Lab Task # 09

1. Write a program that shifts multiple bytes.

```
.data
        byte1 byte 3Bh
        byte2 byte 46h
        byte3 byte 0FFh
After execution of program the value should be
        byte1=03
        byte2=B4
        byte3=6F
```

2. Write an assembly program to input any **16bit** value in AX register and do the followings;
   a. Swap every pair of bits in the AX register.
   b. Swap the nibbles in each byte of the AX register.
3. AX contains a non-zero number. Count the number of ones in it and store the result back in AX.

   Repeat the process on the result (AX) until AX contains one. Calculate in BX the number of iterations it took to make AX one.

   For example BX should contain 2 in the following case:

   AX = 1100 0101 1010 0011 (input - 8 ones)

   AX = 0000 0000 0000 1000 (after first iteration - 1 one)

   AX = 0000 0000 0000 0001 (after second iteration - 1 one) STOP

4. **Bitwise Multiplication**

   Write a procedure named **BitwiseMultiply** that multiplies any unsigned 32-bit integer by EAX, using only shifting and addition. Pass the integer to the procedure in the EBX register, and return the product in the EAX register. Write a short test program that calls the procedure and displays the product. (We will assume that the product is never larger than 32 bits.) This is a fairly challenging program to write.

   One possible approach is to use a loop to shift the multiplier to the right, keeping track of the number of shifts that occur before the Carry flag is set. The resulting shift count can then be applied to the SHR instruction, using the multiplicand as the destination operand. Then, the same process must be repeated until you find the next highest bit in the multiplier.

5. Declare a WORD array having length of six elements, initialize it with Hex values from user? Your task is to replace central BYTE of each array element with corresponding element of the following BYTE array; (Use Shifting & Rotations)

   Given : arrayByte byte 0ABh, 0BCh, 0CDh, 0DEh, 0EFh, 0FAh

   Sample Input Array: arrayInput word 01234h, 05678h, 0789Ah, 09ABCh, 0BCDEh, 0DE12h

   Output Array Elements: 1AB4, 5BC8, 7CDA...etc