

ICS 233 – Fall 2010

Computer Architecture and Assembly Language

Programming Assignment 2

Problem 1:

Write and test a MIPS assembly language program to **count the letters** in a text file and **sort them according to their frequency**. The program should do the following:

- Open a text file and read all characters into an array. The maximum number of characters to be read should be limited to the size of the array, which should be 100,000 characters. MARS provides the system calls for opening a file, reading from a file, etc.
- Traverse the array character by character. Count only the letters. You should have **an array of 26 counters**, one for each letter. Each entry in the array should store a letter and its corresponding counter value. Do not distinguish between a capital or lowercase letter. It should be counted as the same letter. Other characters (spaces, commas, periods, parentheses, digits, etc.) should not be counted.
- **Sort the letters** according to their counter value. For example if letter E has the highest counter value then it should appear on top of the sorted array.
- Display the sorted array, starting with the letter that appeared most frequently in the text file and its counter value.

Your program should be well divided into procedures. A sample run is shown below:

Enter input text filename: **input.txt**

Letter Count (sorted)

E	267
A	209
S	130
.	
.	
X	5

Problem 2:

Write a **recursive function** to compute the **greatest common divisor** of two integers as follows:

gcd(a, 0) = a

gcd(a, b) = gcd(b, a%b) // a%b is the remainder of dividing a/b

For example: **gcd(8, 12)=gcd(12, 8)=gcd(8, 4)=gcd(4, 0)=4**

Ask the user to enter two integer values, compute and print the greatest common divisor. Make sure to implement the recursive function properly using the runtime stack.

Submission Guidelines:

All submissions will be done through WebCT.

Submit the source code of the program. Make sure that your program is well documented.

Grading Policy:

The grade will be divided according to the following components:

- Correctness of code: program works properly and produces correct results
- Design and Coding: program is well designed and divided into procedures
- Documentation of code: program is well documented