Name: Abdul Ghani Khan

Roll No: 22P-9037

Section: BCS-3D

Course: Computer Organization & Assembly Language

Assignment: 01

Question No. 1:

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

the following flags.

(Write four programs i.e. One for each part)

A) Zero Flag

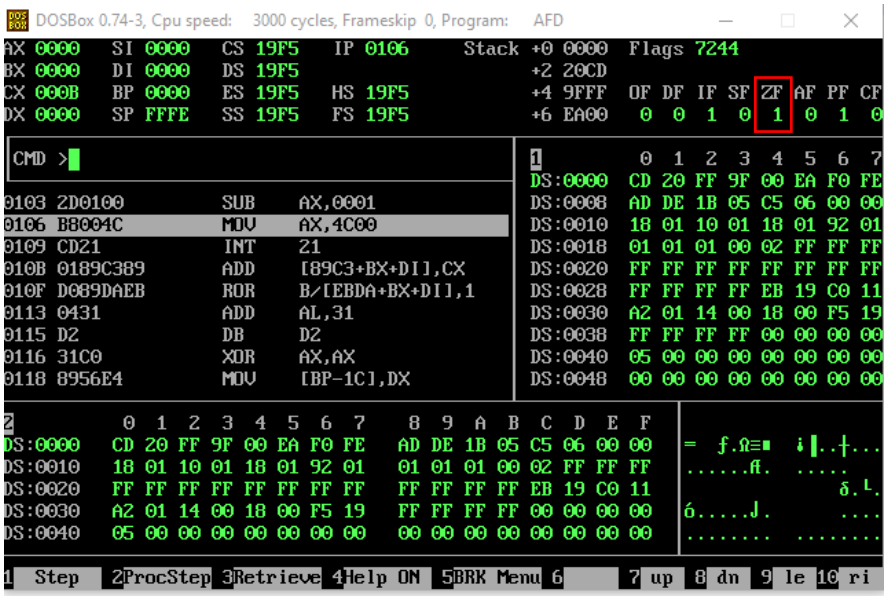
Code

```
[org 0x0100]

mov ax , 1
sub ax , 1

mov ax , 0x4c00
int 0x21
```

Result



B) Carry Flag

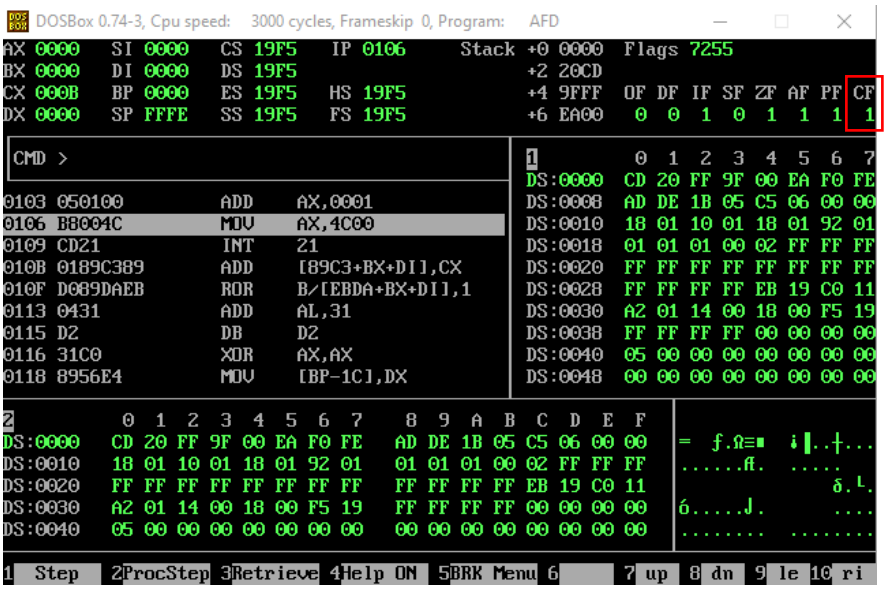
Code

```
[org 0x0100]

mov ax , 0xFFFF
add ax , 1

mov ax , 0x4c00
int 0x21
```

