EECS 645: Computer Architecture

Computer Assignment #1

Due Date: 11:59 PM Feb. 16th

Computer Assignment Submission Instructions:

- Late submission policy: 10% penalty daily for a maximum of 3 days. After 3 days, you will get a zero.
- Completing the MIPS Programs:
  - You have been provided with partial MIPS code for various programs.
  - Your task is to complete each program by inserting the necessary MIPS instructions to ensure its correct functionality.
  - Pay careful attention to detail and follow the instructions provided in the comments.
  - Do not modify the existing code unless explicitly instructed to do so.
- File Naming Convention & Submission:
  - Create a ***zip*** file containing all your completed MIPS program files.
  - Ensure that the file name follows the specified format: ***LastName-FirstName.zip***.
  - Replace "***LastName***" with your last name and "***FirstName***" with your first name.
  - Submit the zip file through the Canvas.
- Program Functionality:
  - Each program should be functional and free of syntax errors.
  - **Programs that do not run or contain syntax errors will receive no points**.
  - Partial grading will not be considered; the entire program must be correct for points to be awarded.
- Instructions within the Code:
  - Do not modify the upper part of the provided code, as indicated by the comments.
  - Add your instructions where specified with the comment "***## Your instructions go here ##.***"
  - **Ensure that the results are stored in the designated register, as mentioned in the comments.**
  - Follow the given structure and maintain the integrity of the code.
- Please forward your questions to the GTAs:
  - Amin Mamandipoor – amin.mamandi@ku.edu
  - Johnson Umeike – johnson.chinedu@ku.edu

Please refer to the link below which will direct you to the programming assignments available on GitLab.
https://gitlab.ku.edu/a972m888/Computer-Assignment-1

Programs Descriptions:

00-Print: Write a MIPS assembly program that prints the string "Hello, MIPS!" to the console.

01-UserInput: Write a MIPS assembly program that prompts the user to enter an integer, reads the input, and then prints "You entered: <number>" to the console.

02-ArithmeticOps: Write a MIPS assembly program that performs arithmetic operations such as addition, subtraction, multiplication, and division on num1 and num2. The program stores the results in $t0, $t1, $t2, and $t3, respectively.

03-Conditions: Write a MIPS assembly program that determines if the two numbers stored in $t0 and $t1 are the same or different.

04-Loop: Write a MIPS assembly program that utilizes a loop to print the loop's iteration in the console.

05-Arrays: Write a MIPS assembly program that iterates through a predefined array, sums the array's elements, and prints out the sum of them.