CS224 - Spring 2022 - Lab #2 (Version 1: February 24, 9:00)

Subprograms, Use of Stack, Dynamic Storage Allocation, Bit Processing in MIPS Assembly Language Programs

Dates:

Section 1: Mon, 7 Mar, 8:30-12:20 in EA-Z04
Section 2: Wed, 9 Mar, 13:30-17:20 in EA-Z04
Section 3: Tue, 8 Mar, 13:30-17:20 in EA-Z04
Section 4: Fri, 4 Mar, Fri 08:30-12:20 in EA-Z04
Section 5: Wed, 9 Mar, 8:30-12:20 in EA-Z04
Section 6: Fri, 4 Mar, 13:30-17:20 in EA-Z04

Lab3 Days:

Due to the lab rotation policy the first Lab3 session will be with Section 3 (March 15, Tuesday). After that Section 2, 4, 5, 6 will do their labs in the same week on their respective lab days. Section 1 will do its lab in the following week on March 21. See the course syllabus for the lab days.

TAs;Tutor:

Section 1: Pouya Ghahramanian, Pouria Hasani; Fazıl Keskin
Section 2: Alper Şahıstan, Hüseyin Eren Çalık; Burak Öçalan
Section 3: Kemal Büyükkaya, Kenan Çağrı Hırlak
Section 4: Pouria Hasani, Sepehr Bakhshi
Section 5: Kenan Çağrı Hırlak, Soheil Abadifard; Alper Mumcular
Section 6: Alper Şahıstan, Soheil Abadifard

TA name (x No of labs): email address

Alper Şahıstan (x2): alper.sahistan@bilkent.edu.tr
Hüseyin Eren Çalık (x1): eren.calik@bilkent.edu.tr
Kemal Büyükkaya (x1): kemal.buyukkaya@bilkent.edu.tr
Kenan Çağrı Hırlak (x2): cagri.hirlak@bilkent.edu.tr
Pouria Hasani (x2): pouria.hasani@bilkent.edu.tr
Pouya Ghahramanian (x1): ghahramanian@bilkent.edu.tr
Sepehr Bakhshi (x1): sepehr.bakhshi@bilkent.edu.tr
Soheil Abadifard (x2): soheil.abadifard@bilkent.edu.tr

Tutor name (x No of labs): email address

Alper Mumcular (x1): alper.mumcular@ug.bilkent.edu.tr
Burak Öçalan (x1): burak.ocalan@ug.bilkent.edu.tr (Tutoring in the lab Thursdays, 6:15 - 8:00 pm)
Fazıl Keskin (x1): fazil.keskin@ug.bilkent.edu.tr

Purpose: 1. Understanding preliminary principles of using stack for saving $s registers, **2**. Passing arguments to and receiving results from subprograms, **3**. Dynamic storage allocation, **4**. Array processing, **5**. Bit processing. See the following pointers for your implementation.

In the implementation of subprograms you are not allowed to use $t registers: use $s registers to get used to the MIPS software development principles. When you enter to a subprogram save $s registers you use in the subprogram to the stack etc. Remember and use the rules of passing parameters to subprograms and results to caller.

If you use $t registers in subprograms (once or more) in Part 1 (Preliminary Work) you will lose 10 points. If you use $t registers in subprograms (once or more) in Part 2 (Lab Work) you will lose another 10 points.

Preliminary Work: 45 points

1. Circular shifts (20 points)
2. Array processing (... check if palindrome) (25 points)

Lab Work: 55 points

1. Bubble sort (35 points)
2. Process sorted array (20 points)

Important Notes for All Labs About Attendance, Performing and Presenting the Work

1. You are obliged to read this document word by word and are responsible for the mistakes you make by not following the rules.
2. Not attending to the lab means 0 out of 100 for that lab. If you attend the lab but do not submit the preliminary part you will lose only the points for the preliminary part.
3. 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.
4. In all labs if you are not told you may assume that inputs are correct.
5. In all labs when needed you have to provide a simple user interface for inputs and outputs.
6. Presentation of your work

   You have to provide a neat presentation prepared in txt form. Your programs must be easy to understand and well structured.

   Provide following six lines at the top of your submission for preliminary and lab work (make sure that you include the course no. CS224, important for ABET documentation).

