1. (12 points, 3 points each) Perform the following using the last 3 digits of your true ERAU ID (write the value here $N_{ID} =$):
    a. Calculate the remainder of $N_{10,7b} = N_{ID}\%128 =$ (7b here stands for 7 bits)
    b. Calculate the remainder of $N_{10,8b} = N_{ID}\%256 =$ (8b here stands for 8 bits)
    c. Determine the expression of $N_{10,8b}$ in hexadecima,l form : N_{16} =0x
    d. Determine the expression of $N_{10,8b}$ in binary form : N_2 =0b

    Note that we will use the above values below. Sometimes, we use the above **mathematical** notations, other times, we will use the corresponding **text** notations. The latter happens in the code snippets.

2. (24 points total, 8 points each—4 points for the correct operation result in decimal, 1 point for the result in binary, and 3 points for the C or V flag) Determine the result in both decimal and binary format and the C or V flag of the following calculations for an 8-bit system using the C and V flag convention of an ARM processor:
    a. Unsigned operation of (-128) - $N_{10,8b}$
    b. Unsigned operation of (-128) + $N_{10,8b}$
    c. Signed operation of (-128) + (-$N_{10,7b}$)

3. (12 points total, 3 points each) Assume `A` is an uint16_t variable and N_10_8b be the value of $N_{10,8b}$ determined in Prob 1. Determine the following values in hexadecimal form after running each line of code:

```
A = ((uint16_t)N_10_8b << 8) + N_10_8b;

A ^= 0x0FF0;

A |= 0x0FF0;

A &= 0x0FFF;
```

4. (18 points total, 3 points each) Determine the **last 8 bits** of the values of r4 to r9 in both hexadecimal and binary after running the following code assuming r1 is preloaded with an 8-bit value of $N_{10,8b}$.

```
asm_logic_code  PROC
        PUSH {r4-r9, lr}

        LDR  r1, =0xF0          ; r1 = 0x000000F0 after executing this command
        LDR  r2, =0x0F          ; r2 = 0x0000000F after executing this command
        LDR  r3, =0x50          ; r3 = 0x00000050 after executing this command

        MVN  r4, r0             ; r4 = 0x    = 0b
        ORN  r5, r0, r1         ; r5 = 0x    = 0b

        AND  r6, r0, r2         ; r6 = 0x    = 0b
        ORR  r7, r6, r3         ; r7 = 0x    = 0b

        EOR  r8, r0, r2         ; r8 = 0x    = 0b
        BIC  r9, r0, r1         ; r9 = 0x    = 0b
```

5. (9 points total, 3 points each) Assume we are using an 8-bit system. Determine the binary expressions of $-N_{10,7b}$ using sign-and-magnitude, one's complement, and two's complement, respectively.

6. (6 points total) Assume we want to express real numbers using a Q2.5 system. Determine the value of the real number represented by an 8-bit code of 128 + $N_{10,7b}$ using this system.

7. (12 points total) Consider the following C code snippet:

```
// Global variable
uint8_t ipt1[6];     // Input data 1
uint16_t ipt2[3];    // Input data 2

int main(void) {
    for (int i = 0; i < 3; i++) {
        ipt1[2*i] = i*10 + N_10_7b;
        ipt2[i] = i*128 + N_10_7b;
    }
}
```

Assuming the addresses of arrays `ipt1` and `ipt2` are `0x2000_0000` and `0x2000_0010`, respectively. Draw the memory map (contents in decimal plus addresses in hexadecimal). (Please use the following assumptions: (1) the contents of the memory is filled with values of 0x7F before the running of the main function; (2) we use little-endian for the number expression; and (3) the address at the top of the drawing is the lower address.)

a. (6 points) Draw the memory map of the first 6 (byte) addresses starting from `ipt1`.
b. (6 points) Draw the memory map of the first 6 (byte) addresses starting from `ipt2`.

8. (7 points points) Assume the 16 BYTE values starting from address `0x2000_0000` are 0+N_10_7b, 1+N_10_7b, 2+N_10_7b, …, 15+N_10_7b. Consider the running of the following asm code (using little-endian):

```
LDR r4, =0x20000000;  (1)
LDR r0, [r4, #2];     (2)
LDR r1, [r4, #2]!;    (3)
LDR r2, [r4], #4;     (4)
LDR r3, [r4];         (5)
```

Answer the following questions in hexadecimal format:

a. (2 points) What are the value of `r0` and `r4` after running (2)?
b. (2 points) What are the value of `r1` and `r4` after running (3)?
c. (2 points) What are the value of `r2` and `r4` after running (4)?
d. (1 point) What is the value of `r3` after running (5)?