LAB 01 - Setup and Instruction Set

- MPLAB X IDE 的下載及安裝
  - Link: https://www.youtube.com/watch?v=akHZKwJf8D4
- Introduction to Instruction Set
  - Link: https://www.youtube.com/watch?v=rDVW7ZjWRyg
- Lab requirements
  - **基本題(70%)**

➢ **題目敘述**

首先將兩數字分別存入[0x000],[0x001]，並相加後把結果(A1)存入[0x002](A1<=0xFF)，接著將兩數字(下稱 C D)分別存入[0x010]、[0x011]，並相減把結果(A2)存入[0x012](C>=D)。

最後比較數字 A1 和 A2 的大小：

若 A1>A2，則將[0x020]設為 0xAA。

若 A1=A2，將[0x020]設為 0xBB。

若 A1<A2，將[0x020]設為 0xCC。

➢ **範例測資 (第一列為儲存位址)**

[0x000]	[0x001]	[0x002]	[0x010]	[0x011]	[0x012]	[0x020]
0x01	0x02	0x03	0x04	0x03	0x01	0xAA
0x11	0x12	0x23	0x2A	0x07	0x23	0xBB
0x07	0x09	0x10	0x12	0x01	0x11	0xCC

➢ **評分標準**

1. 結果需存放於正確儲存格

- **進階題(30%)**

➢ **題目敘述:**

首先將兩數分別放入[0x000]、[0x001]，接著將[0x000]的數字前 4bit 與[0x001]的數字後 4bit 組合成新數字，存入[0x002]。計算該數字中 bit 值為 0 的個數，將該結果存入[0x003]。

➢ **範例測資 (第一列為儲存位址):**

[0x000]	[0x001]	[0x002]	[0x003]
b'01010101	b'01111101	0x5D	0X03
0xFF	0X1E	0xFE	0x01

➢ **評分標準**

1. 結果需存放在正確儲存格

- **加分題(20%)**

➢ **題目敘述:**

將兩數存於[0x000], [0x010]，並設計迴圈，查看[0x000]數字是否為 4 的倍數。若為 4 的倍數，將[0x010]中數字+2。若不是，則查看是否為 2 的倍數。為 2 的倍數則[0x010]中數字+1，否則-1。每次迴圈要右旋(Right Rotate)位址[0x000]中數字，直到數字與原本存入值相同後結束迴圈。

➢ **範例測資 (第一列為儲存位址):**

step	[0x000]	[0x010]
存入	b' 10000001	0x05
1	b' 10000001	0x04
2	b' 11000000	0x06
3	b' 01100000	0x08
4	b' 00110000	0x0A
5	b' 00011000	0x0C
6	b' 00001100	0x0E
7	b' 00000110	0x0F
8	b' 00000011	0x0E
9	b' 10000001	Do nothing

➢ **評分標準**

1. 請使用迴圈，禁止暴力解，請善用 GOTO,DECFSZ 等指令。
2. 必須使用到指令 RRNCF

LAB 01 - Setup and Instruction Set

- **MPLAB X IDE Download and Install**
  - Link: https://www.youtube.com/watch?v=akHZKwJf8D4
- **Introduction to instruction set**
  - Link: https://www.youtube.com/watch?v=rDVW7ZjWRyg
- **Lab requirements**
  - **Basic (70%)**

➢ **Description:**

Store 2 numbers in memory locations [0x000], and [0x001], calculate sum of these numbers and store the result(A1) in [0x002] (the sum will not exceed 0xFF).

Store numbers (referred to as C and D) in memory locations [0x010], and [0x011] and calculate their difference(C-D). Store the result (A2) in [0x012] (C must be greater than or equal to D).

Determine whether A1>A2, A1=A2, or A1<A2:

If A1>A2 : Set the value of [0x020] as 0xAA

If A1=A2 : Set the value of [0x020] as 0xBB

If A1<A2 : Set the value of [0x020] as 0Xcc

➢ **Sample test data (First row is address):**

[0x000]	[0x001]	[0x002]	[0x010]	[0x011]	[0x012]	[0x020]
0x01	0x02	0x03	0x04	0x03	0x01	0xAA
0x11	0x12	0x23	0x2A	0x07	0x23	0xBB
0x07	0x09	0x10	0x12	0x01	0x11	0xCC

➢ **Criteria:**

1. You must store the results in correct memory locations.
2. Make good use of instructions CPFSEQ CPFSGT CPFSLT.

● Advanced (30%)

➢ **Description:**

Store 2 numbers in memory locations [0x000], and [0x001], combine the first 4 bits of the number in [0x000] with the last 4 bits of the number in [0x001] to form a new number, and store the number in memory location [0x002]. Count the number of bits in the new number that are 0, and store the result in [0x003].

➢ **Sample test data:**

[0x000]	[0x001]	[0x002]	[0x003]
b'01010101	b'01111101	0x5D	0X03
0xFF	0X1E	0xFE	0x01

➢ **Criteria:**

1. You must store the results in correct memory locations.
2. You can design the process on your own.

- Bonus (20%)

- **Description:**

  Store 2 numbers in memory locations [0x000], and [0x010],and design a loop to check whether the number at memory location [0x000] is a multiple of 4. If it is a multiple of 4, add 2 to the number in [0x010]. If not, check if it is a multiple of 2. If it is a multiple of 2, add 1 to the number in [0x010]. Otherwise, subtract 1 from the number in [0x010]. Right rotate the number in [0x000] each time in a loop until the number matches the originally stored number, then stop.

- **Sample test data:**

step	[0x000]	[0x010]
Store the numbers	b' 10000001	0x05
1	b' 10000001	0x04
2	b' 11000000	0x06
3	b' 01100000	0x08
4	b' 00110000	0x0A
5	b' 00011000	0x0C
6	b' 00001100	0x0E
7	b' 00000110	0x0F
8	b' 00000011	0x0E
9	b' 10000001	Do nothing

- **Criteria**

  1. Please use a loop to complete this task without using any brute force methods. Make good use of instructions like

GOTO, DECFSZ.

2. You must use the instruction RRNCF.