P1: Addressing Data in Memory and Segments (16 bits)

Part I

1. Write the program below and examine the register contents of **eax, ebx, ecx** and **edx** for each step (Press F10: Step Over).

a)

```
mov  cl, 42
mov  dl, 29
add  cl, dl
```

Command	EAX	EBX	ECX	EDX
mov cl, 42	7513342B	7EFDE000	0000002A	01271005
mov dl, 29	7513342B	7EFDE000	0000002A	012711D
add cl, dl	7513342B	7EFDE000	00000047	0127101D

b)

```
mov  ax,0123
add  ax,0025
mov  bx,ax
add  bx,ax
mov  cx,bx
sub  cx,ax
sub  ax,ax
```

Command	EAX	EBX	ECX	EDX
mov ax,0123	0133007B	0101F000	00B71005	00B71005
add ax,0025	01330094	0101F000	00B71005	00B71005
mov bx,ax	01330094	01010094	00B71005	00B71005
add bx,ax	01330094	01010128	00B71005	00B71005
mov cx,bx	01330094	01010128	00B70128	00B71005
sub cx,ax	01330094	01010128	00B70094	00B71005
sub ax,ax	01330000	01010128	00B70094	00B71005

2. To obtain the memory address that stores the executing command, examine the register contents of **EIP.**
   Write the following program and list the memory addresses that stores each line of codes. (HINT: disassembly)

```
mov ax, 56h
mov bx, 02h
mul bx
```

Command	**EIP**
`mov ax, 56h`	00AA1014
`mov bx, 02h`	00AA1018
`mul bx`	00AA101B

3. Disassembly the following machine code to assembly code/symbolic code:

   a) `B8 54 01 05 25 00`
   mov eax,25050154h
   add ah,cl

   b) `B8 05 1B 00 2C EB F8`
   mov eax,2C001B05h
   jmp _ExitProcess@4+0Fh (0F31027h)

4. Consider the machine language instructions

```
B0  1C  D0  E0  B3  12  F6  E3  EB  F6
```

   Which instruction performs the following operations?

   a) Move hex value `1C` to the `AL` register.
   mov al,1Ch
   b) Shift the contents of `AL` one bit to the left.
   shl al,1
   c) Move the hex value `12` to `BL`.
   mov bl,12h
   d) Multiply `AL` by `BL`.
   mul al,bl

Trace the program and find out the final product in AX? Confirm the result by manual calculation.

5. What is the output in AX?

```
MOV   AL, 5            ; AL = multiplicand
AX= 00D3FD05
MOV   BL, 10           ; BL = multiplier (operand)
AX= 00D3FD05
MUL   BL
AX= 00D30032
```

6. What is the output in AX and DX?

```
MOV  AX, 0083          ; dividend
AX= 001A0053
DX= 00CA1005
MOV  BL, 2             ; divisor (8 bits)
AX= 001A0053
DX= 00CA1005
DIV  BL
AX= 001A0129
DX= 00CA1018
```

Part II

1. Enter the following instructions:

```
MOV AX, 0010
MOV BX, 0020
MOV CX, 0030
ADD AX, BX
INC BX
SUB CX, AX
DEC CX
```

What is the content of register AX, BX, CX and IP for each instruction?

Value of registers

		AX	BX	CX
MOV	AX,010	006F000A	0051E000	012A1005
MOV	BX,020	006F000A	00510014	012A1005
MOV	CX,030	006F000A	00510014	012A001E
ADD	AX,BX	006F001E	00510014	012A001E
INC	BX	006F001E	00510015	012A001E
SUB	CX,AX	006F001E	00510015	012A0000
DEC	CX	006F001E	00510015	012AFFFF

What is the value in decimal for CX register?

65535

2. What is the final value of AX and BX?

```
     MOV  CX,3 ;Initialize for 3 loops
L1:  MOV  AX,00
     MOV  BX,00
     ADD  BX,AX
     LOOP L1   ;Decrement CX ;Repeat if nonzero
```

AX	BX	CX
00CFFEC4	00B64000	00070003
00CF0000	00B64000	00070003
00CF0000	00B60000	00070003
00CF0000	00B60000	00070003
00CF0000	00B60000	00070002