Computer Organization And Assembly Language

Assignment # 1

Name : Junaid Saeed

Roll No : 22p-9200

Section : BCS-3D

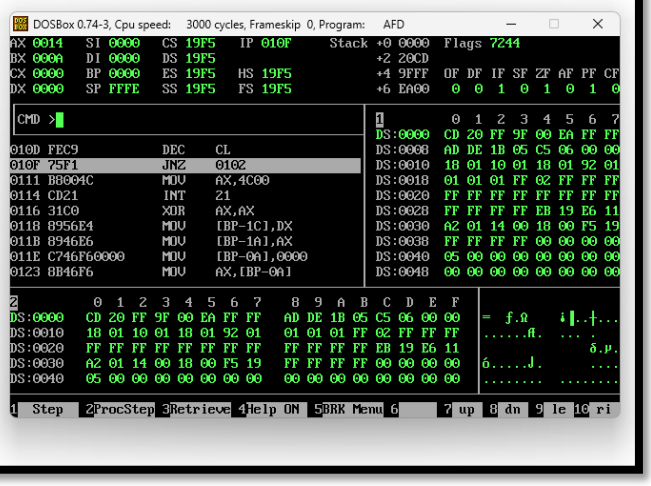
Question :-> 1

➔ **Write a program in assembly language for each of the below separately that sets ?**

➔ **Answer :**

a) ZERO FLAG(ZF) :

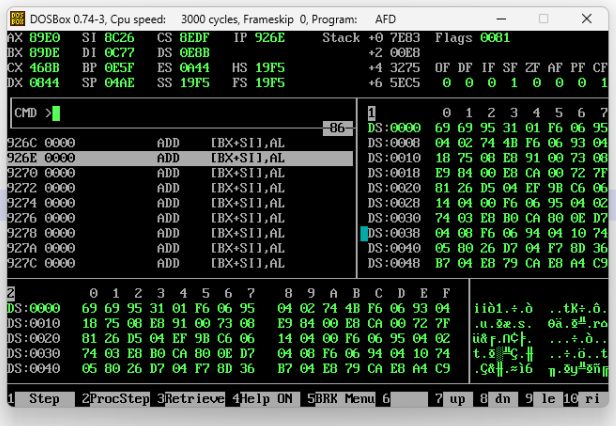
```
[org 0x0100]

mov cl, 2
start:
mov ax, 10
mov bx, 10
add ax, bx
cmp ax, 0
dec cl
jnz start

mov ax, 0x4c00
int 0x21
```



---

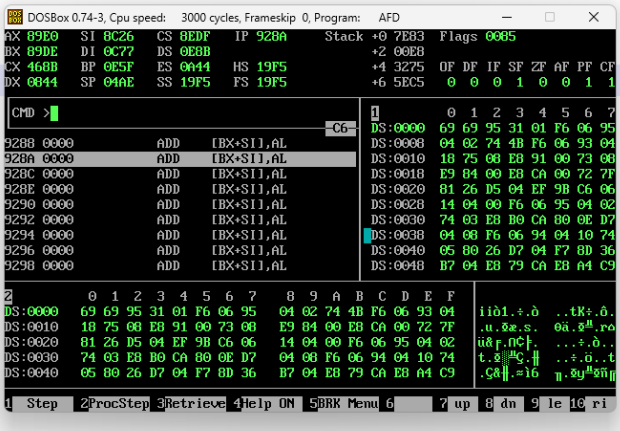
b) CARRY FALG(CF) :

```
//Carry flag.

[org 0x0100]

mov ax, [num1]
start:
add ax, [num2]
jnc start

mov ax, 0x4c00
int 0x21

num1: dw 78
num2: dw 12
```



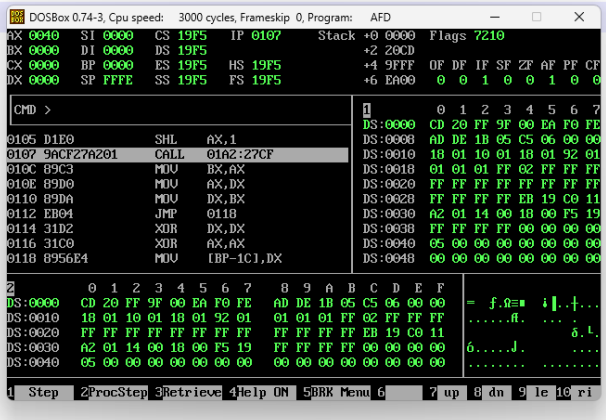
c) PARITY FLAG(PF) :

```
//Parity flag.

[org 0x0100]

    mov dx,40
    mov bx,80
    add dx,bx

    mov ax,0x4c00
    int 0x21
```



d) AUXILLIARY FLAG(AF) :

