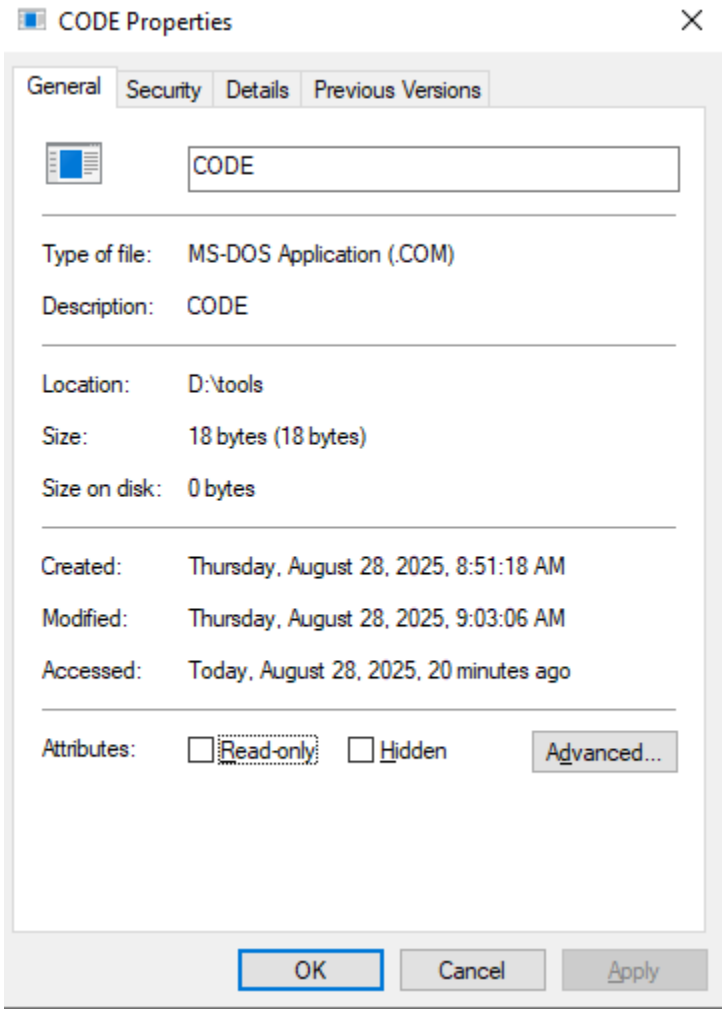
Activity 2:

1. B8
2. Yes, B8 is also used when moving terminate address in it (mov ax, 0x4c00)
3. B8 is opcode, and 0500 is the operand. 0500 is actually 0005 (5 in hexa) which is being moved to AX
4. BB, it is used in both instructions where memory is moved to BX.
5. 0100
6. Because the first and second instruction are 3 bytes long (as they are mov operations)
7. 18 bytes (3+3+2+3+2+3+2)
   mov ax, 5 ; 3 bytes
   mov bx, 10 ; 3 bytes
   add ax, bx ; 2 bytes
   mov bx, 15 ; 3 bytes
   add ax, bx ; 2 bytes
   mov ax, 0x4c00 ; 3 bytes
   int 0x21 ; 2 bytes
8. Yes it is 18 bytes (ss attached)

