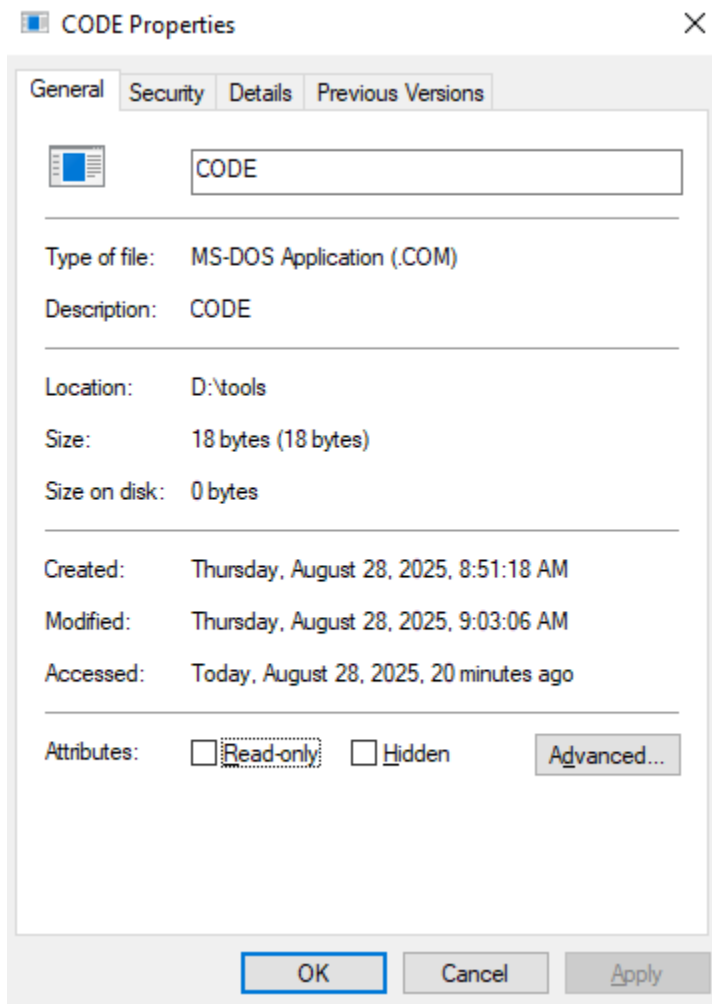


Activity 2:

