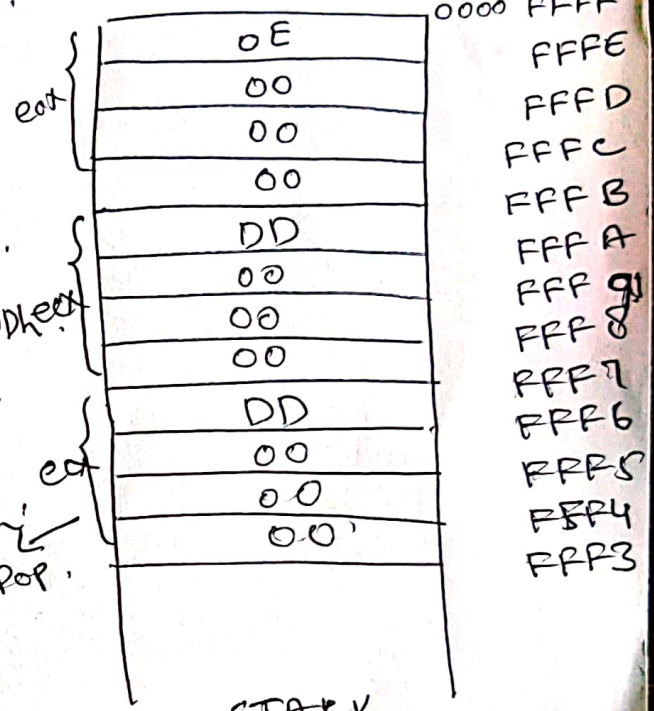
Ques No 1(a)

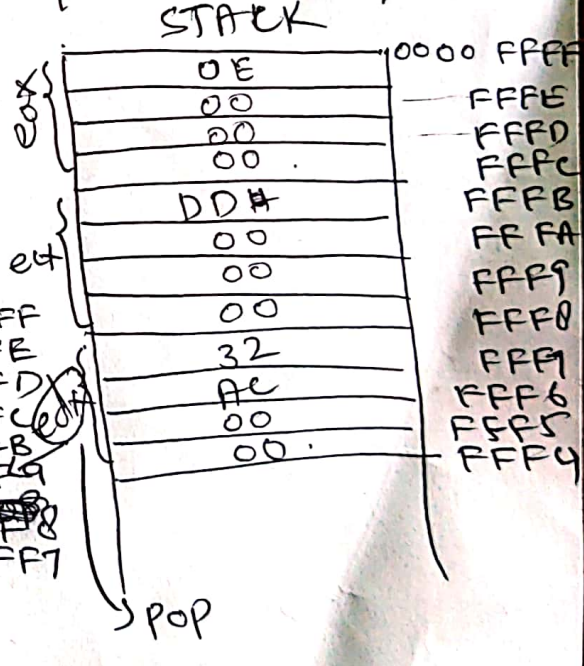
```
Main PROC.
    ADD  AX, 1      ; 0000 000Eh
    INC  DH         ; 0000 ACCDh
    PUSH EAX        ; 0000 000Eh
    PUSH ECX.       ; 0000 00DDh.
    CMP  CH, 0      ;  00 , 0.
    JNZ  L1         ; false.
    PUSH ECX        ; 0000 00DDh.
    POP  EBX        ; EBX = 0000 00DDh
L1: NOT  DL         ; 0000 AC32h
    PUSH EDX.       ; 0000 AC32h.
    POP  EAX.       ; EAX = 0000 AC32h
Main endp.
End Main
```

EAX = 0000 000Dh.
ECX = 0000 00DDh
EDX = 0000 ABCDh
ESP = 0000 FFFFh.

