22C:060: Computer Organization

Homework 2

Total points = 40

Assigned September 18, 2012, due September 27, 2012, 11:59: 59 PM

Be generous about using comments to improve readability. Ideally you should add a comment with each line of your program. Insufficient comments will lead to loss of grade. Include a comment at the beginning specifying the purpose of the program.

You should turn in an executable program with adequate comments about the use of the registers and the strategy that you used to solve the problem. To submit the program, *zip* (or *tar*) them into a single file, and submit your solution through ICON dropbox.

Problem 1. **(10 points)**
Write a program to reverse a string of characters from a string input.txt, and print it. Thus if the input string contains

My computer broke down

Then the output should be

nwod ekorb retupmoc yM

[Hint: use a stack]

Problem 2. **(30 points)**

Write a program that reads a character string myinput.txt and does the following:

(a) Counts the frequency of each character appearing in the text (excluding blanks spaces and punctuation symbols), and records its count. Avoid the use of capital letters in the input string.
(b) Prints the character that occurred most frequently in the text, along with the number of occurrences of that character. Ignore the blank spaces.
(c) If there are multiple characters in this category, then print the one that ranks the highest in the alphabetical order (thus, if t, e, m occur 25 times each, then, the output should be e <space> 75)

To solve this,

(1) First create in the memory a table (see Figure 1) containing an array of entries <character, count>, and initialize it to (a,0), (b,0), (c,0) … (z,0). Each entry will consist of two bytes: one for the character, and the other for its count. Clearly, the count will not exceed 255.

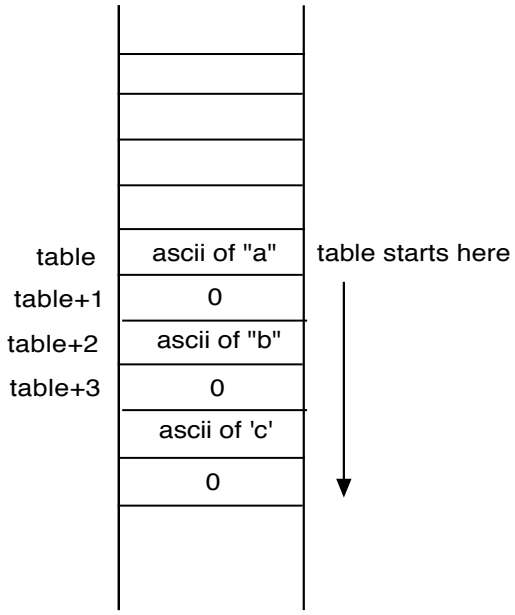

Figure 1. An outline of how table will be stored in the memory.
Each address in table is a byte address

(2) As your program reads a character, it will call a subroutine lookup that looks up the table, and updates the count of that character.

(3) The remaining piece of your code will find out which character is the most frequent (reduces to maxima finding).