Assembly Quiz-2(H)

Total Marks: 10

Time Allowed: 15 min

Q1. In the following code, tell whether the jump will be taken or not. Show each and every step to get marks. [Marks 5]

```
mov ax, 0x7BFE
mov bx, 0xFFE1
add ax, bx
jo exit
```

Q2. Suppose AX contains any 16-bit number. Write a piece of code that clears 4th bit, set 14th bit and complements 7th bit of the number in ax register. [Marks 5]
(Numbering start from right side i.e., 0th bit is the least significant bit.)

1. ax = 7BFEh
   bx = FFE1h

After add ax, bx (17BDFh)
ax = 7BDFh (Discard MSB)

Adding negative to a positive number gives us a positive number, so there is no overflow and jump is not taken.

2.
```
and ax, FFEFh
or ax, 4000h
xor ax, 0080h
```

Solution

Total Marks: 10

Assembly Quiz-2(II)

Time Allowed: 15 min

Q1. In the following code, tell whether the jump will be taken or not. Show each and every step to get marks. [Marks 5]

```
mov ax, 0x7720
mov bx, 0x8601
add ax, bx
jo exit
```

Q2. Suppose AX contains any 16-bit number. Write a piece of code that clears 7th bit, set 2nd bit and complements 13th bit of the number in ax register.
(Numbering start from right side i.e., 0th bit is the least significant bit.)

1. ax = 7720h
   bx = 8601h

   7720
   8601
   FD21

   After add ax, bx (FD21h)

   ax = FD21h

   Adding negative to a positive number gives us a negative number, so there is no overflow and jump will not be taken.

2. and ax, FF7Fh
   or ax, 0004h
   xor ax, 4000h