Advanced MARS Programming
Due: February 6, 2024

INSTRUCTIONS

This assignment explores functions in MIPS. Submit your answers in a single ASM text file called **mini3.asm**. The teaching assistants will hold office hours the week of the 1st.

You are expected to do all your work on your own. Plagiarism and cheating are a serious offence. You may ask classmates, the Tas, and the professor clarification questions.

QUESTION

You must write and execute the code for this question in the MARS MIPS Emulator provided to you on our myCourses webpage under Content/Software. Your TA will only grade your work from the provided emulator.

Write the following program using MARS, your program will have a .data area and a .text area divided into a main program and a single function:

1. In the MIPS `main()` code area, prompt the user to input an integer number greater than or equal to zero. Say to the user: “Please input an integer value greater than or equal to 0: “.
2. If the user inputs a value less than zero, then terminate the program with the message: “The value you entered is less than zero. This program only works with values greater than or equal to zero.” Do not perform any other validation.
3. If the user inputs a value greater than or equal to zero, then call a function called **factorial** with the following C signature to calculate the factorial, **recursively**:
   (you must translate this signature into its MIPS version)
   (the MIPS factorial function is encapsulated in its own scope using the Stack)

   ```
   int factorial(int);
   ```

4. After the function **factorial** returns the answer to the **main** code area, from the **main** code area, print the following messages:
   a. “Your input:” and the value they entered from the keyboard. New line.
   b. “The factorial is:” and the value returned by the function factorial. New line.
5. Now, prompt the user to see if they would like to do this again. “Would you like to do this again (Y/N): “. Tell the user to input a single character ‘Y’ (capital letter Y) to do it again, all other characters will terminate the program. If the user inputs ‘Y’ then go to step 1.

WHAT TO HAND IN

A single ASM text file named **mini3.asm** with the answers to the above question. Submit this to the assignment box “mini 3” on myCourses.

HOW IT WILL BE GRADED

This assignment is worth 20 points.

Point Deductions:

- For not following submission instructions: -3 points
- Your program must execute to be graded (part marks will be given)
- Not using 3-column format: -2 points
- Not using $t, $s, $a, $v registers following scope convention: -2 points

Points Awarded:

- Step 1 is 2 points
- Step 2 is 3 points
- Step 3 is 10 points (5 points following Stack convention, 5 points for recursion)
- Step 4 is 2 points
- Step 5 is 3 points

TA GRADING INSTRUCTIONS

- **Grade the student’s program only with the MARS emulator provided from myCourses**.
- If the program does not run at all, give a grade of zero. Do not fix the code.
- 10% off per day late, max 2 late days.
- Award the score to the student by comparing their work with the solution sheet.
- If the student submits a waiver, do not deduct late points. Students can only submit a waiver once, so record their name in the waiver spreadsheet. If a student has previously submitted a waiver, then ignore this one. In your comments to the student tell them that you ignored their waiver.
- Remember to award grades proportionally, this means if the student got a question half correct then they receive half of the points.
- Note that this assignment explores scope rules. This means that the students must show proper function encapsulations using the run-time stack and the register conventions. This shows in the points awarded at Step 3 and the deduction points. Please make sure to look at the code to make sure that this was implemented correctly.
- Grade in the following way:
  - Run their program to make sure it works properly (award & deduct points)
  - Then, open the file to verify (a) the 3-column format, and then (b) that the encapsulation and register convention rules were followed/implemented (award & deduct points)
  - Last, adjust the grades when a waiver is present.