EE 352 Lab 1

Nazarian Spring 2010

Name: _____

Due: Fri. Jan 29th at 1:00 p.m. (online) **Score:** _____

Note: Attach all work to receive full credit

The purpose of this assignment is to write an assembly language program using various control structures (loops, conditionals, etc.). You will also use appropriate syscalls to input & output information from/to the user.

Introduction & Background Notes

1) Syscalls – There are several syscalls that you will need to use to complete your program. Before executing the 'syscall' instruction you must place an argument in $v0 and often $a0 or $a1. The value in $v0 indicates which syscall should be executed with additional arguments in $a0 and $a1.

Syscall	$v0	Additional arguments	Notes
'read string'	8	$a0 = address of input buffer $a1 = maximum # of characters to read	Leaves the LF character at the end of the string
'print string'	4	$a0 = address of null terminated ASCII string	
'print character'	11	$a0 = ASCII char. to print	

Program Requirements

1) Your program shall allow a user to enter up to 20 single digit numbers on the same line separated by spaces. Your program shall then sort those numbers in ascending order (0 to 9). It shall then display the characters to the user in sorted order. Finally, it shall printout the median and mode (the digit with the most occurrences within the sequence).

2) The requirements for your program are as follows:
   a) Output a prompt for the user to enter a sequence of up to 20 single-digit numbers separated by a single space character. (You may assume that the input will always be entered in correct format. You are not responsible for handling deviations from this format.) Note: There may be less than 20 numbers though.

   b) Use the "Read string" syscall provided by MARS to accept the input and store it in the memory buffer, BUF1, which is allocates 40 bytes. Note: The "read string" syscall will store the ASCII line-feed (LF=0x0a) from the user hitting 'Enter' to complete the string. You can ignore this or use it if desired, just be careful not to sort it or the spaces between the digits.

   c) You shall then echo back the string to the user after which you may strip out the spaces to perform the sort, or perform the sort in place.

d) Though there are several ways to sort numbers, implement the sort using a simple bubble sort algorithm similar to that described below:

```
for (i=0; i < MAX; i++){
  for (j=0; j < MAX-i; j++){
    if (BUF1[j] > BUF1[j+1] ){
      temp = BUF1[j]
      BUF1[j] = BUF1[j+1]
      BUF1[j+1] = temp;
    }
  }
}
```

e) Next, printout a message to the user, "Sorted numbers: " followed by the sorted sequence with spaces reinserted.

f) Finally, perform some analysis on the sorted numbers and output a message indicating the median & mode.

   i) Note 1: if there is an even number of digits, pick the n/2 – 1 indexed item

   ii) Note 2: if there is a tie for the mode (most occurrences) pick the larger digit to output.

Programming and Submission Guidelines

1) A program skeleton is provided. Use it as a baseline.

2) You should be careful about a few things. First, the numbers you will be receiving as input will be *ASCII representations* of 0-9. You should think carefully whether it is beneficial to convert them to their actual 0-9 representations or leave them as their ASCII representations. Second, be careful to use appropriately sized load and store instructions. ASCII characters are each a byte.

3) You may use any pseudo-instructions you desire. "li – load immediate", "la – load address", and branch pseudo-instructions will be helpful.

4) Submit your source file by going to the Lab1 assignment on Blackboard (Assignments..Labs..Lab1) and then attach and submit your ee352_lab1.s file. Note: You must "Submit" and not just "Save" when you perform the submission.

See back for sample program execution and program skeleton.

Sample Program Execution:

```
Enter up to 20 digits separated by spaces: 5 9 3 7 1 0 4 8 8 1
Original list: 5 9 3 7 1 0 4 8 8 1
Sorted numbers: 0 1 1 3 4 5 7 8 8 9
Median: 4
Mode: 8
```

```
# ee352_lab1.s
# EE 352, Spring 2008
# Name:  _____________________
#

                .data
buf1:   .space  41   # receive original input in this buffer
buf2:   .space  41   # can use this buffer for any other purpose (strip spaces, etc.)

# the following are constant strings that you can use for your prompts and messages
msgin:  .asciiz "Enter up to 20 digits separated by spaces: "
msg1:   .asciiz "Original list: "
msg2:   .asciiz "Sorted numbers: "
msg3:   .asciiz "Median: "
msg4:   .asciiz "Mode: "

# print this string for a newline character
Nline:  .asciiz "\n"


            .text
main:




# exit
            li     $v0,10
            syscall
```