

Lab05 - Mixing with C

- Introduction to addressing mode:
 - Video Link: [Lab05: Mixing with C – YouTube](#)
 - HackMD Link: [Lab05: Mixing with C – HackMD](#)
- Lab requirements:
 - Basic (70%):
 - Description:

Given an 8-bit unsigned integer **a**, please implement:

ceil(sqrt(a))

The function returns an 8-bit unsigned integer.
 - Hint:

The function definition above can be referred to C standard library.

cplusplus.com/reference/cmath/
 - Example:

$\text{ceil}(\text{sqrt}(20)) = 5$
 $\text{ceil}(\text{sqrt}(50)) = 8$
 - Standard of grading:
 1. Integrate the required function into the C program. Implement the function's features in assembly, then call it from the main function.
 2. All test cases will fall between 1 and 255.
 3. Using function signature as follows:

```
extern unsigned char mysqrt(unsigned char a);
```
 4. Display the output in the WATCH window and provide a detailed explanation of your code logic.
 - Advanced(30%):
 - Description:

The “**main.c**” program calls a “**gcd**” function to find the greatest common

divisor (GCD) of two numbers. Please complete the “gcd” function using the PIC18F assembly language. The function accepts two 16-bit unsigned integers, 'a' and 'b', and returns their GCD as a 16-bit result.

■ Example:

a = 5, b = 10, gcd(5, 10) = 5

a = 1200, b = 180, gcd(1200, 180) = 60

■ Standard of grading:

1. Integrate the required function into the C program. Implement the function's features in assembly, then call it from the main function.
2. All test cases will fall between 1 and 65535.
3. Using the function signature as follows:

```
extern unsigned int gcd(unsigned int a, unsigned int b);
```

4. Display the output in the WATCH window and provide a detailed explanation of your code logic.

○ Bonus(20%):

■ Description:

The “main.c” program calls a “multi_signed” function to perform signed multiplication. Please complete the “multi_signed” function using the PIC18F assembly language. This function takes an 8-bit signed integer 'a' (ranging from -128 to 127) and a 4-bit signed integer 'b' (ranging from -8 to 7) as inputs and returns an unsigned integer 'res' representing the result of their multiplication. The output will be a 16-bit result. The result should be stored in an unsigned integer variable and then shown in the WATCH window. Please note that the signed data will be formatted in two's complement.

■ Example:

multiplicand = 127, multiplier = -6, multi_signed(127, -6) = 64774

■ Standard of grading:

1. Integrate the required function into the C program. Implement the function's features in assembly, then call it from the main function.
2. Using the function signature as follows:

```
extern unsigned int multi_signed(unsigned char a, unsigned char b);
```

3. Display the output in the WATCH window and provide a detailed explanation of your code logic.
4. Please avoid using the MULLW or MULWF instructions in your implementation.