Homework#2 for ECE 152
Instruction Sets (Chapter 2)

Due in class on February 6

All homework must be done in a group of 2 students. Each group should turn in one hard-copy in class. If your handwriting is unreadable, please type your homework.

1) [5 points] Patterson & Hennessy 2.10.1

2) [5 points] P&H 2.11.5

3) [10 points] For a stack-based ISA, write (on paper) an assembly program that computes X = A*[B + C*(D+E+F)]. Assume that you have the following instructions:

```
push A --> puts A on top of stack
pop A --> removes A from top of stack
add --> consumes top two items of stack and puts sum on top of stack
mult --> consumes top two items of stack and puts product on top
```

4) [30] Write a MIPS assembly program that determines whether a given integer is prime. Assume that the integer is in the range between zero and 511.

Use the spim simulator (available on the textbook's CD - please refer to page B-42 of the textbook for more information) to run and test your assembly program. Spim (and xspim) is a program that simulates the behavior of MIPS32 computers and can run MIPS32 assembly language programs. Documentation for spim is available in Appendix B of your textbook and at: http://www.cs.wisc.edu/~larus/spim.html. This spim website also contains a link for downloading a PC version of spim, if you'd rather run it on your PC than on a dsil workstation. A helpful reference is a simple program that I've provided for you at: http://www.ee.duke.edu/~sorin/ece152/resources/simple.s. This simple program sums the entries in a list of 9 integers.

To submit your code for this question and the next question, please send an email to duke.ece.152.spring2012@gmail.com with the program files and the names and NetIDs of the two students who worked on it.
This email must be received by 10:00am on the due date.

[30] Write a MIPS assembly program that counts the number of instances of the word "Duke" in a stream of characters that are inputted. The pseudo-code for your program is below. Once again, use spim to run and test your code. Your program will be tested by the grader with several different arrays of characters, to make sure that your program does indeed work correctly. **You absolutely, positively MUST use procedures for the two functions. I want to see you use procedures and stack frames. If you don't use procedures, you will not receive much credit for your work, even if the end result is functionally correct.**

```
main:     read one char;
          while (char != carriage return){
                call areLastFourCharsDuke(char*);
                read one char;
          }
          call printDuke();

// areLastFourCharsDuke() is a function that returns 1 if the last
// four input chars are “Duke” and 0 otherwise. Its input argument is
// a pointer to an array holding the last four chars inputted.

// printDuke() should print the word “Duke” one time for each time
// it occurred in the input string

// You ABSOLUTELY MUST have procedures for areLastFourCharsDuke()
// and for printDuke().
// I want to see you use procedures and stack frames!!
```