

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 #1

### Reading

- Read [Section 2.1 of Paul Carter's PC Assembly Book](#)

### Practice Exercise:

- Assemble the assembly code (**math.asm**). This will create an object file (**math.o**) for math.asm.

**nasm -f elf math.asm**

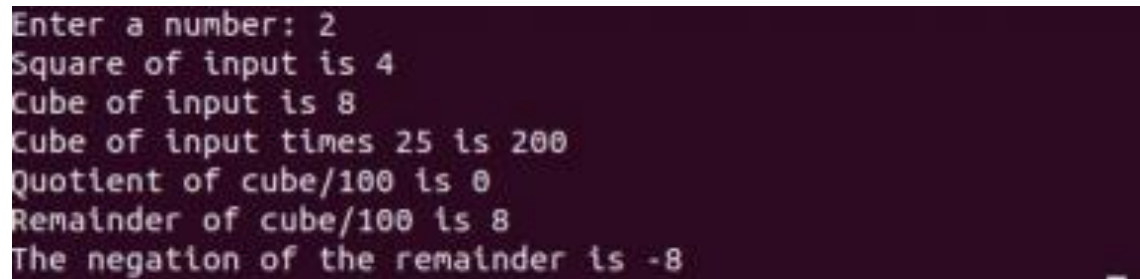
- Compile and link the assembly code with the C program (**driver.c**). In our machine, we will be using 32-bit registers thus we specify “-m32”.

```
gcc -m32 -o math driver.c math.o asm_io.o
```

- Execute the assembly code.

```
./math
```

The code should show the following:



```
Enter a number: 2
Square of input is 4
Cube of input is 8
Cube of input times 25 is 200
Quotient of cube/100 is 0
Remainder of cube/100 is 8
The negation of the remainder is -8
```

- Analyze the assembly code (first.asm). Reflective questions:

*What does the program do?*

*How does “mul” and “imul” instructions differ from each other?*

*How does “div” and “idiv” instructions differ from each other?*

## Problem #1.

You are a traveler from a faraway land and are in search for your long-lost sibling. You meet a flying companion named “Paimon” who always seeks out for food and treasure. One day, while discovering new locations in Mondstadt, you find a treasure chest, seeming left by the Treasure Hoarders out in the open. Strange enough, you come close to it where you found out it is locked using a 5-digit number code. Paimon became frustrated and nagged about not being able to open the code. However, you are a skilled programmer with amazing NASM knowledge, and you decided to solve for the code. There is a clue written on the side of the chest that reads:

**The code uses each digit from 1-5 only once. The first two digits add up to 8. The difference of the second and fifth digits is equal to the fourth digit. The middle digit is the quotient when the product of the first and last digit is divided by 6. Get the code and you get the treasure.**

- Write an assembly program that solves the code using the riddle above.
- Use different variables to represent the values of each digit.

- Your assembly program should solve the above problem using arithmetic operations in assembly (*add, sub, mul, div etc.*).
- Print the values of all digits and the final code. The output of your program is something like this:

```
PROBLEM: The code uses each digit from 1-5 only once. The first two digits add up to 8. The difference of the second and fifth digits is equal to the fourth digit. The middle digit is the quotient when the product of the first and last digit is divided by 6. Get the code and you get the treasure.
```

```
ANSWER:
```

```
The first digit is []
The second digit is []
The third digit is []
The fourth digit is []
The fifth digit is []
```

```
The code is [] [] [] [] []
```

- The brackets “[]” above should be replaced by the correct answer that your program outputs.
- A good programming practice is to *write comments on important line of codes* for readability and documentation.
- Save your program in a file called *SurnameFirstLetterOfFirstName\_lab1.asm* in camel case. For instance, if your surname is “Juan Dela Cruz”, submit it as follows:

*DelaCruzJ\_lab1.asm*

- 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 Code (‘.asm’ file)
2. Screen Recorded Defense Video

**DEADLINE: September 29, 2022, 11:59 PM**

<b>Rubric for Programming Exercises</b>				
<b>Program (50 pts)</b>	<b>Excellent</b>	<b>Good</b>	<b>Fair</b>	<b>Poor</b>
<b><i>Program Execution</i></b>	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)
<b><i>Correct Output</i></b>	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)
<b><i>Design of Output</i></b>	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)
<b><i>Design of Logic</i></b>	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)
<b><i>Standards</i></b>	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)
<b><i>Documentation</i></b>	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)