CS224 - Fall 2021- Lab #2 (Version 1: Oct. 13, 12:20 pm)

MIPS Assembly Language Programs Using Subprograms

Dates: Section 1, Wednesday, 20 October, 13:30-17:20
Section 2, Thursday, 21 October, 13:30-17:20
Section 3, Wednesday, 20 October, 8:30-12:20
Section 4, Friday, 22 October, 13:30-17:20
Lab Location: EA-Z04

TAs
Section 1: Kenan Çağrı Hırlak, Pouya Ghahramanian
Section 2: Kenan Çağrı Hırlak, Soheil Abadifard
Section 3: Ege Berkay Gülcan, Sepehr Bakhshi
Section 4: Alper Şahıstan, Soheil Abadifard (Tutor: Burak Öçalan)

TA/Tutor name (x No of labs) email address
Alper Şahıstan (x1): alper.sahistan@bilkent.edu.tr
Ege Berkay Gülcan (x1): berkay.gulcan@bilkent.edu.tr
Kenan Çağrı Hırlak (x2): cagri.hirlak@bilkent.edu.tr
Pouya Ghahramanian (x1): ghahramanian@bilkent.edu.tr
Sepehr Bakhshi (x1): sepehr.bakhshi@bilkent.edu.tr
Tutor Burak Öçalan (x1): burak.ocalan@ug.bilkent.edu.tr

Purpose: Understanding dynamic array allocation, passing arguments to and receiving results from subprograms, use of stack in subprograms. Understanding logical bit operations.
Note: Use the $s registers for the implementation of all of the subprograms, this is a requirement. In your implementations use the traditions of professional MIPS programmers.

Summary
Preliminary Work: 40 points
1. convertOctalToDec (20 points)
2. switchNibbles (20 points)

Lab Work: 60 points
1. main (10 points)
2. monitor (10 points)
3. bubleSort (30 points)
4. medianMax (10 points)

Important Notes for All Labs
1. Try to complete the lab part at home before coming to the lab. Make sure that you show your work to your TA and answer his questions to show that you know what you are doing before uploading your lab work and follow the instructions of your TAs. In all labs if you are not told you may assume

that inputs are corrects. For all works when needed please provide a simple user interface for inputs and outputs.

2. You are obliged to read this document word by word and are responsible for the mistakes you make by not following the rules. Your programs should be reasonably documented (purpose etc.) and must have a neat presentation in terms of variable names, subprogram names. indentation, comments, blank lines etc.
3. **If we suspect that there is cheating we will send the work with the names of the students to the university disciplinary committee.**

DUE DATE OF PRELIMINARY WORK: SAME FOR ALL SECTIONS

No late submission will be accepted.

a. Please upload your programs of preliminary work to Moodle by 9:30 am on Wednesday Oct 20 for similarity testing by MOSS.

b. For preliminary work upload a file that includes all programs in proper order with the filename **StudentID_FirstName_LastName_SecNo_PRELIM_LabNo.txt** Only a NOTEPAD FILE (txt file) is accepted. Any other form of submission receives 0 (zero).

c. Note that the Moodle submission closes sharp at 9:30 am and no late submissions will be accepted. You can make resubmissions before the system closes, so do not wait for the last moment. Submit your work earlier and change your submitted work if necessary. Note that only the last submission will be graded.

d. Do not send your work by email attachment, they will not be processed. They have to be in the Moodle system to be processed.

DUE DATE OF LAB WORK): (different for each section) YOUR LAB DAY

a. You have to demonstrate your lab work to your TA for grading. Do this by **12:00** in the morning lab and by **17:00** in the afternoon lab. Your TAs may give further instructions on this and they may make changes. If you wait idly and show your work last minute, your work may not be graded. Make sure that you follow your TA's instructions.

b. At the conclusion of the demo for getting your grade, you will **upload your Lab Work** to the Moodle Assignment, for similarity testing by MOSS. See lab part submission details below.

c. Aim to finish all of your lab work before coming to the lab, but make sure that you upload your work after making sure that your work is analyzed by your TA and/or you are given the permission by your TA to upload.

Part 1. Preliminary Work (40 points)

1. **convertOctalToDec (20 points)**: Write a subprogram that receives the beginning address of a null terminated (asciiz) string that contains a octal number; for example, like "17", and returns its decimal (for 17_8 it returns 15_{10}) equivalent. If the number stored in the input string is not a proper octal number it returns a negative value. In your case you will get the string input from the user.