Result



C) Parity Flag

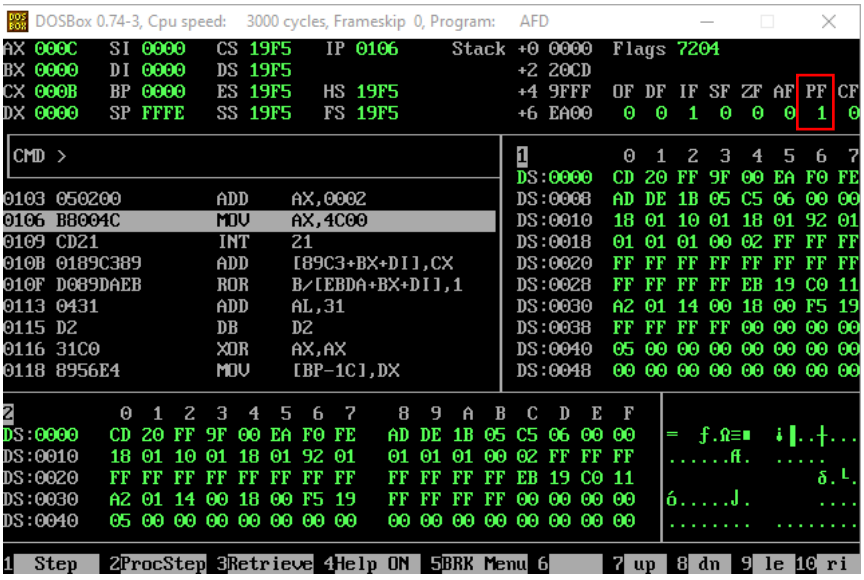
Code

```
[org 0x0100]

mov ax , 10
add ax , 2

mov ax , 0x4c00
int 0x21
```

Result



D) Auxiliary Flag

Code

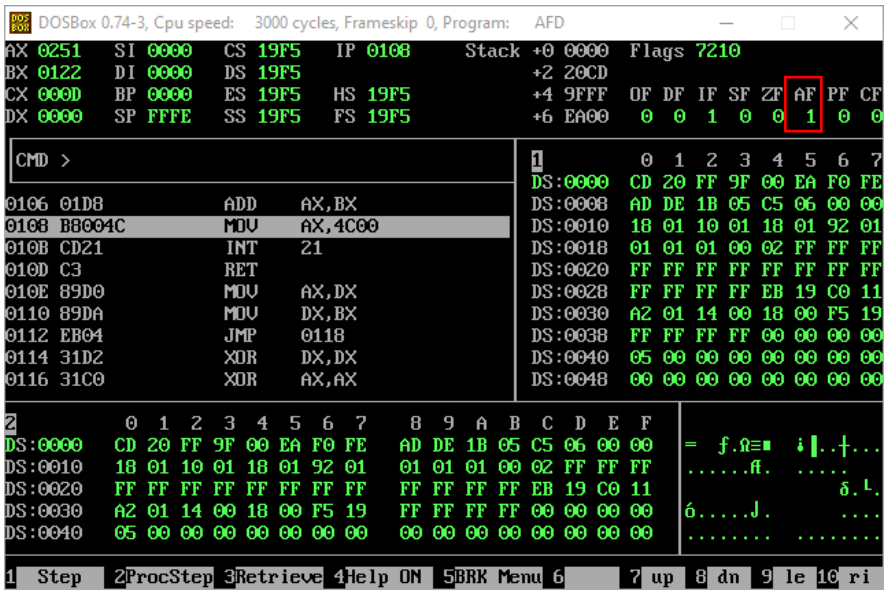
```
[org 0x0100]

mov bx , 0x0122
mov ax , 0x012F

add ax , bx

mov ax , 0x4c00
int 0x21
```

Result



Question No. 2:

What will be the size of the following assembly language program in bytes? Explain your answer using ". lst" file of this code.

```
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
8 0000000D B8004C                   mov ax, 0x4c00
9 00000010 CD21                     int 0x21
```

Explanation: -

Let's Break down one by one as we know that the mov instruction takes 3 bytes and its used four times so 3 * 4 is 12. Also, we know that the add instruction takes 2 bytes and it is used twice so 2 * 2 is 4 so 12 + 4 is 16. Now the last instruction (int 0x21) takes 2 bytes Now adding them 16 + 2 = 18 so this program takes 18 bytes

Line By Line If We Add: 3 + 3 + 2 + 3 + 2 + 3 + 2 = 18

Also looking at the lst file our answer matches by counting the number of bytes each line of the instructions is taking.

