Ho Chi Minh City University of Technology

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

# Lab 3

## for Honor program

Ho Chi Minh City, October $23^{rd}$ 2023

# Contents

1 Introduction 2

2 Exercises 2

2.1 Exercise 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2

2.2 Exercise 2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2

2.3 Exercise 3 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2

2.4 Exercise 4 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

# 1 Introduction

- The main purpose of this session is to get familiar with floating-point instructions and read/write file.
- 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 calculates and print either the volume or total surface area of a rectangular box, cube, cylinder, sphere, or rectangular pyramid. The user is able to choose which shape and the related parameters to calculate. Note that the parameters can be floating-point numbers.

## 2.2 Exercise 2

In this exercise, students are required to write a recursive program although the problem can be solved by iterations.
Write a MIPS program that calculates the sum of all 10 elements in a single precision floating point array with synthetic data. Bellow is pseudo code of the recursive version:

```
float sum(float *v, int k){
        if (k == 1) return v[0];
        return v[0] + sum(&v[1], k-1);
}
```

## 2.3 Exercise 3

In this exercise, students are required to write a recursive program although the problem can be solved by iterations.
Write a MIPS program that print all 10 elements in a single precision floating point array with synthetic data in the reverse order. Bellow is pseudo code of the recursive version:

```
void print(float *v, int k){
        if (k == 1) cout << v[0];
    else {
        print(&v[1], k-1);
        cout << v[0];
    }
}
```

## 2.4 Exercise 4

Taking the same requirements in Exercise 3, write a MIPS program that is able to find the minimum elements in an array. Bellow is pseudo code of the recursive version:

```
float min(float *v, int k){
        if (k == 1) return v[0];
        float temp = min(&v[1], k - 1);
        if (v[0] < temp) return v[0];
        else return temp;
}
```