COE 301 / ICS 233 – Computer Organization

MIPS Programming Assignment 1, Term 181

Problem 1: Counting Letters and Digits in a Text File

Write and test a MIPS assembly language program to **count each letter and digit** in a text file. The program should do the following:

a) Open a text file **`"input.txt"`** and read all characters into an array in memory. Limit the length of the array to 10000 characters. The maximum number of characters to be read should be 10000 characters. MARS provides the system calls for opening and reading a text file. If the file does not exist then the program should display an error message and terminates.

b) Traverse the array character by character. Count all letters and digits. You should use **an array of 26 counters** for the 26 letters in the alphabet, and **one counter** for all digits in the file. Do not distinguish between capital and lowercase letter. It should be counted as the same letter.

c) Display the count of each letter as well as the total digit count. A sample run is shown below:

```
Letter Count
A      67
B      18
C      23
.
.
Z      5

Total digit count = 39
```

Problem 2: Traversing and Displaying Elements of an Integer Square Matrix

Write and test a MIPS assembly language program to **traverse and display some elements of an integer square matrix provided by the user**. The program should do the following:

a) Ask the user to enter the square matrix size *N* and allocate the matrix dynamically. The matrix must have *N* rows by *N* columns. If the user enters a value of *N* which is less than 2, then reject it and ask the user to re-enter the value *N*.

b) Ask the user to input all the matrix integer elements. Each integer input should be entered on a separate line.

c) Display a menu of options:

1: Display a specific row
2: Display a specific column
3: Display the main diagonal
4: Exit the program

d) Ask the user to enter his menu option. If the user chooses either option 1 or 2, then the user is requested to input the desired row/column index number, with the first row/column index

being 0. If the user enters a row/column index number greater or equal to *N*, then an error message should be displayed and the menu should be redisplayed.

e) If the user enters an option other than 1 to 4 then an error message should be displayed and the menu should be redisplayed. The program should terminate only when the user enters option 4. Otherwise, it should repeat displaying the menu and executing the requested option.

Submission Guidelines:

This assignment can be solved individually or in groups of two students only. No group should have more than two students. **At the beginning of your program, write the names of the two students who worked on the program.** If this program was solved individually then write your name only. The rest of the code should be well written and well document.

All submissions should be done through Blackboard. Submit the source code of the program. **If the assignment was solved by two students, then it is sufficient for one student to submit the assignment. The other student can write a note on Blackboard indicating his partner.** The program will be graded according to its correctness and documentation. It is your responsibility to make sure that the program works. A program that does not assemble or run will receive zero on correctness.

Grading Scheme:

Problem 1:	**[10 points]**
Opening and reading text file: "input.txt"	**[1 point]**
Displaying an error message when the file does not exist	**[1 point]**
Counting letters into an array of 26 counters and displaying them	**[5 points]**
Counting and displaying total digits	**[2 point]**
Program readability and comments	**[1 point]**
Problem 2:	**[10 points]**
Reading *N* and Dynamic allocation of the matrix	**[1 point]**
Reading all the matrix elements	**[1 point]**
Displaying Menu of options repeatedly and proper exiting	**[2 points]**
Displaying a Specific row	**[2 points]**
Displaying a Specific column	**[2 points]**
Displaying main diagonal	**[1 point]**
Program readability and comments	**[1 point]**