CSCI 2500: Computer Organization

Homework 3 MAL Programming

Date Assigned: Mar. 4, 2015 **Due date:** Mar. 18, 2015

This is a *team* project (pair programming, except for students who opt to work on their own). The deadline for this assignment is **4 PM, Wednesday, March 18, 2015**. The assignment will not be accepted after **4 PM, Thursday, March 19, 2015**.

Important notes: This assignment also has two parts and both parts must be done in MIPS Assembly Language (MAL). Therefore you will be submitting two MAL source files for this assignment by the deadline specified above. The names of the two files that you submit must be `hw3_1.s` and `hw3_2.s`.

The total grade for the assignment is 100 points, with 35 points for Part 1 and 65 points for Part 2. Your programs must be well documented. Please refer to the MIPS examples posted on LMS to understand how your MAL programs should be documented.

Part 1

For this part, your MAL program must be in a file named `hw3_1.s`. Each integer is represented using 32 bits in MIPS. Recall from our discussion of number systems that bits are numbered right to left from 0 to 31 (i.e. the rightmost bit is numbered 0 and the leftmost bit is numbered 31). Given this numbering convention, bits 0 through 15 form the right half and bits numbered 16 through 31 form the left half of the integer.

You are required to write a MAL program that prompts a user for a positive integer (>0), reads the integer typed by the user and outputs the following four computed values:

- The total number of 0's in the right half of the binary representation of the integer
- The total number of 1's in the left half of the binary representation of the integer
- The highest power of 4 that evenly divides the integer
- The value of the smallest digit in the decimal representation of the integer

Example: Suppose the user inputs the decimal integer 147456. The 32-bit binary representation of 147456 is:

```
0000 0000 0000 0010 0100 0000 0000 0000
```

For the above example, the correct answers are as follows:

- The number of 0's in the right half of the binary representation of the given integer = 15
- The number of 1's in the left half of the binary representation of the given integer = 1
- The largest power of 4 that evenly divides the given integer = 7
- The value of the smallest digit in the decimal representation of the given integer = 1

Note: The largest power of 2 that evenly divides 147456 is 7. This is because 147456 = 9 × 16384 = 9 × 4^7. The highest power in this case represents the number of 0 pairs at the end of the binary representation of 147456 (Recall: each pair of bits represents a power of 4).

Program outline: The outline for your program for Part 1 must be the following.

1. `Prompt the user for a positive integer`
2. `Read the integer from the user`
3. `Compute the four values requested above and print the answers`
4. `Stop`

Each time your program for Part 1 is executed, it should handle just one integer. You may assume that the value typed by the user is a positive decimal integer – there is no need to do any error checking on the user input.

Part 2

For this part, your MAL program must be in a file named `hw3_2.s`. It must have at least one function in addition to the main program. You may make the following assumptions for this part of the assignment:

- Any line of text typed by a user has at most 100 characters including the newline character
- A whitespace character refers to a space, a tab or the newline character
- A word is any sequence of characters that does not contain a whitespace character
- The input search pattern is no more than 11 characters including the newline

In this part, you are required to write a MAL program that prompts the user for a line of text and reads the line typed by the user. The program will need to then prompt the user for a pattern to match (Note: this is not a regular expression just a simple text pattern). If the input line contains only white space characters your program should just output a message similar to "Line contains whitespace characters only!" and stop. Otherwise, your program should compute and output the following:

- The number of whitespace and non-whitespace characters in the line
- The number of words in the line
- A "yes" or "no" message indicating if the pattern the user entered was found in the main input line

Example Program Usage (User typed input is in red):

```
Enter a input line: C programs don't always do what you want but they always do something
Enter a pattern to search for: always
Results:
# of whitespace characters: 13
# of non-whitespace characters: 57
# of words: 13
The user pattern was found within the input line


Enter a input line: The road to heck in C is paved with dangling   pointers
Enter a pattern to search for: Java
Results:
# of whitespace characters: 13
# of non-whitespace characters: 43
# of words: 11
The user pattern was NOT found within the input line


Enter a input line: The Java language is object oriented.  Java is also coffee!
Enter a pattern to search for: Java
Results:
# of whitespace characters: 11
# of non-whitespace characters: 49
# of words: 10
The user pattern was found within the input line

Enter a input line:
Line contains whitespace characters only!
```

Program outline: The outline for your program for Part 2 must be the following.

1. Prompt the user for a line of text
2. If the line has only whitespace characters then print the message “Line contains whitespace characters only” and then stop the program
3. Prompt the user for an pattern to search for in the input line
4. Compute the three quantities requested above and let the user know if the search pattern was found in the input line
5. Stop

Each time your program for Part 2 is executed, it should handle just one line of input and one pattern to match. Please note that no error checking is needed except for checking whether the input line consists of only whitespace characters.