

Homework 3 – Integer and Floating Point Arithmetic Using the Stack

Objectives:

This exercise aims to give you practice with the floating-point unit and the stack.

Description:

Write a MASM program to perform the following tasks:

1. Display the program title and your name.
2. Then get the user's name and greet the user.
3. Prompt the user to choose between integer or floating-point arithmetic.
4. Using a counted loop, prompt the user to give 5 integer or floating-point numbers based on the choice they did. Push the numbers into the integer or floating-point stack based on the choice they did.
5. Calculate and display the sum and the average of the numbers entered.
6. Ask the user if they want to perform another calculation or exit.
7. When the user chooses to exit the program, display a goodbye message that includes the user's name and terminate the program.

Requirements:

1. The main procedure must be divided into the following sections:
 - a. Introduction
 - b. User instructions
 - c. Get the data
 - d. Calculate sum
 - e. Calculate average
 - f. Display numbers
 - g. Choose between continue or exit
 - h. Say goodbye
2. The loop that gets the numbers from the user should be implemented as counted loop.
3. The program should not use any variables to store the numbers. Everything should be done using the stack and registers should be used only for the operations.
4. All the requirements regarding documentation, readability, user-friendliness apply.

Example execution (user input is in *italics*):

Integer and Floating-Point Arithmetic

Programmed by Leonardo Pisano

What's your name? *Paul*

Hello, Paul, do you want to perform integer or floating-point arithmetic? Please type "0" for integer or "1" for floating point arithmetic.

You choose to perform floating point arithmetic. Please enter 5 numbers

The sum of the numbers you enter is: *50.5*

The average of the numbers is: *10.1*

Do you want to perform another calculation? Please enter "0" for "no" or "1" for "yes".

Goodbye, Paul

Questions:

- 1) What is the address EIP has after the "loop" instruction? Do you expect EIP to have that address? Why?
- 2) What is the address EIP has after the user chooses to perform another calculation? Do you expect EIP to have that address? Why?
- 3) How does the address ESP change as the user enters the numbers for the integer arithmetic? How does it change when the operations are performed? Why?
- 4) How was the experience working with the stack? Did you struggle at all?

EXTRA CREDIT (+5 points):

1. Perform the floating-point calculation using the FPU and calculate the result. Showing all the steps you followed to construct the RPN notation.

$$Y = 2.5 + \frac{1.5 * 3.5}{2} / \frac{3 * 1.4 * (2.2 - 2)}{3!}$$

2. Do something extraordinary.

-ATTENTION-

To ensure you receive credit for any extra credit options you did, you must add one print statement to your program output PER EXTRA CREDIT which describes the extra credit you chose to work on. You will not receive extra credit points unless you do this. The statement must be formatted as follows,

--Program Intro--

EC: DESCRIPTION

--Program prompts, etc--

What to submit:

The .asm file with your code and a pdf with the answers to the questions