CSE 3038 – Computer Organization **Spring 2023**
Prof. Haluk Topcuoğlu

PROGRAMMING PROJECT 1
Due: 17 / 04 / 2022 – 23:30 (No late submission)

In this project, you are required to implement a set of functions in MIPS assembly language. You will use a MIPS simulator (QTSPIM or MARS) to develop and test your code. There will be four questions in the project which are unrelated.

Question 1. *(12 points)*
Write a MIPS procedure that will display the longest palindrome which can be generated from a from a given string. Although the given input may contain letters (lowercase or uppercase), punctuation marks, and/or digits, you should consider only the letters when creating the palindrome. The procedure is not case sensitive; specifically, "aA" is considered as a palindrome in your procedure. Your MIPS code must print the longest palindrome and its length. Note that there may be more than one correct answer which have the same length.

Example Run 1:
Input: "aaabbbbfffffffcd"
Output: The longest palindrome is abbfffffffbba, and its length is 12.

Example Run 2:
Input: "12aBbdddeee!!"
Output: The longest palindrome is bdeedb, and its length is 6.

Explanation for the second example:
We only consider letters; therefore, there are 1 a, 2 b (it is not case sensitive), 3 d and 3 e.
Then an example for the longest palindrome is: " bdeedb", where its length is 6.
Another correct answer is: edbbde.

Question 2. *(12 points)*
Write a MIPS procedure that will reverse vowels in each string. The procedure prints the given string by reordering vowels (a, e, i, o, u) in the string. Assume that the first vowel in the string is "a" and the last one is "e"; then, when your procedure prints the string, the first vowel should be "e" and the last one should be "a"; where it will reverse all other vowels in the string accordingly.
Note that the input string may contain alphanumeric characters. The program is case sensitive.

Example Run 1:
Input: Hello World!!!
Output: Hollo Werld!!!

Example Run 2:
Input: This is the first CSE3038 project.
Output: Thes os thE first CSE3038 prijict.

Question 3. *(12 points)*
Any number that is greater than 1 can be expressed as the multiplication of the prime factors. A positive integer is called square-free if it is a product of distinct primes. Since the prime numbers can be divided by only themselves, they are also considered square-free numbers. In this question, your MIPS procedure will decide whether the given integer is a square-free number or not. For example, the prime factors of 6 are 2 and 3; so 6 is a square-free number. The prime factor of 8 is 2 (2*2*2), so 8 is not a square-free number. Your MIPS code will determine that the input integer is square free number or not; and if it is a square-free number it will print its distinct prime factors.

Example Run 1:
Enter an integer number: 8
Output: 8 is not a square-free number.

Example Run 2:
Enter an integer number: 12
Output: 12 is not a square-free number.

Example Run 3: Enter an integer number: 15
Output: 15 is a square-free number and has two distinct prime factors: 3 5

Question 4. *(20 points)*
In an *nxm* matrix, the *lucky number* in the matrix (let us call it y) is the number where all elements in the same row are greater than y, and all elements in the same column are less than y. In this question you are required to write a MIPS procedure for finding the lucky number of an *nxm* matrix. The MIPS code asks the integer array from the user and then prints the lucky number of the matrix. The elements of the matrix must be unique; and it is your responsibility that make sure the integers given by user are distinct.
Example Run 1:
Input: Enter the number of rows: *3*
Enter the number of columns: 4
Enter the elements of the matrix: 1 5 3 7 8 1 4 9 9 6 4 12
Output: The matrix should have only unique values.

Example Run 2:
Input: Enter the number of rows: *3*

Enter the number of columns: 4
Enter the elements of the matrix: 1 5 3 7 8 2 4 10 9 6 4 12
Output: The lucky number is 6.
Explanation for the second example:

6 is smaller than {9, 11, 12} and it is greater than {5, 2}, so for the following matrix the lucky number is 6.

1	5	3	7
8	2	4	10
9	6	11	12

MENU *(9 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. Find Palindrome
2. Reverse Vowels
3. Find Distinct Prime
4. Lucky Number
5. 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:

```
Input: “aaabbcd”
Output: The longest palindrome is abba, and its length is 4.

Main Menu:
1. Find Palindrome
2. Reverse Vowels
3. Find Distinct Prime
4. Lucky Number
5. Exit
Please select an option: 2
Input: Hello World!!!
Output: Hollo Werld!!!

Main Menu:
1. Find Palindrome
2. Reverse Vowels
3. Find Distinct Prime
4. Lucky Number
5. Exit
```

```
Please select an option:  3
Enter an integer number: 7
Output: 7 has one distinct prime factor: 7

Main Menu:
1. Find Palindrome
2. Reverse Vowels
3. Find Distinct Prime
4. Lucky Number
5. Exit
Please select an option:  4
Input: Enter the row number: 3
Enter the column number: 4
Enter the elements of the matrix: 1 5 3 7 8 2 4 10 9 6 4 12
Output: The lucky number is 6.

Main Menu:
1. Find Palindrome
2. Reverse Vowels
3. Find Distinct Prime
4. Lucky Number
5. Exit
Please select an option: 5
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 or MARS 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 four 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 such as *surname1_surname2_surname3_surname4.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 3 or 4*. You will select your partners and partners 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 (a Demo session will be also planned).* 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!