CprE 381 Homework 4

[Note: This homework gives you some more practice with MIPS programming and thinkingabout testing your processor. As a happy coincidence, it will help provide a class-wide test program that I will share with you. Then the homework begins to cover processor design choices.]

1. Exam 1 Rework
   Rework and submit your worst graded problem (i.e., 2, 3, ..., 12) from Exam 1. I leave it up to you to select an appropriate problem where you had a genuine misconception. You must rework the entire problem and submit it as a legible, one- or two-page pdf. **Include a description of your misconception.** This problem's grade will be the percent score from your best revised problem. Since you have time and the ability to access resources, partial credit will be less available. In particular, if I made comments on your question and took off few to no points, there may still be mild errors in your solution! Please ask questions in OHs, before class, and on the Homework Help channel to make sure you understand the problem.

2. Instruction Testset
   As a course, we will be constructing a set of instruction "unit tests" for our processor. Each of you will be responsible for one instruction's unit tests. Write a set of MIPS code sequences to stress test one of the MIPS instructions you are implementing for your projects (i.e., it should test the range of operand values, any exceptional behavior, and any other edge cases you can think of suchas overflow). IMPORTANT: since these are aimed at being self-contained tests, use as few other instructions as possible. When you need to use other instructions, only use instructions above yours on the list (i.e., if you debug instructions in list order it will be reasonably likely that the instruction you are unit testing is the one failing), There should be ***at least*** three such test programs, each in their own assembly file whose name is your "<your instruction mnemonic>_<your NetID>_<test_number>.s".

   In addition to your instruction under test, you may only use those instructions above yours in the list. In general, use as few other instructions as possible (we wouldn't want to see any false failures). In the comments, you ***must*** justify whyyou are including each specific test case and why the test has value (e.g., what edge case it covers or what common case it covers). I have included an exampletest file for you to use as a template. You don't necessarily need to use as manyinstructions as I have for this test (part of my purpose was checking all registerscould be cleared).

   Note that these files will be used to help test your implementations. As such, upload your function in a separate .s file. The instruction you are implementing

is listed below:

Last Name Starts	Instruction to Stress Test
A	**addi**
Ban, Bark	**lui**
Barn, Bu	**addiu**
Ca, Chi	**addu**
Chr, Den	**and**
Dev, Dr	**andi**
Du, Dw	**add**
Eg, Ek	**lw**
El	**nor**
F	**xor**
Ga, Gel	**xori**
Gey, Gi	**or**
Go, Gra	**ori**
Gri, Gru	**slt**
Ha	**slti**
He, Hi	**sll**
Ho, J	**srl**
Ke, Ki	**sra**
Kl, Kos	**sw**
Kot, La	**sub**
Li, Lu	**subu**
Ma	**beq**
Mc, Me	**bne**
Mo, Mu	**j**
N, Oc	**jal**
Ob, Oh	**jr**
Ot, Pa, Pe	**lb**
Po, Pr, Ri	**lh**
Ro, Ry	**lbu**
Sa, Sc	**lhu**
Sh, Si	**sllv**
Sz, T	**srlv**
V, W	**srav**