CSCI-UA.0201 – 007 Computer Systems Organization
FALL 2022

PROJECT 3
WRITTEN ASSIGNMENT

Due date: 11.11.2022, 11.59pm
Late submissions due date: 11.14.2022, 11.59pm

This is an individual work. No team work is allowed.
Similarity check will be applied to submitted codes.

QUESTION 1 - x86 assembly to C pseudo code

According to the given assembly code on the left, write the updated values of ecx register.

`movl %eax, %ecx`	ecx = -----------
`movl $0x13, %ecx`	ecx = -----------
`movl 0x224, %ecx`	ecx = -----------
`movl(%ebx), %ecx`	ecx = -----------
`movl 2(%ebx), %ecx`	ecx = -----------

QUESTION 2 - x86 assembly to C code

Carefully read the given assembly code. Note that variables x, y and z are initially stored in %rdi, %rsi, and %rdx, respectively. Now, fill in the blanks in the given c code.

```
fun:
     leaq (%rdi,%rsi), %rax
     addq %rdx, %rax
     cmpq $5, %rdi
     jge .L2
     cmpq %rsi, %rdx
     jge .L3
     movq %rdi, %rax
     imulq %rsi, %rax
     ret
.L3:
     movq %rsi, %rax
     imulq %rdx, %rax
     ret
.L2
     cmpq $4, %rdi
     jle .L4
     movq %rdi, %rax
     imulq %rdx, %rax
.L4
     rep; ret
```

```
long fun (long x, long y, long z)
{
     long val -----------
     if(-----------)
     {
          if(-----------)
               val = -----------
          else
               val = -----------
     }
     else if(-----------)
          val = -----------

     return val;
}
```

QUESTION 3 - C code to x86 assembly

For the given for loop, fill in the blanks in the respective assembly code. Note that i is saved in edx and total is saved in ecx registers respectively.

```
for(i=0; i < total; i++)
     ct +=i;
```

```
xorl ----------, ----------- #here you need to initialize i to 0
cmpl -----------, ----------- # now it is time to compare i with total
jge .L4              #jump to end if necessary
movl ct, ----------- #cache value of ct in eax register

.L6:

addl -----------, ----------- #ct +=i is here
incl ----------- # i++
cmpl %ecx, %edx
jl .L6
movl %eax, -----------  #store the value of ct back in memory

.L4:
```

QUESTION 4 - C code to x86 assembly

Check the given fun.s assembly file and try to interpret its purpose. Note that if you try to run this file, it may not run properly due to your system architecture. Therefore, try to guess its purpose.

QUESTION 5 – Simple coding

In this question, you will implement recursive Fibonacci. Save your code in Fibonacci.c file. Then, create your corresponding assembly file Fibonacci.s by following the instructions we covered in the classroom. Depending on your system architecture, it may give different results. Write your guess how it should look like in x86 assembly and now compare it with your fibonacci.s file result. Write down the differences.

<u>How to submit:</u> All questions, except the 5th one, are written questions and can be submitted in a single doc or pdf file. You can even scan and submit your hand-written notes in a single pdf file. For question 5, you need to submit your .c and .s files and an explanation. Please include your written explanation in the previous file of the first 4 solutions. Additionally, do not forget to submit your code files. Therefore, you will submit one .zip file which includes: 1 doc/pdf file, 1 fibonacci.c file and 1 fibonacci.s file.