Lab 5 Specification – Practicing Performance Assessment and Data Transformation
Due (via your git repo) no later than 8 AM, Thursday, 13th October 2022.
50 points

Lab Goals

- Practicing the performance assessment technique(s) discussed in class.
- Practicing the data transformation algorithms discussed in class.

Learning Assignment

If not done previously, it is strongly recommended to read all of the relevant "GitHub Guides", available at the following website:
`https://guides.github.com/`
that explains how to use many of the features that GitHub provides. This reading assignment is useful to understand how to use both GitHub and GitHub Classroom. To do well on this assignment, it is also recommended to do the reading assignment from the section of the course textbook outlined below:

- **PH Chapter 1.6, and 1.7**

Assignment Details

Now that we have discussed some more of Performance Assessment techniques, and Data Transformation algorithms, we are ready to practice these procedures to get comfortable in solving problems that are related to this context.

At any duration during and/or after the lab, students are recommended to team up with the Professor and the TL(s) to clarify if there is any confusion related to the lab and/or class materials. The Professor proofread the document more than once, if there is an error in the document, it will be much appreciated if you can communicate that to the Professor. The class will be then informed as soon as possible regarding the error in the document. Additionally, it is highly recommended that students will reach out to the Professor in advance of the lab submission with any questions. Waiting till the last minute will minimize the student's chances to get proper assistance from the Professor and the Technical Leader(s).

Students are recommended to get started with this part in the laboratory session, by discussing ideas and clarifying with the Professor and the Technical Leader(s). It is acceptable to discuss high-level ideas with your peers, while all the work should be done individually. Late submission is accepted for the part(s) in this section, based on the late policy outlined in the course syllabus.

It is required for all students to follow the honor code. Some important points from the class honor code are outlined below for your reference:

1. Students are not allowed to share their solution files and/or other implementation details. It is acceptable to have a healthy discussion with your peers. However, this discussion should be limited to sharing ideas only.
2. Submitting a copy of the other's program(s) and technical reports is strictly not allowed. Please note that all work done during lab sessions will be an opportunity for students to learn, practice, and master the materials taught in this course. By doing the work individually, students maximize the learning and increase the chances to do well in other assessments such as Quizzes, exams, etc · · ·

Section 1: Performance Assessment Problems

This section is worth 20 points. Each task is 10 points.

Task 1 → Overview

[Refer Example 3 and 4 in Week5/P2 slides to complete this section.]

Suppose the following C fragment is executed on a machine with 1.5 GHz clock rate:

```
int p = ((m+n)*(m-n))/(m*n)
```

———————————-

The instructions at the assembler level are as follows:

———————————-

```
load m
load n
add e1, m, n
sub e2, m, n
mul e3, e1, e2
mul e4, m, n
div p, e3, e4
store p
```

———————————-

Load and Store instructions take 5 clock cycles and other instructions like add, sub, mul, and div take 3 clock cycles.

First find the weighted average CPI. Next, find the total execution time taken to process the fragment by using the weighted average CPI computed previously. The answer provided should include the details similarly as provided in the examples 3 and 4 in the slides. Round off your CPU time to two decimal points and the output for the time taken should be indicated using nano seconds as the unit. Write down everything in your notebook first to make sure that the final answer is correct, and then post your answer to the **sol/problem1.md** file in the lab repository.

Task 2 → Overview

[Refer Example 5 and 6 in Week5/P2 slides to complete this section.]

Suppose a calculator program take a total execution time of 120s (2 minutes). The multiplication task take up 40s to execute, and the other tasks in the program take the rest of the time. We had been asked to evaluate the program from two perspectives.

(a) Is it possible to speed up all the tasks in the program except multiplication? If so by how much? Hint: execution time affected should be equal to 80s and not 40s, ask why to yourself!

(b) Is it possible to speed up the multiplication task only? If so by how much?

Assume that the overall goal is to make the program 2 times faster. Write down everything in your notebook first to make sure that the final answer is correct, and then post your answer to the **sol/problem2.md** file in the lab repository.

Section 2: Data Transformation Problems

This section is worth 30 points. Each subtask in Task3 is 5 points.

Data Transformations at the internal level is the conversion of data by computers at different levels of granularity. For example: data is displayed and represented in different ways, namely:

1. decimal for human users to read, and write (in code).
2. binary for machines to read and write.
3. hexa-decimal for managing memory, using a wider range of memory addresses.

Task 3 → Overview

[Refer to the files in the resources folder. These files include the complete solution for all the six problems posted in the Week6/P1 slides and the class participation activity. Also refer to the slides for mastering the algorithm and the technique used for each transformation.]

In this section, we will practice the data transformation schemes using the algorithms discussed in class. Write down everything in your notebook first to make sure that the final answer is correct, and then post your detailed answers by following the templated provided in the files within the resources folder.

Solve the following six problems:

(a) Convert $(11101000)_2$ to Decimal. Post your answers in sol/b2d.md file.

(b) Convert $(298)_{10}$ to Binary. Post your answers in sol/d2b.md file.

(c) Convert $(BEAF)_{16}$ to Decimal. Post your answers in sol/h2d.md file.

(d) Convert $(987654321)_{10}$ to HexaDecimal. Post your answers in sol/d2h.md file.

(e) Convert $(DECADE)_{16}$ to Binary. Post your answers in sol/h2b.md file.

(f) Convert $(10001111010111101011101)_2$ to HexaDecimal. Post your answers in sol/b2h.md file.

Start thinking about how to do these transformations within a C Program. The implementation of this set of transformations may be a course project option at the end of this semester. We will also get an opportunity to explore a few transformation(s) in the next lab assignment.

Submission Details

For this assignment, please submit the following to your GitHub lab repository.

1. updated, completed version of **problem1.md** file.
2. updated, completed version of **problem2.md**file.
3. updated, completed version of the six files namely: **d2b.md**, **b2d.md**, **d2h.md**, **h2d.md**, **h2b.md**, **b2h.md** files.
4. Add a Reflection to the repository by modifying the `reflection` file in the lab repository. List out the biggest learning points and any challenges that you have encountered during this lab.
5. It is highly important, for you to meet the honor code standards provided by the college and to ensure that the submission is completed before the deadline. The honor code policy can be accessed through the course syllabus. Make sure to sign the honor-code file.

Grading Rubric

1. Details including the points breakdown are provided in the individual sections above.
2. If a student needs any clarification on their lab credits, it is strongly recommended to talk to the Professor. The lab credits may be changed if deemed appropriate.