STACK



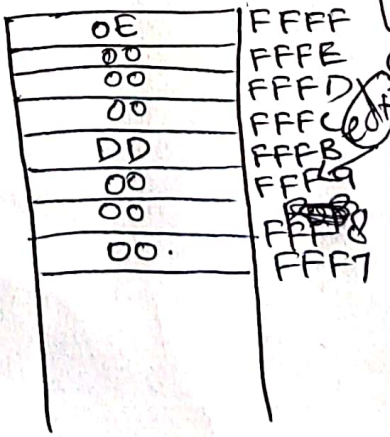
STACK



Final Values.

EAX = 0000 AC32h.
EBX = 0000 00DDh
ECX = 0000 00DDh
EDX = 0000 AC32h.

STACK.



EDX = 0000 ACCDh
DH DL

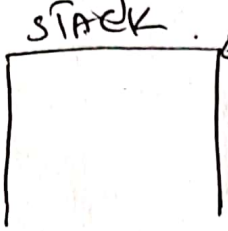
DL = CDh.
1100 1101
NOT DL = 0011 0010
3 2 h.

After NOT DL
↓
EDX = 0000 AC32h

Ques No 1 (b).

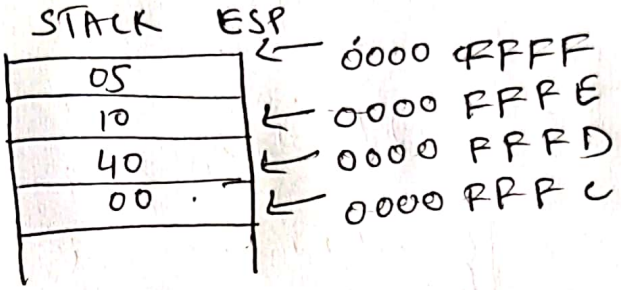
Assuming the ESP = 0000 FFFF

IP = 0040 1000

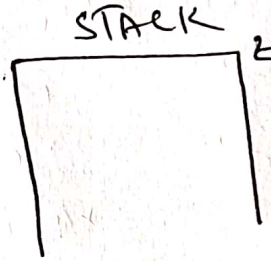


(Before func call)
Closer:

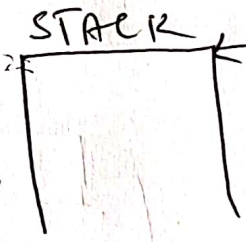
Closer:
IP = 0040 1023



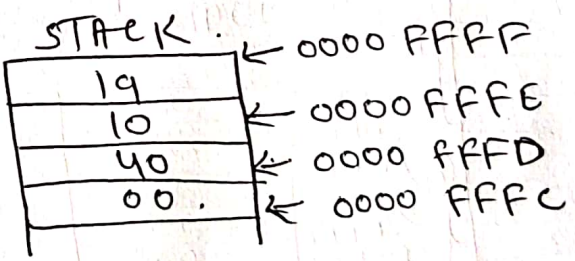
{Back to main}.
IP = 0040 1005



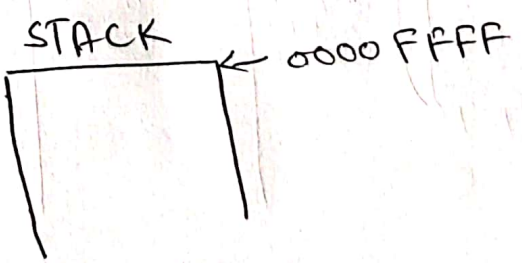
IP = 0040 1014



(ArraySum)
IP = 0040 102B



{Back to Main}.
IP = 0040 1019.



Qus No 2(a).

.data

list byte "computer organization and assembly language", 0.

toReplace byte ?

msgFound byte → Here is the updated string", 0.

msgNFound byte → sorry the character not found in the string", 0.

msg byte → Enter a character to search", 0.

sflag byte 0.

.code

Main PROC.

→ mov ecx, lengthof list - 1.

mov eax, 0.

mov edx, offset msg.

Call WriteString.

call ~~WriteChar~~ ReadChar ; eax/al

mov toReplace, '@'.

