Ho Chi Minh City University of Technology

Faculty of Computer Science and Engineering
Course: Computer Architecture Lab (CO2008)

Lab 2

For the Honor Program

Ho Chi Minh City, October 23rd 2023

Contents

1 Introduction

- The main purpose of this session is to get familiar with conditional branch and unconditional jump instructions and work with recursion.
- Students are also expected to be able to call procedures in this lab.
- Students must submit their answers to the BKeL system no later than the last period of the lab section. Then, the instructor will evaluate all students' work during the lab section's final period. Please note that we will randomly choose ~50% of the questions to mark.

2 Exercises

2.1 Exercise 1

Write a MIPS program that does the following steps:

1. Declare a string with only alphabet letters in the memory and count the number of each characters that appears in the string.
2. Print the characters and their number of appearances by descending order (if the number of appearance is the same between some numbers, print the one that has the larger ASCII code first).

For example, if the input string is "**abdeefggff**" the output should be **f: 3; ; g: 2; e: 2; d: 1; b: 1; a: 1;**.

2.2 Exercise 2

Write a MIPS program that requests two positive integers called a and b from the user. The program then prints their greatest common divisor (GCD) and least common multiple (LCM).

Below is the pseudocode of the recursive functions:

```
    int GCD(int a, int b) {
        if (b == 0) return a;
        return GCD(b, a % b);
    }
    int LCD(int a, int b) {
        return (a * b) / GCD(a, b);
    }
```

For example, if the user input **a = 12, b = 34**, the output should be **GCD = 2, LCM = 204**.

2.3 Exercise 3

Write a MIPS program with the following requirements:

1. Declare an array that can store 8 data elements.
2. Request integers from the user and store them into the array.
3. Check whether each element in the array is divisible by 3.
4. If an element is divisible by 3, divide it by 3 and add the result with 1.
5. If an element is not divisible by 3, change it to the number divisible by 3 that is closest to it, i.e, if the number is 32, the result will be 33.
6. Print the final array to the terminal.

2.4 Exercise 4

Using the same result as 2.3, write a MIPS program to find the second smallest element in a 15-elements array. If the array has more than one second smallest element, find all their indexes. Print the value and all of its indexes.
For example, if the array is **1, 2, 7, 2, 3, 7, 4, 5, 6, 7, 7, 8, 8, 8, 7** the output should be **Second smallest value is 2, found in index 1, 3**.

2.5 Exercise 5

Write a MIPS program to check if the elements of a 10-elements array are unique (appears only once in the array). If there are duplicated values in the array, print those values.
For example, if the array is **1, 2, 3, 3, 3, 1, 7 ,8, 9, 10** then the output should be **Unique values: 2, 7, 8, 9, 10. Duplicated value: 1, repeated 2 times; 3, repeated 3 times.**