Question 1 [10 Marks]

Consider the following data declaration and fill in the given memory in hexadecimal (h).

Note*: *ASCII for 'A' = 041H

```
.data
          quard Qword 'ABCD',0ABCDABCDH
    doubleword dd 'ABC',01234ABCDH
               word 'AB','CD',0ABCDH
         mybyte db "AB",0ABh,23q,17t,1000101b
         byte2  sbyte -1,255,'A'+2*4
```

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
0100	44 D	43 C	42 B	41 A	00	00	00	00	CD	AB	CD	AB	00	00	00	00
0110	43 C	42 B	41 A	00	CD	AB	34	12	42 B	41 A	44 D	43 A	CD	AB	41 A	42 B
0120	AB	13	11	45	FF	FF	49									

Question 2 [10 Marks]

Update the given registers after executing following code. ***Note*: *Consider starting address at 0x0100***

```
.data
     size1=3
     ary1 db 1,2 dup(size1 dup(1,0AH))
     size1=5
     ary2 byte 2,1 dup(size1 dup(2))
     ary3 dw 2,3,
             7,8,9
          word 3
          sbyte 2
     bary4 LABEL BYTE
     wary5 LABEL WORD
     Dary6 dd 0ABCDEF12H,1
     Pary7 dd ary1
.code
     mov ax,0
     mov al,SIZEOF ary1
     mov ah,LENGTHOF ary2
     mov bl,LENGTHOF ary3
     mov esi,OFFSET ary1
     mov edi,pary7
     mov al,bary4
     mov bx,wary5
     mov cx,WORD PTR[Dary6+4]
```

01	01	0a	01	0a
01	0a	01	0a	01
0a	01	0a	02	02
02	02	02	02	02
00	03	00	07	00
08	00	09	00	03
00	02	12	EF	CD
AB	01	00	00	00
00	00	00	00	

AL	0D (13)
AH	*06*
BL	*05*
ESI	*0100*
EDI	*0100*
AL	*12*
BX	*EF12*
CX	*0001*

Question 3 [10 Marks]

Implement following C++ code using **LOOP** statement. Update the final value of SI after execution of the program.

```
int si=0;
for(int al=3;al>=0;al--)
{
    for(int bl=2;bl>=0;bl--)
    {
        for(int ecx=1;ecx>=0;ecx--)
        {
            cout<<si++;
        }
    }
}
```

SI= 24

```
mov si,0
mov ecx,
outer:
    mov eax, ecx
    mov ecx,3
    inner:
        mov ebx,ecx
        mov ecx,2
        innermost:
            inc si
        Loop innermost
       mov ecx,ebx
    Loop inner
    mov ecx,eax
Loop outer
```

Question 4 [8+2 Marks]

Consider the following data declaration. Copy **String** to **Stringcopy** using LOOP.

```
.data
string db "Encircle your course instructor name to score
the bonus"
s1=$-string
stringcopy dw s1 dup("?")
```

```
mov ecx,lengthof string1
mov ax,0
mov esi,offset string1
mov edi,offset stringcopy
L1:
     mov al,[esi]
     inc esi
     mov [edi],ax
     add edi,2

loop L1
mov bx,0
```

```
mov ecx,lengthof string1
mov ax,0
mov esi,0
mov edi,0
L1:
     mov al,string1[esi]
     inc esi
     mov stringcopy[edi],ax
     add edi,2
loop L1
mov bx,0
```

Question 5 [20 Marks]

i. Update the given flags after executing the following code? No marks for direct answer perform operation in rough work.

```
mov ax, 0FA12H
add ax, 5EEH
```

OF	SIGN	ZF	AF	PF	CF
0	**0**	**1**	**1**	**1**	**1**

```
 1 1111 1111 1111 11
   1111 1010 0001 0010
   0000 0101 1110 1110
   _________________
   0000 0000 0000 0000
```

ii. Write a code that add **v1** with **v2** and store result in **sum**. Your code should be for X86 architecture.

```
.data
      v1 sbyte -5
      v2 dw 0FABCh
      sum dw 0
.code
     mov ax,0
     mov ax,0FFFFh
      mov al,v1
      add ax,v2
      mov sum,ax
```

iii. Write a code that swap content of **v1** with **v2**.

```
.data
        v1 db 5
        v2 db 4
.code
     mov ax,0
     mov al,v1
     xchg al,v2
     xchg al,v1
```

iv. Consider the following data declaration and fill in the given memory in hexadecimal (h).

```
.data
        v1 db 1
        align WORD
        v2 word 12EFH
        v3 byte 3
        align DWORD
        v4 db 4
        align 2
        v5 dw 0ABCDH
```

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
0100	01	00	EF	12	03	00	00	00	04	00	CD	AB				
0130																