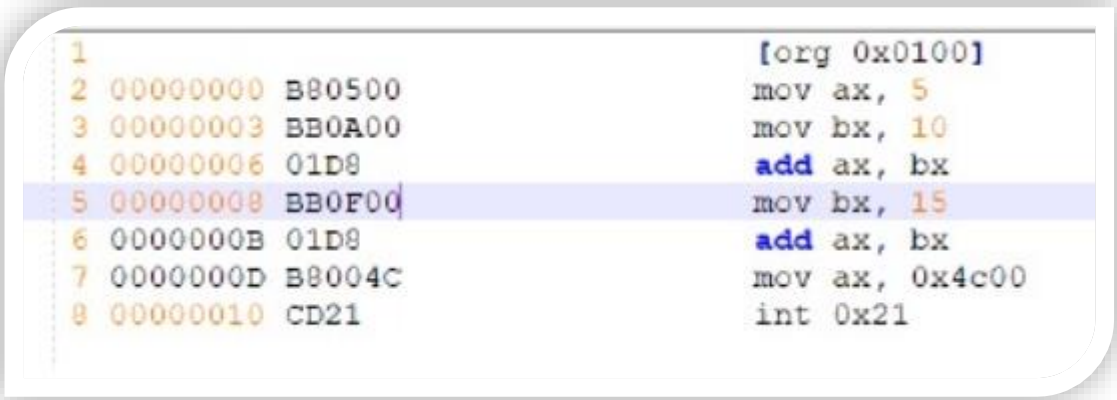
```
//auxilary flag.

[org 0x0100]

        mov al,20
        mov b1,30
start:
        add al,bl
        jnc start

        mov ax,0x4c00
        int 0x21
```



Question :-> 2

➔ **What will be the size of the following assembly language program in bytes?**

Answer :



```
1                                   [org 0x0100]
2 00000000 B80500                   mov ax, 5
3 00000003 BB0A00                   mov bx, 10
4 00000006 01D8                     add ax, bx
5 00000008 BB0F00                   mov bx, 15
6 0000000B 01D8                     add ax, bx
7 0000000D B8004C                   mov ax, 0x4c00
8 00000010 CD21                     int 0x21
```

Explanation ::

1. mov ax, 5 = 3 bytes.

2. mov bx, 10 = 3 bytes.
3. add ax, bx = 2 bytes.
4. mov bx, 15 = 3 bytes.
5. add ax, bx = 2 bytes.
6. mov ax, 0x4c00 = 3 bytes.
7. int 0x21 = 2 bytes.

➔ **Now, add up the sizes of each instruction:**

→ 3 + 3 + 2 + 3 + 2 + 3 + 2 = 18 bytes.

→ (The size is 18 bytes.)

Question :-> 3

➔ **Calculate the physical memory address generated by the following segment - offset pairs ?**

➔ **Answers :**

a) **1DDD:0436**
Physical adress = 1E206h

b) **1234:7920**
Physical adress = 19C60h

c) **74F0:2123**
Physical adress = 77023h

d) **0000:6727**
Physical adress = 06727h

e) **FFFF:4336**
Physical adress = 04326h

f) **<u>1080:0100</u>**
Physical adress = 10900h

➔ **<u>Solutions :</u>**

Physical adress = segment + offset.

→ add (0) at the end of segment
→ add (0) at the start of segment.

{ Solution }

(a) 1DDD : 0436

segment :- 1DDD0 .
offset :- 00436

A : 10
B : 11
C : 12
D : 13
E : 14
F : 15

```
   1 1  1 1 1  1 1 1
  0001 1101 1101 1101 0000
+ 0000 0000 0100 0011 0110
  0001 1110 0010 0000 0110
    ↓    ↓    ↓    ↓    ↓
    1    E    2    0    6
```

⇒ Physical adress = 1E206h

(b) 1234 : 7920

segment : 12340 .
offset : 07920

```
     1 1    1 1
  0001 0010 0011 0100 0000
+ 0000 0111 1001 0010 0000
  0001 1001 1100 0110 0000
    ↓    ↓    ↓    ↓    ↓
    1    9    C    6    0
```

⇒ Physical adress = 19C60h

(c) 74F0 : 2123

segment : 74F00 .
offset : 02123

```
  0111   0100   1111   0000   0000
+ 0000   0010   0001   0010   0011
  0111   0111   0000   0010   0011
    ↓      ↓      ↓      ↓      ↓
    7      7      0      2      3
```

Physical adress = 77023h.

(d) 0000 : 6727

segment : 0000 0
offset : 06727

```
  0000   0000   0000   0000   0000
+ 0000   0110   0111   0010   0111
  0000   0110   0111   0010   0111
    ↓      ↓      ↓      ↓      ↓
    0      6      7      2      7
```

physical adress = 06727h

(e) FFFF : 4336

segment : FFFF0
offset : 04336

```
 1111  1111  1111  1111  0000
+0000  0100  0011  0011  0110
-----------------------------
(1)0000 0100 0011 0010  0110
-----------------------------
  0     4     3     2     6
```

physical adress = 09326h.

(f) 1080 : 0100

segment : 10800
offset : 00100

```
 0001  0000  1000  0000  0000
+0000  0000  0001  0000  0000
-----------------------------
 0001  0000  1001  0000  0000
-----------------------------
  1     0     9     0     0
```

physical adress = 10900h.