Computer Science 3B

Practical Assignment 08

Assignment date: 2024-10-03

Deadline: 2024-10-03 17h05

Marks: 70

This practical assignment must be uploaded to eve.uj.ac.za before 2024-10-03 17h05. Late[1] or incorrect submissions will not be accepted and will therefore not be marked. You are not allowed to collaborate with any other student. Plagiarism is not tolerated. All submissions are tested for plagiarism.

Good coding practices include a proper coding convention and a good use of commenting. Marks will be deducted if these are not present.

The reminder page includes details for submission. Please ensure that ALL submissions follow the guidelines. The reminder page can be found on the last page of this assignment.

BeatWatch is making progress with their capabilities and would like to handle Floating Point Units (FPU). Having learnt that you can process floating point numbers with the FPU, they ask that you create a task using your newfound knowledge.

Write an **80x86** assembly program with the following:

Main program structure (requirements):

1. Use *general-purpose **32-bit** registers* only (such as, **eax, ebx, ecx, edx, esi, <u>etc</u>**.)
2. Create *global variables* only. *No need for temporary/local variables, however you can use it if it contributes to your solution*.
3. Create the **inputArray** array holding seven integer numbers for the simulated **7** days rewards received for your fitness goals achieved (*see the table for examples provided, for now the currency is irrelevant*). There is *no need for user-input functionality, i.e., a static array will suffice as the focus is on being able to work with floating point numbers*.
4. <u>You must create **2**</u> functions to calculate the average simulated monetary rewards received for the fitness goals achieved:
    a. Function 1: **calculateIntAverage** to sum up the rewards received and compute the average. *Note the average does not need to be displayed it should be stored in the **intResult** variable*.
    b. Function 2: **calculateFloatAverage** to sum up the rewards received and compute the average. *The average should be stored in the **floatResult** variable.*

[1] Alternate arrangements for exceptional circumstances will be posted on eve.

Testing set – Use these values to test your program

Rewards for Fitness Goals	CPU Integer Daily Avg.	FPU Float Daily Avg.
[5, 15, 9, 24, 0, 4, 54]	15	15.85
[25, 60, 1, 4, 9, 14, 3]	16	16.57
[99, 80, 57, 0, 0, 0, 110]	49	49.42

Mark sheet

1. Design **[10]**
2. Function **calculateIntAverage** **[10]**
3. Function **calculateFloatAverage** **[20]**
4. Structure and layout (sufficient variables, relevant data types) **[05]**
5. Commenting **[05]**
6. Correct execution **[20]**

TOTAL [70]

NB

Submissions that **do not assemble** will be capped at 40%!

Practical marks are awarded subject to the student's ability to explain the concepts and decisions made in preparing the practical assignment solution.

(Inability to explain code → inability to be given marks.)

Execution marks are awarded for a correctly functioning application and not for related code.

Reminder

Your submission must follow the naming convention below:

`SURNAME_INITIALS_STUDENTNUMBER_SUBJECTCODE_YEAR_PRACTICALNUMBER`

Example: `Berners-Lee_TJ_209912345_CSC03B3_2024_P05`

Surname	Berners-Lee	**Module Code**	CSC03B3
Initials	TJ	**Current Year**	2024
Student number	209912345	**Practical number**	P05

Your submission must be **a single zip (compressed) file**!

Your submission must include the following:

File	Naming	Folder	Purpose
`Design`	`STUDENTNUMBER_P05.pdf`	`docs`	Contains your program design. All files must be in **PDF** format. Your details must be included at the top of any **PDF** files submitted[0].
`Source`	`STUDENTNUMBER_P05.asm`	`src`	Contains all relevant source code. Your details must be included at the top of the source code[0].

Multiple uploads

Note that only **one** submission is marked. If you already have submitted once and want to upload

a newer version then submit a newer file with the same name as the uploaded file in order to overwrite it.

[0]Failure to correctly indicate your details will result in a penalty.