**CSE 338 – Computer Organization** **Spring 2020**
**Prof. Haluk Topcuoğlu**

# PROGRAMMING PROJECT 1
# Due: 01 / 05 / 2020 – 23:30 ( No late submission)

In this project, you are required to implement some procedures in MIPS assembly language. You will use SPIM simulator [1] to develop and test your code. There will be three questions in the project which are unrelated.

**QUESTION 1.** *(10 points)* In this program you are required to implement a number series for square root of 2 approximation based on given inputs. Following algorithm used to approximate square root of 2 by the ancient mathematicians.

| Iteration | *a* | *b* | $a^2-2b^2$ | *a/b* |
|---|---|---|---|---|
| 1 | 1 | 1 | -1 | 1 |
| 2 | 3 | 2 | +1 | 1.500 |
| 3 | 7 | 5 | -1 | 1.400 |
| 4 | 17 | 12 | +1 | 1.417 |
| 5 | 41 | 29 | -1 | 1.414 |

$a_2 = a_1+2b_1$
$b_2 = a_1+b_1$

Your program should take a iteration count and print the list of nominator and denominator for approximation. Next iteration *a* value is calculated by *a+2b* and next iteration *b* is calculated by *a+b*. All the numbers in the series should fit in a word!

An example run:
Enter the number of iteration for the series: 5

a: 1 3 7 17 41
b: 1 2 5 12 29

---

**QUESTION 2.** *(22 points)* In this question, you are required to implement matrix multiplication in MIPS. Given two list of linear arrays and matrices dimensions as inputs, output multiplication matrix will be printed. You can assume that matrix input sizes will always be valid. For example if first matrix in length of ***mxn***, second matrix will be in length of ***nxk***. User will only enter dimensions ***m*** and ***n*** where you can derive ***k*** from length of second matrix**.** You will assume matrices are given in row and then column order. For example if matrix elements are:
3 5 4 6 2 8
and dimensions are 3x2, matrix is:

3 5
4 6
2 8

or if the dimensions are 2x3, matrix is:
3 5 4
6 2 8

An example run:
Enter the first matrix: 3 7 8 12 5 6 7 2 4 3 2 5
Enter the second matrix: 2 5 6 3 7 8 9 1
Enter the first dimension of first matrix: 3
Enter the second dimension of first matrix: 4

Multiplication matrix:
212 112
113 101
85 50

---

**QUESTION 3.** *(15 points)* Write a MIPS program that performs case insensitive palindrome checker. Given any string with the maximum length of 100 characters, you program will decide if the given string is palindrome or not. Palindrome strings are strings that are equivalent to their reverse. If the lower case letter is matched with the upper case letter in the reverse, it is a valid palindrome.

Examples palindrome strings:
ABBA
Step on no pets
abGcdEEedcgba

Example Runs:

Enter an input string: abGcdEEedcgba

abgcdeeedcgba is palindrome.

Enter an input string: ABBnMBBa

abbnmbba is not palindrome.

---

**MENU** *(8 points):* Your program should support a *Menu* including all questions above. A sample execution scenario given below:

```
Welcome to our MIPS project!
Main Menu:
1. Square Root Approximate
2. Matrix Multiplication
3. Palindrome
4. Exit
Please select an option: 1
```

These options must be printed inside a loop until "Exit" option is selected.
When the user select option 1, you should print the followings:

```
Enter the number of iteration for the series: 5
a: 1 3 7 17 41
b: 1 2 5 12 29

Main Menu:
1. Square Root Approximate
2. Matrix Multiplication
3. Palindrome
4. Exit
Please select an option: 2
Enter the first matrix: 3 7 8 12 5 6 7 2 4 3 2 5
Enter the second matrix: 2 5 6 3 7 8 9 1
Enter the first dimension of first matrix: 3
Enter the second dimension of first matrix: 4

Multiplication matrix:
212  112
113  101
85   50

Main Menu:
1. Square Root Approximate
2. Matrix Multiplication
3. Palindrome
4. Exit
Please select an option: 3
Enter an input string: abGcdEEedcgba

abgcdeeedcgba is palindrome.

Main Menu:
```

```
1. Square Root Approximate
2. Matrix Multiplication
3. Palindrome
4. Exit
Please select an option: 4
Program ends. Bye :)
```

## Assumptions and Requirements

- The arguments to the procedures are stored in $a registers; i.e., the first one is in $a0, the second one is in $a1, and so on.
- Only valid arguments are passed into the procedures. Therefore, you do not need to check the arguments for their validity.
- When you invoke a procedure, the values of all $a registers should be preserved. Their values should be same at the end of the procedure call as they were at the time of call.
- You have to use QtSpim simulator in your implementation. Any other simulator is not allowed.
- You are required to submit a minimum 2-page report (***5 points***) explaining implementation details of your project. Your report will have three parts (one for each question) and it will also include screenshot of your sample runs, as well.
- You should submit a fully commented source code that includes details of your implementation. Note that the name of the file should include surnames of the group members. (ex: surname1_surname2.s)
- Zip your fully commented source code file and the project report into a single file and submit the zip file via Canvas.

## General Policies for the Project

- *You have to work in groups of 2*. You will select your partner and the partner will not be changed throughout the semester. *It is not acceptable of a partner team to work with other teams*.
- *A portion of your project grade will be set with a Project Quiz*. Note that if you do not submit the project, you will not attend the Project Quiz
- Copying (partially or full) solutions from other students is a form of cheating. Copying (partially or full) solutions from Web including Github (and similar sites) is another form of cheating. It is NOT acceptable to copy (or start your) solutions from Web. **In case of any forms of cheating or copying among the groups, the penalties will be severe. Both Giver and Receiver are equally culpable and suffer equal penalties!!!**
- No late submission will be accepted!

## References

[1] http://spimsimulator.sourceforge.net/