Department of Electrical and Computer Engineering
Rutgers, The State University of New Jersey

Computer Architecture and Assembly Lab
Spring 2021

Lab 5
RISC-V functions and arrays

Instructions

Please answer all the questions below. You need to use the Venus RISC-V simulator for running and testing your code. Note that the simulator is 32-bit, and we do not consider overflow here.
Upload your lab report with the department cover page and your source code using Sakai.

Exercises

1. [30 pts] Write a RISC-V program in Venus simulator that accepts an input integer x and uses two methods to compute a factorial:

- Recursive method: $f(x) = x * f(x - 1)$

- Iterative/loop method: $f(x) = x! = x * (x - 1) * (x - 2) * ... * 2 * 1$

You can assume x is always a positive number.

In the program, please have a **main** function that takes the value of the input, performs the factorial computations by the two methods, and prints the outputs of the two methods in the console.

- Verify your program for 3 different input values x_1, x_2, and x_3 such that $x_1 \neq x_2 \neq x_3 \neq 0$.

Note: you need to provide your own inputs and show screenshots of the outputs based on the given inputs. (Each method and function worth 10 pts.)

2. [30 pts] Write a RISC-V program in Venus simulator that splits the given array {4, 37, 0, 12, 1, 0, 6} into the following three sub-arrays:

- Array 1: the elements are the odd positive numbers

- Array 2: the elements are the even positive numbers

- Array 3: the elements are all zeros

In the program, please have a **main** function that obtains the value of the input, splits the array, and prints the three sub-arrays in the console. Your implementation should work for arrays of different values as well.

Note: you need to show screenshots of the outputs based on the given input. (Each array worth 10 pts.)

3. [40 pts] Write a RISC-V program in Venus simulator that computes matrix multiplication. You can assume all the entries in each matrix are positive numbers. In the program, you need to consider the following cases:

- When the number of rows for matrix A and the number of columns for matrix B are equal, compute the multiplication and obtain the product matrix C: $C = A * B$.

- When the number of rows for matrix A and the number of columns for matrix are not equal, return the error code 99. Hint: please refer to the Ecall wiki page about returning the error code (https://github.com/kvakil/venus/wiki/Environmental-Calls).

In the program, please have a **main** function that checks the condition, perform the appropriate computations, and print the matrix C or the error code in the console.

- Verify your program for 3 input matrices m_1, m_2, and m_3 such that the dimensions of m_1, m_2, and m_3 are all unique.

- Demonstrate each possible case at least once.

- Your implementation should work for matrices of different values as well.

Note: you need to provide your own inputs and show screenshots of the outputs based on the given inputs. (The **main** function worth 15 pts, and the multiplication function worth 25 pts.)