Chapter 4 - Bit Manipulations

For all the programs given below, use best instructions studied so far. For bit testing try both, the TEST instruction and shifting.

New Instructions:

```
CLC     ; Clear Carry Flag C=0
STC     ; Set Carry Flag C=1
CMC     ; Complement Carry Flag
```

Lab Manual

1- Calculate the number of one bits in BX and complement an equal number of least significant bits in AX. HINT: Use the XOR instruction and Rotate through carry

Sample Run:

Initial value of BX	Total No of 1 Bits in BX	Initial value of AX	AX after Complementing 7 least significant bits
1011 0001 1000 1001	7	1010 1011 1**010 0101**	1010 1 1**101 1010**

2- Write a program that shifts a 64-bit number.

3- Write a program that adds two 64-bit numbers.

4- Write a program to multiply two 32-bit numbers and store the answer in a 64-bit location.

Sample Run:

```
        a:      dq 0xABCDD4E1     ; dq allocates 64 bit memory space. a is 32-bit number but
                                  it has space allocation of 64 bits
        b:      dd 0xAB5C32       ; 32-bit space for multiplier
result: dq 0x0                    ; result should be 0x73005CB8FF6FF2 verify on calculator
programmer's view
```

5- Write a program to swap the nibbles (4-bits = 1 nibble) in each byte after complementing it of the AX register.

Sample Run:

AX before Swap	**1011 0010 0101 1101**	**0xB25D**
AX after Swap	**1101 0100 0010 1010**	**0xD42A**

6- Write a program that performs the following operations on the **AX register**:

Swap every adjacent pair of bits (e.g., swap bit #0 with bit #1, bit #2 with bit #3, bit #4 with bit #5, and so on).

Sample Run:

AX before Swap	**10 11 00 10 01 01 11 01**
AX after Swap	**01 11 00 01 10 10 11 10**