   CS224
   Lab No.
   Section No.
   Your Full Name
   Bilkent ID
   Date

   Please also make sure that you work is identifiable: In terms of which program corresponds to which part of the lab.
7. **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 PRELIMINARY WORK: SAME FOR ALL SECTIONS

No late submission will be accepted. Please do not try to break this rule and any other rule we set.

1. Please upload your programs of preliminary work to Moodle by 9:30 am on Friday March 4, 2022.
2. Please note that the submission closes sharp at 9:30 am and no late submissions will be accepted. You can make resubmissions 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.
3. Please familiarize yourself with the Moodle course interface, find the submission entry early, and avoid sending an email like "I cannot see the submission facility." (As of now it is not yet opened.)
4. Do not send your work by email attachment they will not be processed. They have to be in the Moodle system to be processed.
5. Use filename **StudentID_FirstName_LastName_SecNo_PRELIM_LabNo.txt** Only a NOTEPAD FILE (txt file) is accepted. Any other form of submission receives 0 (zero).

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

1. 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. If you wait idly and show your work last minute, your work may not be graded.
2. 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 below for the details of lab work submission.
3. Try to finish all of your lab work before coming to the lab, but make sure that you upload your work after making sure that it is analyzed by your TA and/or you are given the permission by your TA to upload.

Part 1. Preliminary Work (45 points)

1. **Circular Shifts (20 points)**: In this part implement two different bit circulations for registers

Shift Left Circular (SLC): Implement a method (subprogram) called shiftLeftCircular that shifts the contents of a register and works as demonstrated below:

# input 8 hex digits (shift left amount is in no. of bits) ==> output 8 hex digits

0X**AA** 00 00 BB (SLC 8) ==> 0X00 00 BB **AA**

As shown above the bits falling down from left comes to right in the order they drop. The no. of bits can have any positive value (make sure that you provide an efficient implementation).

Shift Right Circular (SRC): Implement a method (subprogram) called shiftRightCircular. It works like SLC but this time the bits falling down from right comes to left in the order they drop.

0XAA 00 00 **BB** (SRC 8) ==> 0X**BB** AA 00 00

Write two separate subprograms for these circular shift operations. Provide the necessary interface for testing your code in the main (top level) program. Ask the user to enter the decimal integer number to be shifted and a number that indicates the amount of shift. Display the number to be shifted, the shift amount and direction, and the shifted number in hexadecimal on the console.

You must pass the number to be shifted and the shift amount in $a0, and $a1 respectively and return the result in $v0 (as required by the rules of MIPS software development).

How to display an integer in hexadecimal: See Mars help menu on syscalls.

Make sure that you have an efficient implementation.

2. **Array Processing (25 points)** : The main program invokes two subprograms: **createArray** and **arrayOperations**.

createArray: Asks the user the array size in number of words, performs dynamic storage allocation using syscall (9) and initializes the array elements with interaction on the console. It returns to the main program the array beginning address and size respectively.

arrayOperations: Receives array address and array size from the main program respectively in $a0 in $a1 and invokes **min, max, sum, palindrome** subprograms to perform the following actions by the use of seperat subprograms with proper simple user interaction.

- **min**: returns the minimum value stored in the array.
- **max**: returns the maximum value stored in the array..
- **sum**: returns the summation of array elements.
- **palindrome**: Checks if array contents defines a palindrome. An array that contains for example 1, 4, 4, 1 is a palindrome. An array with zero or one element is a palindrome by definition. The method returns 1 in $v0 if it is a palindrome, otherwise it returns 0.

Part 2. Lab Work (55 points)

Use **createArray** in this part as needed.

1. **bubbleSort (35 points)**: Main program invokes a subprogram (**bubbleSort**) that sorts the array in ascending order using the bubble sort algorithm. The array size can be 0 or more. It invokes the subprogram **processSortedArray** after sorting.

2. **processSortedArray (20 points):** This subprogram displays the sorted array by providing array index position, array element value, and summation of the digits of the element (using a method called **sumDigits**, implement this subprogram). Therefore, for each array element it generates one line with four items. For example, if the array position 5 contains 25 it displays

```
5	25	7
```

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. <u>Only a NOTEPAD FILE (txt file) is accepted.</u> 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. The questions asked by the TA will have an effect on your lab score.
3. Lab score <u>will be reduced to 0</u> 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.