~~cmp al~~

→ mov esi, 0.

→ L1: cmp al, list[esi]

~~jcxz~~ jnz LIterate.

mov list[esi], '@'

mov sflag, 1

LIterate: inc esi

Loop L1.

cmp sflag, 0.

jz LmsgNFound.

~~mov~~ mov edx, offset msgFound.

Call WriteString.

jmp Lexit

LmsgNFound: mov edx, offset msgFound.

Call WriteString.

→ Lexit:

Main endp.

End Main.

Qus No 2(b)

~~int myarray~~

```
.data
    myarray byte 100 DUP(?)
    j byte 100
.code
    Main PROC
    mov ecx, lengthof myarray

L1: cmp j, 0
    jl Lexit Loop.
    cmp j, ecx
    jg Lexit Loop.
    mov al, myarray[j].
    mov bl myarray[j+1]
    mov myarray[j+1], al
    mov myarray[j], bl.
    dec j
    jmp L1
```

Qus No 3 a)

```
MOV   CL, 2h     ;  CL = 02h.
MOV   AL, 8Ch    ;  AL = 8C
MOV   BL, C8h    ;  BL = C8
SHL   AL, CL     ;  AL = 30h ; CF = 0.
SHR   BL, CL     ;  BL = 32h.
INC   CL.        ;  CL = 03h.
SAR   BL, CL     ;  BL = 06h, CF = 0
ROL   AL, CL     ;  AL = 80h  CF = 1
CLC              ;  CLC = 0
DEC   CL         ;  CL = 02h.
RCL   AL, CL     ;  AL = 02h  CF = 0
STC              ;  CF = 1
RCR   BL, CL     ;  BL = A0h  CF = 1
SHRD  AL, BL, 2  ;  AL = 00h. BL = A0h
                    CF = 0.
SHLD  BL, AL, 2  ;  BL = A0h, AL = 0h.
                    CF = 0
```

AL = 8C
= 1000 1100.
SHL AL,2 = 0011 0000.
CF = 0, 3, 0.

BL = C8.
= 1100 1000.
shr bl, 2 = 0011 0010.
CF = 0, (3 2)h.

BL = 32.
BL = 0011 0010.
sar bl, 3

		CF
1 →	0001 1001	0
2 →	0000 1100	1
3 →	0000 0110	0
	(0 6)h.	

AL = 30h
= 0011 0000
ROL AL, 3

CF		
	0110	0000
1	1100	0000
1	1000	0000
	(8	0)h.

AL = 80h.
= 1000 0000.
CL = 02h. ; CF = 0.
RCL AL, 2 →

CF		
0	1000	0000
1	0000	0000
0	0000	0001
0	0000	0010

AL = (0 2)h.

BL = 06h. CF = 1
ROR BL, 02.

		CF
1 →	0000	0110
0 →	1000	0011
1 →	0100	0001
	1010	0000
	(A	0)h

CF = 1

SHRD AL, BL, 2
02, A0, 2

0000 0010 | 1010 0000
0000 0000 | 1010 1000
(00) | (A 8)h
AL (00) BL (A8)
CF = 0.

SHRD BL, AL, 2.
A8 00, 2

CF 1010 1000 | 0000 0000
0 1010 1000 | 0000 0000 CF
(A 0)h | (0 0)h.
BL | AL.

Ques. No 3(b)

```
.data
Second  Word  00 00h.
Minute  Word  00 00h.
Hour    Word  00 00h.
TimeStamp Word. 00 100110  0110 1010
                   Year     Month  Day
```

0—4 : second
5—10 : Minutes.
11—15 : hours.

```
.code Main PROC
mov dx, TimeStamp.
mov al, dl
and al, 0001 1111b
movzx second, al.

mov ax, dx
shr ax, 5
and al, 0000 1111b
movzx Minute, al

mov al, dh
shr al, 1
mov ah, 0
[illegible].

mov Hour, ax

Main endp
End Main
```