Question No. 3:

Calculate the physical memory address generated by the following segment-offset pairs:

A. 1DDD:0436

Physical Address: **1E206**

B. 1234:7920

Physical Address: **19C60**

C. 74F0:2123

Physical Address: **77023**

D. 0000:6727

Physical Address: **06727**

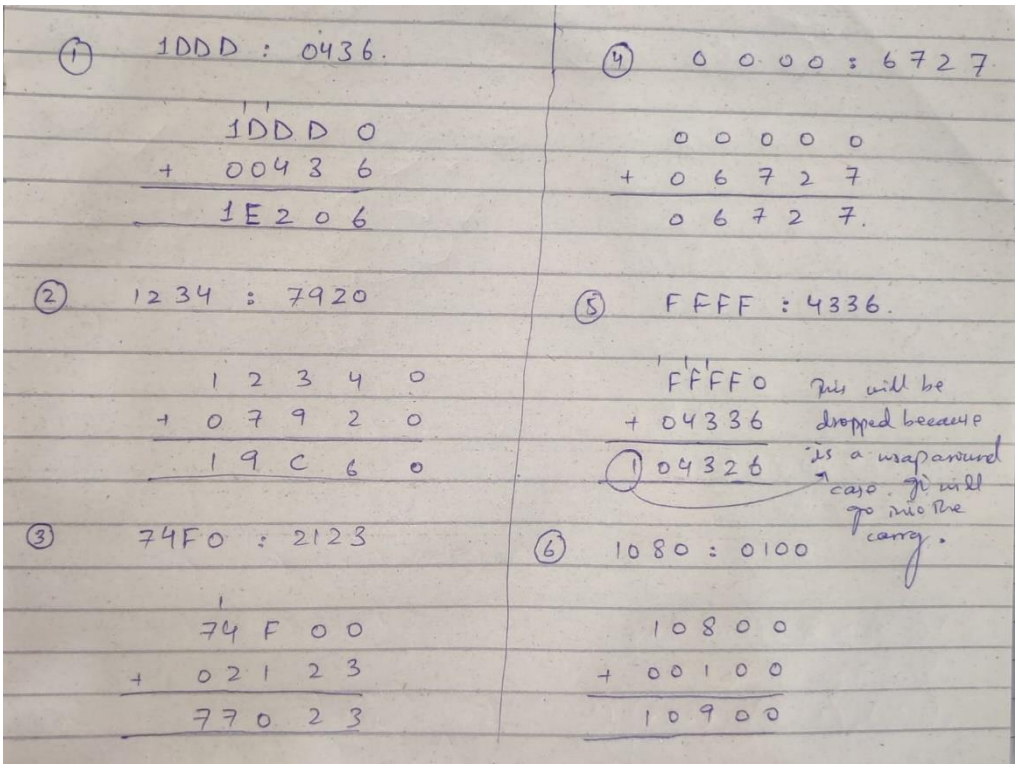
E. FFFF:4336

Physical Address: **04326**

F. 1080:0100

Physical Address: **10900**

Rough Work



(1) 1DDD : 0436.

```
  1DDD0
+ 00436
  1E206
```

(2) 1234 : 7920

```
  12340
+ 07920
  19C60
```

(3) 74F0 : 2123

```
  74F00
+ 02123
  77023
```

(4) 0000 : 6727

```
  00000
+ 06727
  06727.
```

(5) FFFF : 4336.

```
  FFFF0
+ 04336
(1)04326
```

This will be dropped because its a wraparound case. It will go into the carry.

(6) 1080 : 0100

```
  10800
+ 00100
  10900
```