For this assignment understand and use lbu: load byte unsigned instruction.
lbu $t1,0($t2) : Sets $t1 to zero-extended 8-bit (byte) value from the memory address indicated by the second operand.

For example

```
la $t2, myString
lbu $t1,0($t2)
....
myString .asciiz "0123"  # Defined in data segment
```

makes the contents of $t1 equal to 0x 00 00 00 30 since ASCII representation of 0 (the first character of "0123") is 0x30.

- Remember ASCII representation of characters (see 2nd page of Greencard).
- Use syscall 8 to read character strings. Please read the explanation provided in MARS help, experiment with it and learn it by yourself.
- Your program should be interactive must work as long as input is correct and must stop when the user enters an illegal number. The top (main program) must provide the user interface.

2. **switchNibbles (20 points)**: Write a subprogram, called switchNibbles that switches nibbles of each byte of its argument (stored in $a0) and returns switched version as its result. For example, if $a0 contains 0xa1b2c3d4, the leftmost byte contains 0xa1 and its nibbles are "a" and "1"; for the next byte its nibbles are "b" and "2". When the subprogram is done it returns 0x1a2b3c4d.

Your program should be interactive it will accept a decimal number and display input and output of the subprogram in hexadecimal. The main program should continue and stop when the user wants to quit. The main must provide the user interface.

Part 2. Lab Work (60 points)

In this part when needed please use the stack. Make sure that in the subprograms you use $s registers. The use of $t registers is not allowed in sub programs.

1. main (10 points) Gets array size from the user, performs dynamic storage allocation for the array, invokes monitor, after returning from monitor displays sorted array and median and maximum values returned by the monitor.

2. **monitor (10 pts.)**: Provides a user interface to use the following subprograms in an interactive manner. the subprogram monitor:
Receives array beginning address and array size from main,
Initializes array contents by user interaction,
Calls bubbleSort to sort the array,
Calls medianMax and it returns the median and max values to main. The main also receives the sorted array since it knows the beginning address of the array.

3. **bubbleSort (30 points)**: Sorts an integer array in ascending order using the bubble sort algorithm. The subprogram receives the beginning address of the array in $a0, and the array size in $a1. The array size can be 1 or more.

4. **medianMax (10 points)**: Returns the median and maximum values of a sorted array. It receives array address and array size. Note that array contains 1 or more elements.

Part 3. Submit Lab Work for MOSS Similarity Testing

1. Submit your Lab Work MIPS codes for similarity testing to Moodle.
2. You will upload one file. Use filename **StudentID_FirstName_LastName_SecNo_LAB_LabNo.txt**
3. Only a NOTEPAD FILE (txt file) is accepted. No txt file upload means you get 0 from the lab. Please note that we have several students and efficiency is important.
4. *Even if you didn't finish, or didn't get the MIPS codes working, you must submit your code to the Moodle Assignment for similarity checking.*
5. Your codes will be compared against all the other codes in the class, by the MOSS program, to determine how similar it is (as an indication of plagiarism). So be sure that the code you submit is code that you actually wrote yourself!

Part 4 . Cleanup

1. After saving any files that you might want to have in the future to your own storage device, erase all the files you created from the computer in the lab.
2. When applicable put back all the hardware, boards, wires, tools, etc where they came from.
3. Clean up your lab desk, to leave it completely clean and ready for the next group who will come.

LAB POLICIES

1. You can do the lab only in your section. Missing your section time and doing in another day is not allowed.
2. Students will earn their own individual lab grade. The questions asked by the TA will have an effect on your individual lab score.
3. Lab score will be reduced to 0 if the code is not submitted for similarity testing, or if it is plagiarized. MOSS-testing will be done, to determine similarity rates. Trivial changes to code will not hide plagiarism from MOSS—the algorithm is quite sophisticated and powerful. Please also note that obviously you should not use any program available on the web, or in a book, etc. since MOSS will find it. The use of the ideas we discussed in the classroom is not a problem.
4. You must be in lab, working on the lab, from the time lab starts until your work is finished and you leave.
5. No cell phone usage during lab.
6. Internet usage is permitted only to lab-related technical sites.