
ACCESS TO INTERNET/ONLINE MATERIALS DURING THE EXAM IS STRICTLY NOT ALLOWED

There are two parts to the lab exam - the first part involves writing an assembly code and the second part involves writing a C/C++ code.

Part-1: Assembly code for GCD (Greatest Common Divisor) calculation: [6]

We need to calculate the GCD of multiple groups of two 64-bit unsigned numbers. The set of input numbers are provided as dwords in the data segment and the GCD computed from your code should be present starting from address 0x10000100 in the memory (data segment base + 0x100), as shown below. The first word in the data segment indicates the total group of input numbers (total number of gcd computations to be done) followed by the actual input numbers.

You can use any instruction from the base class of RISC-V, including any pseudo instructions for your implementation. **NO** mul, div, rem, float instructions are allowed.

The logic to find gcd should be implemented as a separate function/macro following proper caller-callee conventions. **20% marks will be deducted** if this not followed.

```
.data
.dword count_of_gcd, input11, input12, input21, input22, input31, input32, ...

.text
#your code starts here

#The final result should be in memory starting from address 0x10000100
#The first dword location at 0x10000100 contains gcd of input11, input12
#The second location from 0x10000100 contains gcd of input21, input22, and so on.
```

Example: If the data section contains .dword 3, 12, 3, 125, 50, 32, 16 - it indicates that we have 3 sets of gcd to be calculated: (12, 3); (125, 50); (32, 16).

After executing your code, memory location from 0x10000100 should contain dwords like 3, 25, 16

A representative GCD algorithm (any other algorithm can also be used within the allowed instructions):

```
gcd(a, b)
while a ≠ b
  if a > b
    a := a - b
  else
    b := b - a
return a
```

Submission instructions:

1. File should be named LABEXAM1_ROLLNUM.zip (LABEXAM1_CSYYBTECHXXXXX.zip)
2. It should contain a report.pdf and the assembly code (main.s)
3. report.pdf should contain your implementation details and inputs used to verify the code
4. Submit on moodle on the link corresponding to the Lab Exam (assembly)
5. The code should be working in RIPES simulator (64-bit RISC-V CPU) without any changes
6. The input and output format/register should exactly match as specified
7. Verify your code properly with various input combinations
8. You can use code from your previous submissions, if needed

Part-2: C/C++ code for floating point addition without using single/double float data type: [7]

Given two numbers in hex format (32-bit standard single float format), write a C/C++ program to add these two numbers and compute the result in 32-bit single format hex format. Rounding should be done by truncating any extra bits.

Develop the entire code as a single C/C++ file, do not split into multiple files.

The input numbers to be added are provided in a file named input.txt (with each line containing two numbers) as shown below:

2A0B2000, DBE239F2
700D3400, 200B2D67

...

7412DE40, 921C2A81

The output hex values (sum of the two values) for each line can be printed on the screen like below (values shown are not correct answer for given inputs):

DBE239F2
921C2A81

...

200B2D67

Submission instructions:

1. File should be named LABEXAM2_ROLLNUM.zip (LABEXAM2_CSYYBTECHXXXXX.zip)
2. It should contain a report.pdf and the C/C++ code (main.c or main.cpp)
3. report.pdf should contain your implementation details and inputs used to verify the code
4. Submit on moodle on the link corresponding to the Lab Exam (C/C++)
5. The code should work on Ubuntu system without any changes
6. The input and output format should exactly match as specified
7. Verify your code properly with various input combinations.
Suggestion/Hint: You can write a separate program by using float datatype to verify your generated results (this program does not need to be submitted). One example was also discussed in class.
8. You can use the code to parse/read input file from your previous lab submissions

ACCESS TO INTERNET/ONLINE MATERIALS DURING THE EXAM IS STRICTLY NOT ALLOWED
