UNIVERSITY OF THE PHILIPPINES VISAYAS
COLLEGE OF ARTS AND SCIENCES
DIVISION OF PHYSICAL SCIENCES AND MATHEMATICS

CMSC 131
Introduction to Computer Organization and Machine Level Computing
A.Y. 2022 - 2023

Assignment Guide

Prepared by:

Jayvee B. Castañeda
Instructor

ACADEMIC INTEGRITY

As a student of the University of the Philippines, I pledge to act ethically and uphold the value of honor and excellence. I understand that suspected misconduct on given assignments/examinations will be reported to the appropriate office and if established, will result in disciplinary action in accordance with University rules, policies and procedures. I may work with others only to the extent allowed by the Instructor.

Laboratory Exercise #9

Reading

- Read Section 4.8 of Paul Carter's PC Assembly Book

Practice Exercise:

- Execute **"sub6.asm"** and interface it with **"main6.c"**. *What is the purpose of sub6.asm? What is the purpose of main6.c? Explain the relationship between them.*

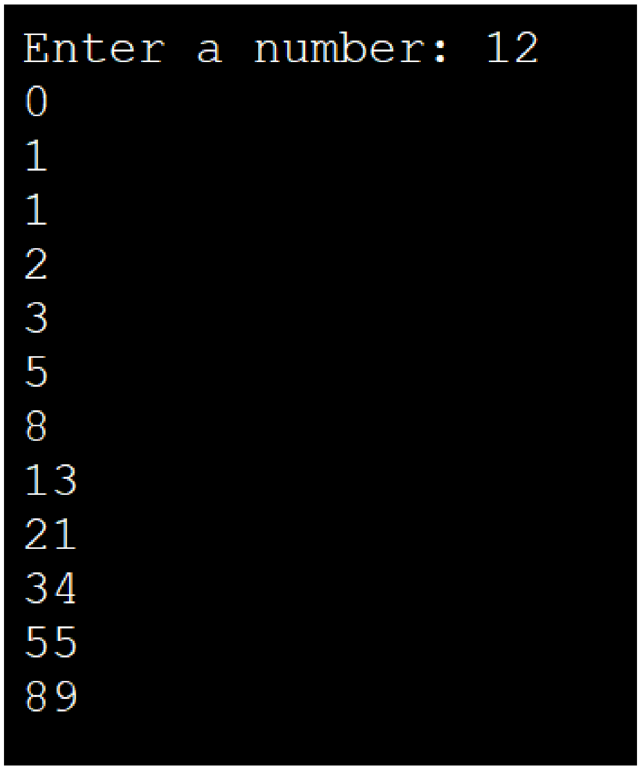
- Analyze the sample codes **sub5.asm** and **main.c**. *What is/are the stack register(s) used in the program? Explain the output of sub6.asm implementing stacks.*

Problem #9.

Teacher: Why are you doing your multiplication on the floor?
Student: You told me not to use tables.

Teacher: ??? …Well, this time, you will.
Student: :O

- Write an assembly program that ***prints the Fibonacci series.***
- Create a program named ***"fibo.asm" that computes for each of the Fibonacci numbers in the series***, which should interface with a file named **"main.c"**. *You are free to create your own "main.c" file. You may use the main.c in the previous lab exercises as reference.*
- The **"main.c"** should call an assembly subprogram named **"fibonacci", which solves and provides the Fibonacci numbers** contained in "**fibo.asm"**. You are free whether to print the results in "main.c" or "fibo.asm" *as long as* ***fibo.asm interfaces with main.c.***
- The output of your program should be something like this:



```
Enter a number: 12
0
1
1
2
3
5
8
13
21
34
55
89
```

- A good programming practice is to ***write comments on important line of codes*** for readability and documentation.

- Save both files *“main.c”* and *“fibo.asm”* in a compressed zip file called ***SurnameFirstLetterOfFirstName_lab8.zip*** in camel case. For instance, if your surname is “Juan Dela Cruz”, submit it as follows:

  DelaCruzJ_lab8.zip

- Take a screen recording of your working code and make sure to **record a video explaining each line of your code** as well as showing the correct output of your code. Use screen recorder application in Ubuntu (https:/itsfoss.com/best-linux-screen-recorders/) or Windows (https://atomisystems.com/screencasting/record-screen-windows-10/)

Submission Requirements:

1. **Program Codes Zip File (‘.zip file)**
2. **Screen Recorded Defense Video**

DEADLINE: December 15, 2022, 11:59 PM

Rubric for Programming Exercises				
Program (50 pts)	**Excellent**	**Good**	**Fair**	**Poor**
Program Execution	Program executes correctly with no syntax or runtime errors (9-10)	Program executes with minor (easily fixed) error (4-8)	Program executes with a major (not easily fixed) error (2-3)	Program does not execute (0-1)
Correct Output	Program displays correct output with no errors (9- 10)	Output has minor errors (6-8)	Output has multiple errors (3-5)	Output is incorrect (0- 2)
Design of Output	Program displays more than expected (7-8)	Program displays minimally expected output (5-6)	Program does not display the required output (3-4)	Output is poorly designed (0-2)
Design of Logic	Program is logically well-designed (9-10)	Program has slight logic errors that do not significantly affect the results (6-8)	Program has significant logic errors (3-5)	Program is incorrect (0-2)
Standards	Program is stylistically well designed (6-7)	Few inappropriate design choices (i.e., poor variable names, improper indentation) (4-5)	Several inappropriate design choices (i.e., poor variable names, improper indentation) (2-3)	Program is poorly written (0-1)
Documentation	Program is well documented (5)	Missing one required comment (4)	Missing two or more required comments (2- 3)	Most or all documentation missing (0-1)