Computer Organization
Lab 1 - MIPS Assembly

教授：蔡文錦

TAs：單宇晟 吳年茵 曾偉杰

Objectives

In this lab, we are going to learn how to write assembly code of MIPS architecture, and understand the difference between it and high-level languages such as C/C++. We will give an example to you.

1. Learn how to write **assembly code** and understand how it works
2. Learn how to run your assembly code by using **the MIPS simulator - SPIM**

Example: Factorial

- This is an example about computing factorial n!, where n is a given positive integer.
- The attached files factorial.c and factorial.s are the given example code.
- Please check the above files before moving on to the other tasks, for you to get familiar with assembly code.

```
$ gcc factorial.c -o factorial.c
$ ./factorial
Please input a number: 5
120
```

Tasks

For each task, you need to complete the C code **only TODO section**, and then translate it into MIPS assembly:

1. GCD & LCM
2. Exponentiation
3. Find max, min in array
4. Compute sum of digits

```c
void findMaxMin(int arr[], int size, int* max, int* min) {
    //TODO
}

int main() {
    // DO NOT modify this section
    int numbers[5];
```

each task account for **25%** of your score.

SPIM: A MIPS32 Simulator

- SPIM is a simulator for the MIPS, you can simply install SPIM with following command
  - macOS: ***brew install spim***
    - need to install homebrew first : https://brew.sh/
  - ubuntu: ***sudo apt-get install spim***
  - centOS: ***sudo yum install spim***
- You can using SPIM by using command line interface.
  - e.g. spim -file exp.s

```
$ spim -file ./factorial.s
Loaded: /opt/homebrew/Cellar/spim/9.1.24/share/exceptions.s
Please input a number: 5
The result of factorial(n) is 120
```

Task 1. GCD & LCM (25%)

Compute greatest common divisor (GCD) and least common multiple (LCM) of given two positive integers n1 and n2.

Input: Two numbers n1 and n2.

Output: GCD(n1,n2) LCM(n1,n2)

e.g.

```
$ ./gcd_lcm
Please enter the first number: 5
Please enter the second number: 10
5 10
```

Task 2. Exponentiation (recursive version) (25%)

Compute the exponentiation of given number n raised to the power of d **in the recursive way (Please refer to the given C code)** , where n and d are positive integers.

Input: Two numbers n and d.

Output: n^d

$$n^d = n * n^{d-1}$$

$$n^{d-1} = n * n^{d-2}$$

...

e.g.

```
$ ./exp
Enter base (positive integers): 5
Enter exponent (positive integers): 3
125
```

Task 3. Find max, min (25%)

Find the maximum value, minimum value of given five positive integers n1 to n5.

Input: Five numbers n1 to n5.

Output: max min

※Note that SPIM can only read one integer at a time. Therefore, you need to enter each number in turn

e.g.

```
$ ./max_min
Enter five positive integers: 1
2
3
4
100
100 1
```

Task 4. Sum of digits (25%)

Enter a positive integer and calculate the sum of all digits.

Input: A positive integer.

Output: The sum of all digits of the input integer

e.g. **5 + 5 + 8 = 18**

```
$ ./sum_of_digits
Enter an integer: 558
18
```

Grading Policy

- Each task have **5 hidden cases**, and you will get 5 points for each correct testcase, totally 25 points for each task.
- **Any assignment work by fraud will get a zero point！**
- **No late submission！**

Submission

- **Please attach student IDs as comments at the top of each file.**
- The files you should hand in include:
  - **gcd_lcm.s**
  - **exp.s**
  - **max_min.s**
  - **sum_of_digits.s**
- Compress the above file into one zip file, and name your zip file as **HW1_{studentID}.zip (e.g. HW1_123456789.zip)**
  - Make sure not to add an extra folder layer.
- **Deadline: 2025/03/19 23:55**
- **Wrong format will have 20% penalty !**

Reference

- https://s3-eu-west-1.amazonaws.com/downloads-mips/documents/MD00565-2B-MIPS32-QRC-01.01.pdf
- https://www.doc.ic.ac.uk/lab/secondyear/spim/node8.html