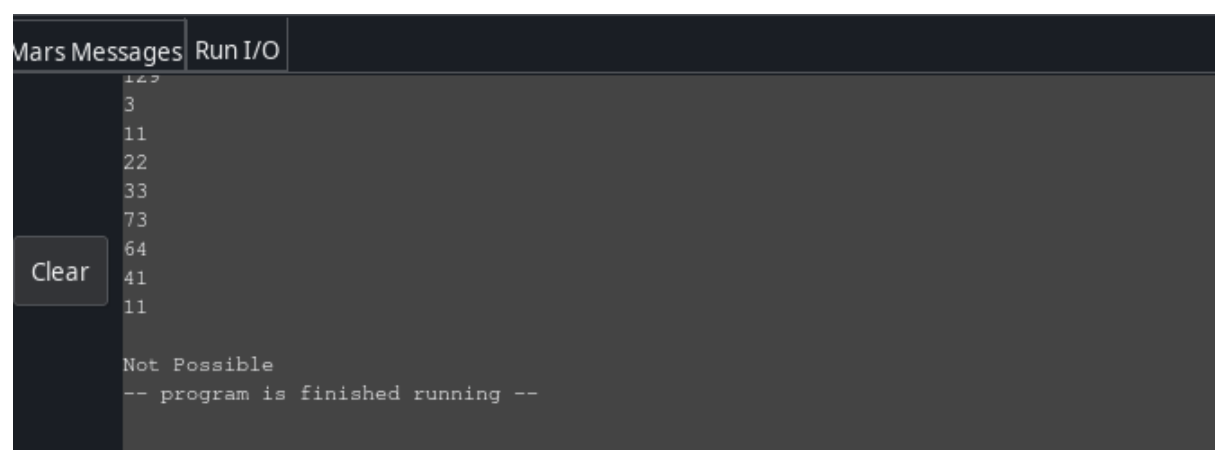
For the MIPS code Firstly I defined almost all variables in `.data` part as word even my array and its size is for the 100 integers also I defined the text messages as `.asciiz` because I did not want to mess with registers because its hard to remember what I put where. So generally if I use a value I basically store or load from `.data` elements.

I can not say that is exactly the same of the cpp code there are some changes like sometimes instead of store a value somewhere i used it immediately if i wont need it any more.

[I believe I stuck with the mips contract.](#) just I did not lw sw specific places I used them if I need it because at first I confused easily then I decided to use them when I need them.[Also I print the sets.](#)

For detail you can check the comments of the mips code I explained almost each steps.

You can check the outputs below for the six inputs of the given which is in the assignment file.



```
Mars Messages Run I/O
123
3
11
22
33
73
64
41
11

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

| Mars Messages  | Run I/O |
|--|---------|
| 8<br>129<br>41<br>67<br>34<br>0<br>69<br>24<br>78<br>58<br><br>Not Possible<br>-- program is finished running -- |         |

| Mars Messages  | Run I/O |
|--|---------|
| 8<br>129<br>62<br>64<br>5<br>45<br>81<br>27<br>61<br>91<br><br>Not Possible<br>-- program is finished running -- |         |

| Mars Messages  | Run I/O |
|--|---------|
| 129<br>71<br>38<br>69<br>12<br>67<br>99<br>35<br>94<br><br>35<br>94<br>Possible! |         |

```
Mars Messages Run I/O
8
129
92
82
21
16
18
95
47
26

Clear

82
21
```

```
Mars Messages Run I/O
21
26

82
47

Clear

16
18
95

92
21
16
Possible!
```

```
Mars Messages Run I/O
42
27
36
91
4
2
53

Clear

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

**Usage:**First you need to enter array size then you enter target number and then you can enter the elements of array. My Max array size is 20 but you can change it if you want to enter more than 100 elements.

For the optimization I add the same thing of cpp algorithm checks the sum of the all array elements.

If the sum of array smaller than the given input it will print as next pic also if sum of array is equal to number it doesnt need to call recursive function as below you can check the asm comments for detail.

| Mars Messages         | Run I/O  |
|-----------------------|--|
| 3<br>4<br>1<br>2<br>1 | Summation of all array elements provides the given valuePossible!<br>-- program is finished running -- |

| Mars Messages         | Run I/O   |
|-----------------------|---|
| 3<br>7<br>4<br>1<br>0 | The Summation given array is smaller than the valueThe Summation given array is smaller than the value<br>-- program is finished running -- |