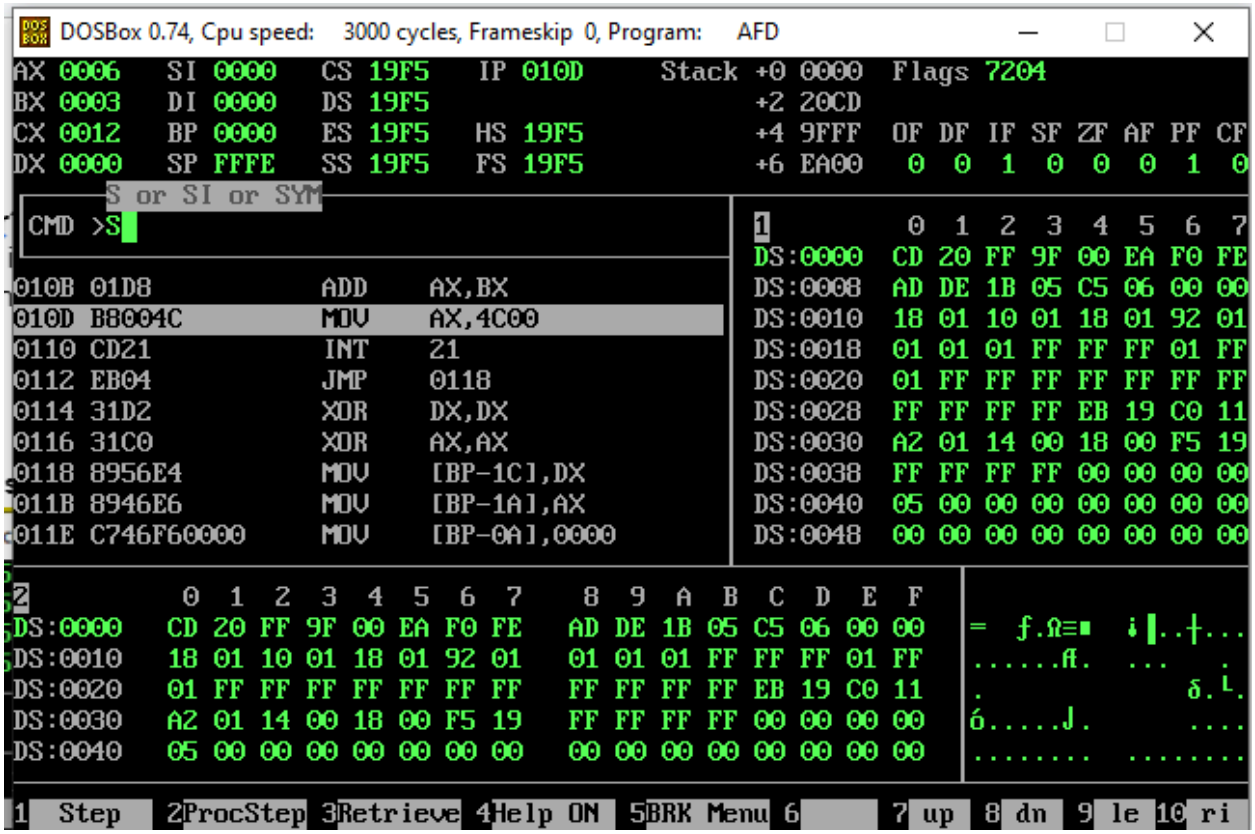


What is the value of IP register? And what will be its effect?
The start value of the IP register is 0100. This value represents the address of the first instruction in the code. This is where the program should start.

Activity 4:
[org 0x0100] ;we will see org directive later

```
    mov ax, 1 ; AX = 1
    mov bx, 2 ; BX = 2
    add ax, bx ; AX = AX + BX
    mov bx, 3 ; BX = 3
    add ax, bx ; AX = AX + BX

mov ax, 0x4c00 ;terminate the program
int 0x21
```

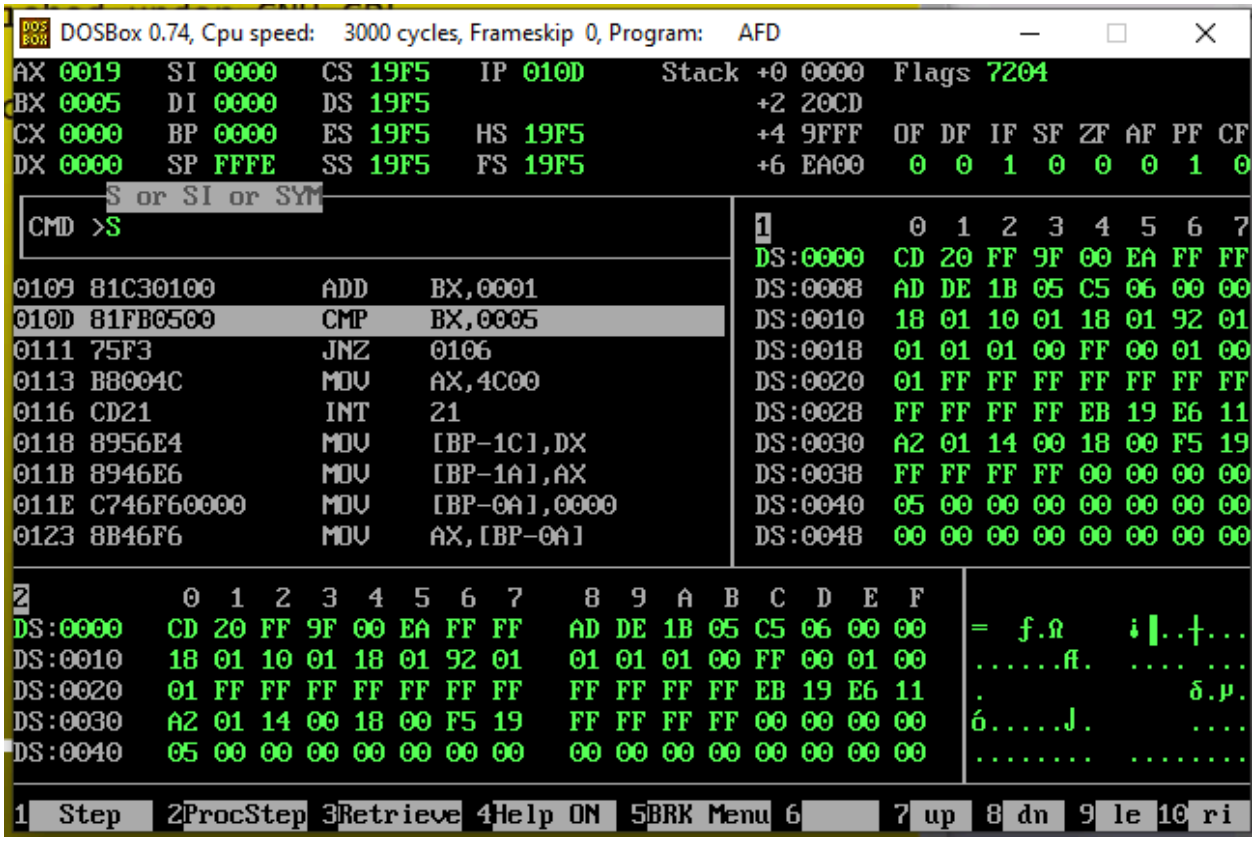


Activity 5: (square of 5 which is 0019h)

```
[org 0x0100] ;we will see org directive later

        mov ax, 5
        mov bx,1
l1:     add ax, 5
        add bx,1
        cmp bx,5
        jne l1

mov ax, 0x4c00 ;terminate the program
int 0x21
```



Activity 6:

```
[org 0x0100] ;we will see org directive later

        mov ax, 0x1234
        mov bx,0x3456
        mov cx, 0x1132
        add ax, bx
        sub ax, cx
        mov dx, ax

mov ax, 0x4c00 ;terminate the program
int 0x21
```