Computer Organization
Lab 1 - MIPS Assembly

教授:蔡文錦
助教:劉益先、鄭吉呈、黃芷柔、林彧頎

Objective

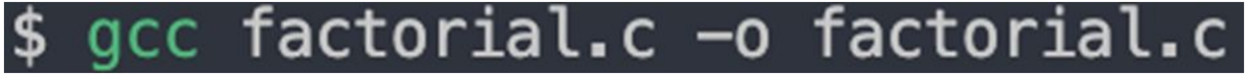
In this lab, we are going to learn how to write assembly code of MIPS architecture, and understand the difference between it and high-level languages such as C/C++. We will give an example to you.

1. Learn how to write assembly code and understand how it works.

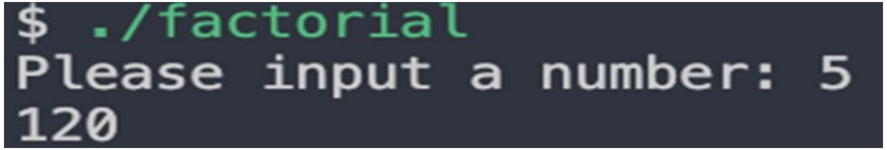
2. Learn how to run your assembly code by using the MIPS simulator - SPIM.

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
```

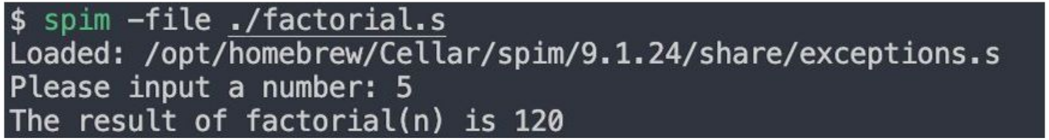
Tasks

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

1. Reverse Number (30%)
2. Sum of Fibonacci (40%)
3. Extended Euclidean algorithm (30%)

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



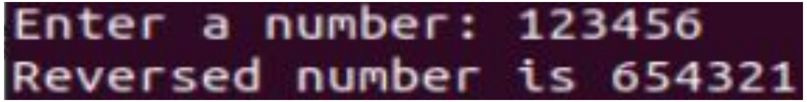
```
$ spim -file ./factorial.s
Loaded: /opt/homebrew/Cellar/spim/9.1.24/share/exceptions.s
Please input a number: 5
The result of factorial(n) is 120
```

Task 1. Reverse Number (30%)

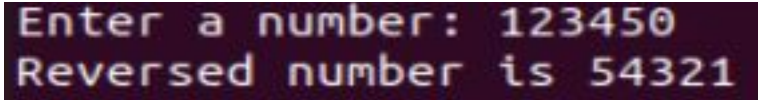
Enter an positive integer and reverse its digits.

Input: an positive interger (length of number < 10).

Output: the reversed result of the interger.



```
Enter a number: 123456
Reversed number is 654321
```



```
Enter a number: 123450
Reversed number is 54321
```

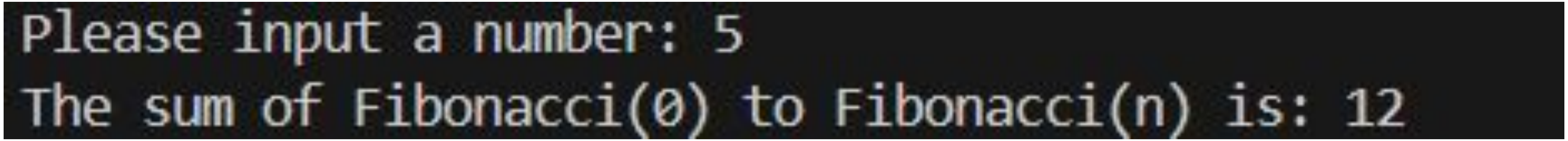
If the last digit of input number is 0,
the output will ignore the 0 in the
first digit.

Task 2. Sum of Fibonacci (40%)

Use recursion to find the sum of the first N Fibonacci numbers.

Input: An positive integer N(< 18).

Output: The sum of the first N Fibonacci numbers.



```
Please input a number: 5
The sum of Fibonacci(0) to Fibonacci(n) is: 12
```

reference:
https://en.wikipedia.org/wiki/Fibonacci_sequence

Task 3. Extended Euclidean algorithm (30%)

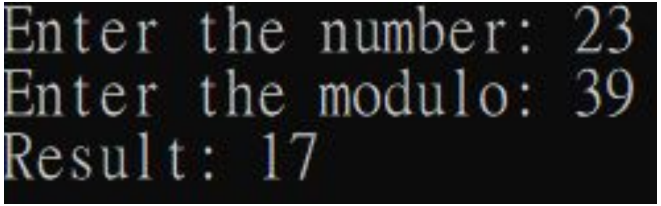
Find the modular inverse of the integer under the given modulus using Extended Euclidean algorithm.

Input: a positive integer and a modulus.

Output: the inverse of the interger (the result should be a positive integer less than the modulus).

reference:
https://zh.wikipedia.org/zh-tw/%E6%89%A9%E5%B1%95%E6%AC%A7%E5%87%A0%E9%87%8C%E5%BE%97%E7%AE%97%E6%B3%95

ex: $23\times17\equiv1(\text{mod } 39)$ ⇨



```
Enter the number: 23
Enter the modulo: 39
Result: 17
```

Grading Policy

- Each task have 5 hidden cases.
- 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:
    - reverse_number.s
    - sum_of_fibonacci.s
    - extended_euclidean.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.
- Wrong format will have 20% penalty！
- Deadline: 7/27 11:55pm