

1. Assume a 32-bit word at memory address 0x40000000 contains 0xBEEFFACE.

Write a program (including 5 **independent** steps below) to

- (1) insert the value 0x8765 into the word so that the final value is 0xB8765ACE,
- (2) set bits 5, 9, 13 and 17 of the word and leave the remaining bits unchanged,
- (3) use **one way** to clear bits 7, 10, and 16 of the word and leave the remaining bits unchanged,
- (4) use **another way** to clear bits 7, 10, and 16 of the word and leave the remaining bits unchanged,
- (5) change bits 18, 25, and 26 of the word and leave the remaining bits unchanged.

2. (1) Compute $j = (2^n + 2^m - 2^p) - 48$ and put j in $r5$, assuming $n = 7$, $m = 6$ and $p = 5$ are respectively in $r2$, $r3$ and $r4$ initially.

- (2) Assume 0xDEADABCD, 0xABCD8765 and 0xBEEFFACE are respectively in memory at addresses 0x40000010, 0x40000020 and 0x40000030. **Write a program** to calculate

- (a) the number of ones (using **TST**) in the word at address 0x40000010.
- (b) the number of ones (using **AND**) in the word at address 0x40000020.
- (c) the number of zeros (using **TST** and **without** using **SUB**) in the word at address 0x40000030
- (d) the number of different bits between the 2 words at addresses 0x40000020 and 0x40000030

leave the results of (a), (b), (c) and (d) respectively in the **bytes** at addresses **0x40000040, 0x40000044, 0x40000048 and 0x4000004C**. (Note: Be sure to use **loops**.)

3. **Write a program** to declare the following variables.

data1 DCB "Midterm Exam in Fall 2023!", 0
data2 DCW 0x1234, 0x5678, 0xBEEF, 0xFACE
data3 DCD 0x8ECC, 0xFE37, 0xABCD, 1, 5, 0x1234FACE
data4 DCB 0xCF, 23, 39, 0x54, 250, 0xFF, 0xAD,
data5 DCD 0xFE37, 1, 5, 20, 0xABCDFACE, 0x12345678

- (1) Point out **each stored value** from the memory window and write down the address of each variable (data1~data5).

(Be sure to use loops in Problems (2)~(9) below.)

- (2) Store data1 into memory started from address 0x40000000.
- (3) Reverse the string in data1 and put the reversed string in the memory started at address 0x40000030.

- (4) Store data2 into memory started from address 0x40000060.
- (5) Store data3 into memory started from address 0x40000070.
- (6) Reverse data3 and put the reversed data3 in the memory started from address 0x40000090.
- (7) Store the string data4 into memory started from address 0x400000B0.
- (8) Store data5 into memory started from address 0x400000C0
- (9) Add the 6 words in data5 and put the sum in the word at memory address 0x400000E0.

Note: Please

- (1) put necessary **Keil Tool DEBUG window screenshots** to show your **program** and **execution results** including **highlighted necessary initial assumptions and subsequent memory and register changes**,
- (2) **comment student ID+your English name in every screenshots**, and
- (3) put reports into one word file named by student_ID+your_name.