CSE 331 – COMPUTER ORGANIZATION HOMEWORK 2 REPORT

Name Surname : Emirhan UZUN
Number : 171044019

REQUESTED

We are askd to write a 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 it outputs a 0.

ALGORITHM AND TO-DO

1 – I reserved memory for values and strings.
2 – Go to print array size label
a – Assign $v0 to 4, $a0 to ArrSizeMessage and syscall
b – jr $ra
3 – Go to read array size label
a – Assign $v0 to 5 and syscall
b – Store word $v0 to Arr Size value
c – jr $ra
4 – Go to print target number label
a – Assign $v0 to 4, $a0 to Target Number Message and syscall
b – jr $ra
5 – Go to read target number label
a – Assign $v0 to 5 and syscall
b – Store word $v0 to Target Number value
c – jr $ra
6 – Assign $t1 to Arr Size value and jal loop (Fill Array)
7 – Print queue message which is "1.number" or "2.number" etc.
a – Take the value from user and store it in the Arr adress.
b – Increment the arr adress 4 by 4($t0 in my code) because integer is 4 byte and we have to go 4 byte forward
c – If the loop finish ($t2 is equal to $t1 which is arr size) , go to assign values label. This label is like an exit label for loop.
8 – Jump printResult label from assign values. This label goes to recursive function and if the came input is true print possible otherwise not possible.
a – Fill the $a0,$a1 and $a2 for function parameters
b – Go to CheckSumPossibilit function
9 – CheckSumPossibility recursive function adjust stack for parameters and return adress.
a – Check first base case which is num == 0. If it is true, return1
b – Check second base which is size == 0. If it is true return 0.
c – Check third if but not base case. The goal of that condition, if the arr[size-1] is bigger than num , then just call one recursive (OPTIMIZATED).
d – Call 2 recursive call (First I call the recursive with num- arr[size-1] parameters because it substracts the array elements from target. The other recursive just decrease the size by 1 and send. So first recursive is faster than the other recursive call (OPTIMIZATED))
e – Turn recursive calls
10 – Again back to the print result label. Print the result
11- Print Array elements and target number

INFORMATIONS

1 – In my code, bigger than 10 element, sometimes it gives respond but late. So if you wait a little, you will get respond.

2 – Optimizated : * If the arr[size-1] is bigger than target, then just call one recursive

*** In third condition, I optimized my code because I do not recursive call unnecessarily**

TEST CASES

First three tests, I used same input and the other 3 test I used different input. Firstly I Show 3 c++ and assembly test for same input

CASE 1 - 3:

Array Size : 10

Array : 104 98 87 12 25 77 34 58 29 81

Case 1 Target Number 61 (Cpp)

D:\Books\Assignments\HW2 Organizasyon\hw2.exe

```
Enter the array size : 10
Enter the target number : 61
1.number : 104
2.number : 98
3.number : 87
4.number : 12
5.number : 25
6.number : 77
7.number : 34
8.number : 58
9.number : 29
10.number : 81
Not possible!

--------------------------------
Process exited after 19.6 seconds with return value 0
Press any key to continue . . .
```

Case 1 Target Number 61 (Assemb

```
Enter the array size (Max 100) : 10

Enter the target number :61
1.number :
104
2.number :
98
3.number :
87
4.number :
12
5.number :
25
6.number :
77
7.number :
34
8.number :
58
9.number :
29
10.number :
81

Not Possible !
The array is as follows : 104 98 87 12 25 77 34 58 29 81
 The target number is : 61
-- program is finished running --
```

Case 2 Target Number 132 (Cpp) (Expected Possible (98,34))

```
D:\Books\Assignments\HW2 Organizasyon\hw2.exe
Enter the array size : 10
Enter the target number : 132
1.number : 104
2.number : 98
3.number : 87
4.number : 12
5.number : 25
6.number : 77
7.number : 34
8.number : 58
9.number : 29
10.number : 81
Possible!

--------------------------------
Process exited after 22.34 seconds with return value 0
Press any key to continue . . .
```