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
```

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD

AX 0006 SI 0000 CS 19F5 IP 010D Stack +0 0000 Flags 7204
 BX 0003 DI 0000 DS 19F5 +2 20CD
 CX 0012 BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF
 DX 0000 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 0 0 0 1 0

S or SI or SYM

CMD >S

010B	01D8	ADD	AX,BX
010D	B8004C	MOV	AX,4C00
0110	CD21	INT	21
0112	EB04	JMP	011B
0114	31D2	XOR	DX,DX
0116	31C0	XOR	AX,AX
0118	8956E4	MOV	[BP-1C],DX
011B	8946E6	MOV	[BP-1A],AX
011E	C746F60000	MOV	[BP-0A],0000

1	0	1	2	3	4	5	6	7
DS:0000	CD	20	FF	9F	00	EA	F0	FE
DS:0008	AD	DE	1B	05	C5	06	00	00
DS:0010	18	01	10	01	18	01	92	01
DS:0018	01	01	01	FF	FF	FF	01	FF
DS:0020	01	FF	FF	FF	FF	FF	FF	FF
DS:0028	FF	FF	FF	FF	EB	19	C0	11
DS:0030	A2	01	14	00	18	00	F5	19
DS:0038	FF	FF	FF	FF	00	00	00	00
DS:0040	05	00	00	00	00	00	00	00
DS:0048	00	00	00	00	00	00	00	00

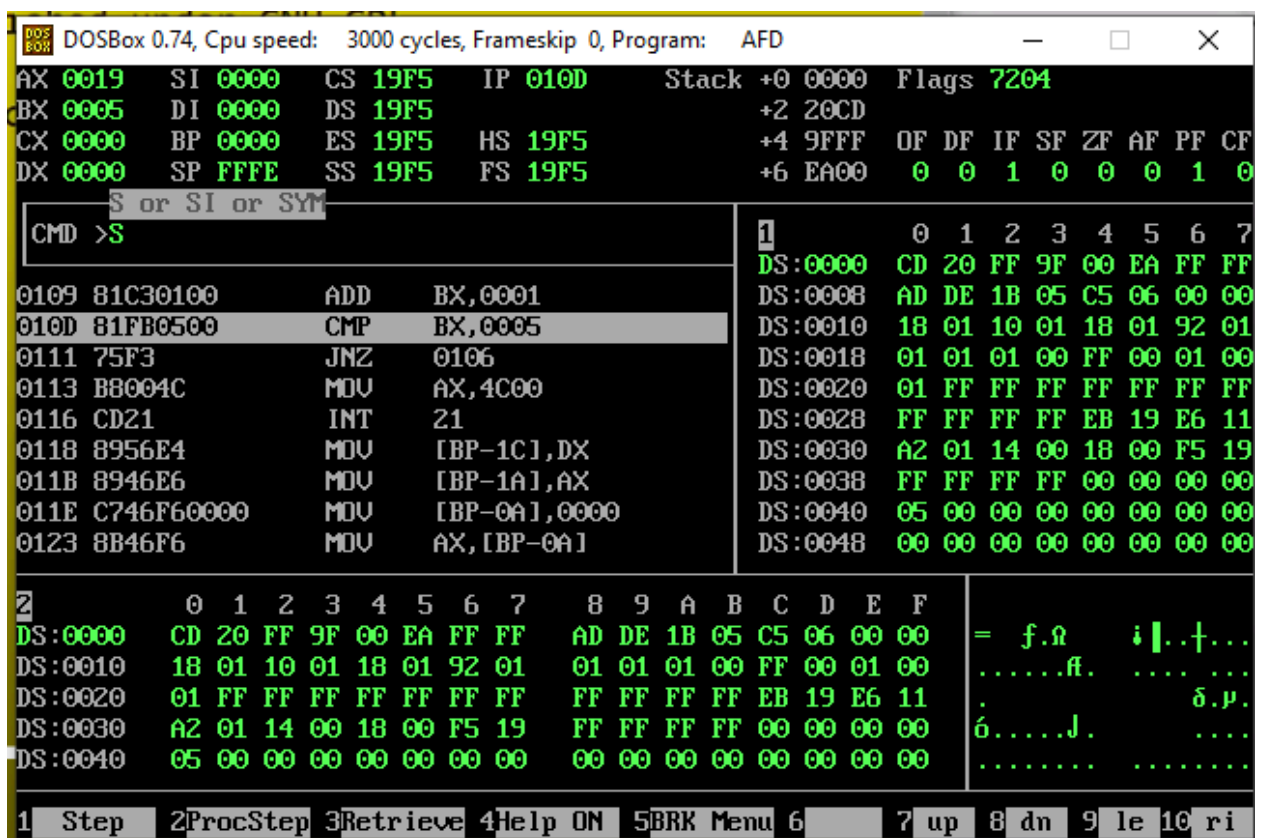
2	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
DS:0000	CD	20	FF	9F	00	EA	F0	FE	AD	DE	1B	05	C5	06	00	00
DS:0010	18	01	10	01	18	01	92	01	01	01	01	FF	FF	FF	01	FF
DS:0020	01	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	EB	19	C0	11
DS:0030	A2	01	14	00	18	00	F5	19	FF	FF	FF	FF	00	00	00	00
DS:0040	05	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

= f.Ω≡ i |..†...
ft. ...
 . δ.L.
 ó.....J.

1 Step 2ProcStep 3Retrieve 4Help ON 5BRK Menu 6 7 up 8 dn 9 le 10 ri

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

```
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
```

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD

AX	SI	CS	IP	Stack	Flags
3558	0000	19F5	010F	+0 0000	7200
BX	DI	DS		+2 20CD	
3456	0000	19F5			
CX	BP	ES	HS	+4 9FFF	OF DF IF SF ZF AF PF CF
1132	0000	19F5	19F5		
DX	SP	SS	FS	+6 EA00	0 0 1 0 0 0 0
3558	FFFE	19F5	19F5		

CMD >S

Address	Instruction	Comment
010D	89C2	MOV DX, AX
010F	B8004C	MOV AX, 4C00
0112	CD21	INT 21
0114	31D2	XOR DX, DX
0116	31C0	XOR AX, AX
0118	8956E4	MOV [BP-1C], DX
011B	8946E6	MOV [BP-1A], AX
011E	C746F60000	MOV [BP-0A], 0000
0123	8B46F6	MOV AX, [BP-0A]

Address	Hex	ASCII
DS:0000	CD 20 FF 9F 00 EA F0 FE	= f. n = i . . .
DS:0008	AD DE 1B 05 C5 06 00 00f. ...
DS:0010	18 01 10 01 18 01 92 01
DS:0018	01 01 01 FF FF FF 01 FF
DS:0020	01 FF FF FF FF FF FF FF
DS:0028	FF FF FF FF EB 19 C0 11
DS:0030	A2 01 14 00 18 00 F5 19	6.....J.
DS:0038	FF FF FF FF 00 00 00 00
DS:0040	05 00 00 00 00 00 00 00
DS:0048	00 00 00 00 00 00 00 00

1 Step 2ProcStep 3Retrieve 4Help ON 5BRK Menu 6 7 up 8 dn 9 le 10 ri