

Coursework for Tutorial 1

Computer Architecture II:

October 2, 2023

1 Assignment

Task #1 Using the `_cdecl` calling convention, write an IA32 program that implements the C code listed in Figure 1, using the same computational logic and recursion. For a positive N , the function computes: $\sum_{i=N, \dots, 0} (i + \max\{i - 1, 0\} + \max\{i - 2, 0\} + \max\{i - 3, 0\})$. Draw the stack diagram for the call: “`my_sum3_wrapper(4)`”.

```
#include <stdio.h>

int my_sum3(int a, int b, int c){
    int x = 0;
    if(a<=0) {return x;}
    x = a--;
    if(b>=0){x+=b--;}
    if(c>=0){x+=c--;}
    return(x+my_sum3(a,b,c));
}

int my_sum3_wrapper(int N){
    return(my_sum3(N,N-1,N-2));
}

int main()
{
    int N = 6;
    int x;
    x = my_sum3_wrapper(10);
    printf("%d",x);
    return 0;
}
```

Figure 1: C implementation of the function for Task #1.

Task #2 Using the Microsoft Calling Convention, write an x64 function that translates a 64-bit input integer value into a string and prints it to the standard output. (Tip: in order to convert a single digit to ASCII you can add 48 to it, and to print it to standard output in JWasm you can use the code provided in the tutorial slides). Draw a stack diagram for the call `my_int2str(456)`.

2 Instructions

- Submit the source code files together with screenshot proof that your code works on 10 integers you randomly select from 0 to 100. The stack frame can be either hand-drawn or done electronically, but must be readable!
- This tutorial will count toward 10% of your final grade. Each task will count toward 50% of the overall tutorial score. The will be evaluated for correctness, readability (put comments!), efficiency of the code and correctness of the stack diagram – according to the rubric provided.

- Due date: October 15 @ 11.59pm.