Lab 6

Back to MIPS!

Hand this sheet in at the end of class; make sure I've checked off at least one problem.

1. If you haven't already done this in a previous lab, prompt the user for a floating point value and display it.

2. Write a program that accepts three single precision floating point values from the user and calculates their single precision average.

   ________ Show me your result.

3. Hmm, what about character input? Write a program that continually prompts the user for a character. The program should stop when the character input is letter 'A'. When this happens, display the message: "Match!"

   This is slightly tricky, as a character is only one byte. However, everything in MIPS is in terms of 4 bytes = 32 bits. Therefore, you need to get to a single byte of a 32-bit word. The instruction `lb` can help you out here.

   You can store the 'A' to compare to the input in the data section as a single byte:

   `A: .ascii "A"`

   ________ Show me your result.

4. Prompt the user for a floating point number. Write a procedure that is passed the floating point value and returns an **integer** rounded to the nearest whole number. The main program should then display the rounded integer.

   You may find the `mfc1` instruction helpful.

   ________ Show me your result.

5. Modify the program in #4 so that your program displays the floating point rounded to the nearest hundredth. For example, 3.975 should round to 3.98.

6. Write a program that accomplishes the same thing as the following C++ code:

```
int loc;
float array[8] = {1.1, 2.2, 3.3, 4.4, 5.5, 6.6, 7.7, 8.8};

cout << "Value is: " << array [0] << endl;
cout << "Address is: " << &array [0] << endl;
cout << "Enter loc: ";
cin >> loc;
cout << "Value is: " << array [loc] << endl;
cout << "Address is: " << &array [loc] << endl;
```

   Do the addresses make sense?

Name: ______________________________