

Lab 2

Due: @11:55 PM, Wed September 11th via [Gradescope](#) and [Autograder](#)

The following assignment is intended to be completed during your assigned lab period. Working in a group is optional for this week. If you work in a group, one member of your group must submit the assignment to Gradescope by the posted deadline and indicate your group members when submitting the assignment.

This assignment has 50 points but will be graded out of 40 (capped at 100%)

Group names and usernames

Group Member Name	Username
Qiulin Fan	rynnefan
Zifei Bai	zifeibai

Problem 1: Short answer [14 points]

Indicate whether or not each of the following is a valid advantage of increasing the number of registers in an ISA while keeping everything else (including overall instruction size) constant, and give a brief (30 words or fewer) justification for each answer. Partial credit may be given for incorrect answers with well reasoned explanations. **[3.5 points each]**

1. ***Increasing the number of registers generally reduces the number of instructions needed to implement a C program.

Yes, because increasing the number of registers can decrease the number of instructions like load and store as more words can be kept in registers rather than have to commute with memory.

2. ***Increasing the number of registers reduces the number of memory accesses (i.e. loads and stores) needed to implement a program.

Yes. Because more registers allow storing more different information into different registers instead of loading and storing for different variables. This can reduce the number of instructions that commute data between memory and registers.

3. Increasing the number of registers reduces the size of each instruction.

No. Increasing the number of registers cannot reduce the size of each instruction, but can only increase the size of instruction encoding since more registers will need more bits to refer to a register in an instruction.

4. Increasing the number of registers allows for more operands to be used in a single instruction.

No. Increasing the number of registers does not increase the number of operands per instruction. The number of operands of an instruction is only decided by how the ISA is implemented.

Problem 2: C-Code [18 Points, [Autograded](#)]

Implement the function `numHighBits` and submit it to the autograder. It should return the number of bits that are set to one in the binary representation of the input number. You should use [sizeof](#) to determine how many bits the input is on the given system

```
int numHighBits(int number);
```

Starter code: `wget eecs370.github.io/labs/lab2.tar.gz`

Problem 3: LC2K Assembler Test Cases [18 points, [Autograded](#)]

For this problem, your group must submit test cases that expose at least 3 of the instructor bugs in project 1a. Once you have written test cases that expose the bugs, you must write the corresponding correct machine code output and submit it to the autograder for full credit. In addition to the constraints listed in the project, each LC2K program you write must be limited to **5 lines** or fewer. No test cases should cause errors on a correct assembler. Each submission will be limited to 3 test cases, but fewer may be needed. Each output file name must be the same as the assembly, with the extension changed to `.mc`.

You are encouraged to use (and submit) these test cases if you are still working on P1a. A great strategy is to run these test cases on your assembler and "diff" your output with the correct output any time you make a change.