Lab 01: Requirement Description

- **MPLAB X IDE的下載及安裝**
  - Link: https://youtu.be/uHa2KxXLEmM
- **Introduction to Instruction Set**
  - Link: https://youtu.be/eGXxTVUOqBo
- **Lab requirements:**
  - **基本題 (70%) ：**
    - **題目敘述：**用COMF與INCF指令將位址[0x000]之值轉成其二補數形式的負數，並將轉換後的值與位址[0x001]之值相比，若相等，則將data memory 位址 [0x002] 的值設為0xFF，否則設為0x01。
    - **舉例測資一：**
      - 在位址[0x000]存入0b01111100 (0x7C)，
      - 在位址[0x001]存入0b00000100 (0x04)，
      - 將位址[0x000]之值轉換後為0b10000100 (0x84)，
      - 與位址[0x001]之值相比後，發現不相等，
      - 因此位址[0x002]之值應設為0x01，當作答案。
    - **舉例測資二：**
      - 在位址[0x000]存入0b11111100 (0xFC)，
      - 在位址[0x001]存入0b00000100 (0x04)，
      - 將位址[0x000]之值轉換後為0b00000100 (0x04)，
      - 與位址[0x001]之值相比後，發現相等，
      - 因此位址[0x002]之值應設為0xFF，當作答案。
    - **評分標準：**
      1. 必須使用到COMF與INCF指令 。
      2. Demo測資有兩筆，與舉例測資的兩筆相同。
      3. 結果需存放在位址[0x002]中。
      4. 請在影片中開啟File Registers，並分別展示與講解兩筆測資的執行過程。

- **進階題 (30%) ：**
  - **題目敘述：**請設計一個迴圈，使用ANDWF及RRNCF計算數字0b00010111當中有多少bit值為1，並將結果存放於位址 [0x002]中。
  - **評分標準：**
    1. 必須使用到ANDWF及RRNCF指令。
    2. Demo測資與題目敘述中的測資相同 (0b00010111)。
    3. 結果需存放在位址[0x002]中，應為0x04。
    4. 必須使用迴圈，實作可自行發揮。
    5. 請在影片中開啟File Registers，並展示與講解執行過程。

- **加分題 (20%)：**
  - **題目敘述：** 請利用ANDWF、IORWF、COMF指令，實作出將位址[0x000]及[0x001]中的值進行XOR(exclusive or)運算的結果，將結果放入位址[0x002]中，但請注意不能直接用XORWF指令。
  - **評分標準：**
    1. 運算邏輯只能使用ANDWF、IORWF、COMF 指令(可以只選擇一種或兩種使用，不一定要三個都使用到)，其餘細節可以自行設計。
    2. 不能使用XORWF指令。
    3. Demo測資為 0b00001111與0b00110011，
    4. 結果需存放在位址[0x002]中。應為0b00111100。
    5. 請在影片中開啟File Registers，並展示與講解執行過程。
  - **提示：**

    題目的三個指令對應AND、OR、NOT邏輯運算

Lab 01: Requirement Description

- **MPLAB X IDE Download and Install**
  - Link: https://youtu.be/uHa2KxXLEmM
- **Introduction to Instruction Set**
  - Link: https://youtu.be/eGXxTVUOqBo
- **Lab requirements:**
  - **Basic (70%):**
    - **Description:** Use instruction COMF and INCF to convert the value of address [0x000] into a negative number in the form of two's complement format. After that, compare the converted value with the value of address [0x001]. If the two values are equal, set the value of data memory address [0x002] to 0xFF; otherwise, set it to 0x01.
    - **Sample test data one:**
      1. Put 0b01111100 (0x7C) in address [0x000].
      2. Put 0b00000100 (0x04) in address [0x001].
      3. After that, the value of address [0x000] will become 0b10000100 (0x84) after it is converted.
      4. Compare the two values stored on addresses [0x001] and [0x000].
      5. Since the two values are different, the value of address[0x002] should be set to 0x01 as the answer.
    - **Sample test data two:**
      1. Put 0b11111100 (0xFC) in address [0x000].
      2. put 0b00000100 (0x04) in address [0x001].
      3. After that, the value of address [0x000] will become 0b00000100 (0x04) after it is converted.
      4. Compare the two values stored on addresses [0x001] and [0x000].
      5. Since the two values are equal, the value of address [0x002] should be set to 0xFF as the answer.

- **Standard of grading:**
  1. YOU MUST USE the two instructions COMF and INCF in your program.
  2. Two sample test data will be used to examine the correctness of your program, and they are as same as the test data in the previous two examples, i.e., sample test data 1 and sample test data 2.
  3. The result should be put in the address [0x002].
  4. Please open File Registers in your uploaded demo vedio. Demonstrate and explain the execution process of your program by inputing the two data sets separately.

- **Advanced(30%):**
  - **Description:** Please design a loop using ANDWF and RRNCF instructions to count how many bit-value 1 in 0b00010111, and put the result in address [0x002].
  - **Standard of grading:**
    1. YOU MUST USE the two instructions ANDWF and RRNCF in your program.
    2. Demo test data is as same as the test data in the description (0b00010111).
    3. The result should be stored in address [0x002], and its value should be 0x04.
    4. The implementation way is not limited; however, YOU MUST use loop in your program.
    5. Please open File Registers in your uploaded demo vedio. Demonstrate and explain the execution process of your program by inputting the assigned test data.

- **Bonus (20%):**
  - **Description：** Implement XOR (exclusive or) function by using the following three instructions ANDWF, IORWF, and COMF.

Use your implemented XOR to manipulate the two values stored on addresses [0x000] and [0x001] and put the final result in address [0x002]. Please notice that you CANNOT directly apply instruction XORWF to finish the requirement.

- **Standard of grading:**
    1. You are only allowed to use the following operational logic instructions, i.e., ANDWF, IORWF, and COMF, in your program. You can use any of the given three instructions in your program. It is up to you.
    2. Using Instruction XORWF IS NOT ALLOWED.
    3. Your program will be examined using the sample test data 0b00001111 and 0b00110011.
    4. The final result should be put in address [0x002], and its value must be 0b00111100.
    5. Please open File Registers in your uploaded demo vedio. Demonstrate and explain the execution process of your program by inputting the assigned test data.