Case 2 Target Number 132 (Assembly) (Expected Possible (98,34))

```
Enter the array size (Max 100) : 10

Enter the target number :132
1.number :
104
2.number :
98
3.number :
87
4.number :
12
5.number :
25
6.number :
77
7.number :
34
8.number :
58
9.number :
29
10.number :
81

Possible !
The array is as follows : 104 98 87 12 25 77 34 58 29 81
 The target number is : 132
-- program is finished running --
```

Case 3 Target Number 262 (Cpp) (Expected Possible (104,81,77))

```
D:\Books\Assignments\HW2 Organizasyon\hw2.exe
Enter the array size : 10
Enter the target number : 262
1.number : 104
2.number : 98
3.number : 87
4.number : 12
5.number : 25
6.number : 77
7.number : 34
8.number : 58
9.number : 29
10.number : 81
Possible!

--------------------------------
Process exited after 20.52 seconds with return value 0
Press any key to continue . . .
```

Case 3 Target Number 262 (Assembly) (Expected Possible (104,81,77))

```
Enter the array size (Max 100) : 10

Enter the target number :262
1.number :
104
2.number :
98
3.number :
87
4.number :
12
5.number :
25
6.number :
77
7.number :
34
8.number :
58
9.number :
29
10.number :
81

Possible !
The array is as follows : 104 98 87 12 25 77 34 58 29 81
 The target number is : 262
-- program is finished running --
```

CASE 4 - 6:

Array Size : 8

Array : 15 28 63 72 91 112 124 8

Case 4 Target Number 211 (Cpp) (Expected Possible (112,8,63,28))

```
D:\Books\Assignments\HW2 Organizasyon\hw2.exe
Enter the array size : 8
Enter the target number : 211
1.number : 15
2.number : 28
3.number : 63
4.number : 72
5.number : 91
6.number : 112
7.number : 124
8.number : 8
Possible!

--------------------------------
Process exited after 15.16 seconds with return value 0
Press any key to continue . . .
```

Case 4 Target Number 211 (Assembly) (Expected Possible (112,8,63,28))

```
Enter the array size (Max 100) : 8

Enter the target number :211
1.number :
15
2.number :
28
3.number :
63
4.number :
72
5.number :
91
6.number :
112
7.number :
124
8.number :
8

Possible !
The array is as follows : 15 28 63 72 91 112 124 8
 The target number is : 211
-- program is finished running --
```

Case 5 Target Number 186 (Cpp) (Expected Possible (91,72,15,8))

```
D:\Books\Assignments\HW2 Organizasyon\hw2.exe
Enter the array size : 8
Enter the target number : 186
1.number : 15
2.number : 28
3.number : 63
4.number : 72
5.number : 91
6.number : 112
7.number : 124
8.number : 8
Possible!

--------------------------------
Process exited after 37.29 seconds with return value 0
Press any key to continue . . .
```

Case 5 Target Number 186 (Assembly) (Expected Possible (91,72,15,8))

```
Enter the array size (Max 100) : 8

Enter the target number :186
1.number :
15
2.number :
28
3.number :
63
4.number :
72
5.number :
91
6.number :
112
7.number :
124
8.number :
8

Possible !
The array is as follows : 15 28 63 72 91 112 124 8
 The target number is : 186
-- program is finished running --
```

Case 6 Target Number 245 (Cpp)

```
D:\Books\Assignments\HW2 Organizasyon\hw2.exe

Enter the array size : 8
Enter the target number : 245
1.number : 15
2.number : 28
3.number : 63
4.number : 72
5.number : 91
6.number : 112
7.number : 124
8.number : 8
Not possible!

--------------------------------
Process exited after 9.738 seconds with return value 0
Press any key to continue . . .
```

Case 6 Target Number 245 (Assembly)

```
Enter the array size (Max 100) : 8

Enter the target number :245
1.number :
15
2.number :
28
3.number :
63
4.number :
72
5.number :
91
6.number :
112
7.number :
124
8.number :
8

Not Possible !
The array is as follows : 15 28 63 72 91 112 124 8
 The target number is : 245
-- program is finished running --
```