

(Question 1)

Addition of 8-bit numbers

(Aim)

To find the sum of two 8-bit numbers using ALP

(Algorithm/Pseudocode)

BEGIN

AL \leftarrow 00H

BL \leftarrow 00H

AL \leftarrow [Store the content of the address stored in SI]

INC SI to get the address of next number

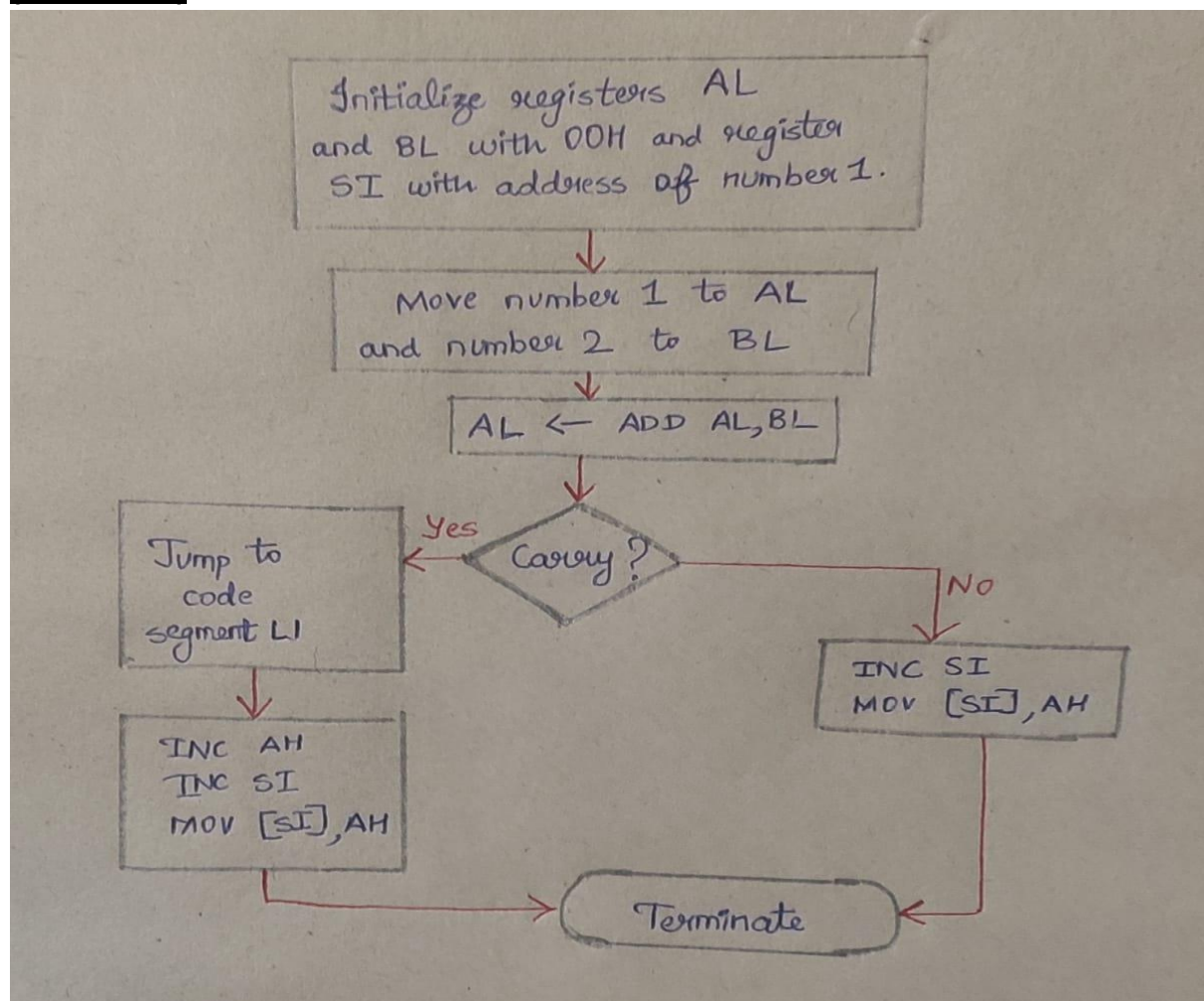
BL \leftarrow [Store the content of the address stored in SI]

ADD AL,BL

Store the result in [SI+]

END

(Flowchart)



(ALP Code)

```
code segment
assume cs:code
;ANISH DESAI
;20BCE0461
start:
    mov ax, 0000h
    mov si, 3000h
    mov al, [si]
    inc si
    mov bl, [si]
    add al, bl
    inc si
    mov [si], al
    jc l1
    inc si
    mov [si], ah
    int 3h
l1:
    inc ah
    inc si
    mov [si], ah
    int 3
code ends
end start
```

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
File Edit Search View Options Help
C:\ADD8BIT.ASM
code segment
assume cs:code
;ANISH DESAI
;20BCE0461
start:
    mov ah, 0000h
    mov si, 3000h
    mov al, [sil]
    inc si
    mov bl, [sil]
    add al, bl
    inc si
    mov [sil], al
    jc l1
    inc si
    mov [sil], ah
    int 3h
l1:
    inc ah
    inc si
    mov [sil], ah
    int 3
code ends
end start
F1=Help | Line:25 Col:1
```

(MASM Output)

```
C:\>debug add8bit.exe
-g=0000

AX=00BE BX=0006 CX=001B DX=0000 SP=0000 BP=0000 SI=3003 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=0014  NU UP EI PL NZ NA PE NC
076A:0014 CC          INT      3
-e 3000
075A:3000 BB.05 06.03

-g=0000

AX=0008 BX=0003 CX=001B DX=0000 SP=0000 BP=0000 SI=3003 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=0014  NU UP EI PL NZ NA PE NC
076A:0014 CC          INT      3
-q
C:\>
```

(Result)

As we can see in the above MASM output screen, we are executing the add8bit.exe.

We are passing 05H and 03H as inputs in AL and BL and the output 08H is stored in AL at the end.

(Question 2)

Addition of 16-bit numbers

(Aim)

To find the sum of two 16-bit numbers using ALP

(Algorithm/Pseudocode)

BEGIN

Load numbers in the registers AX and BX

Initialize register DX to 0000H

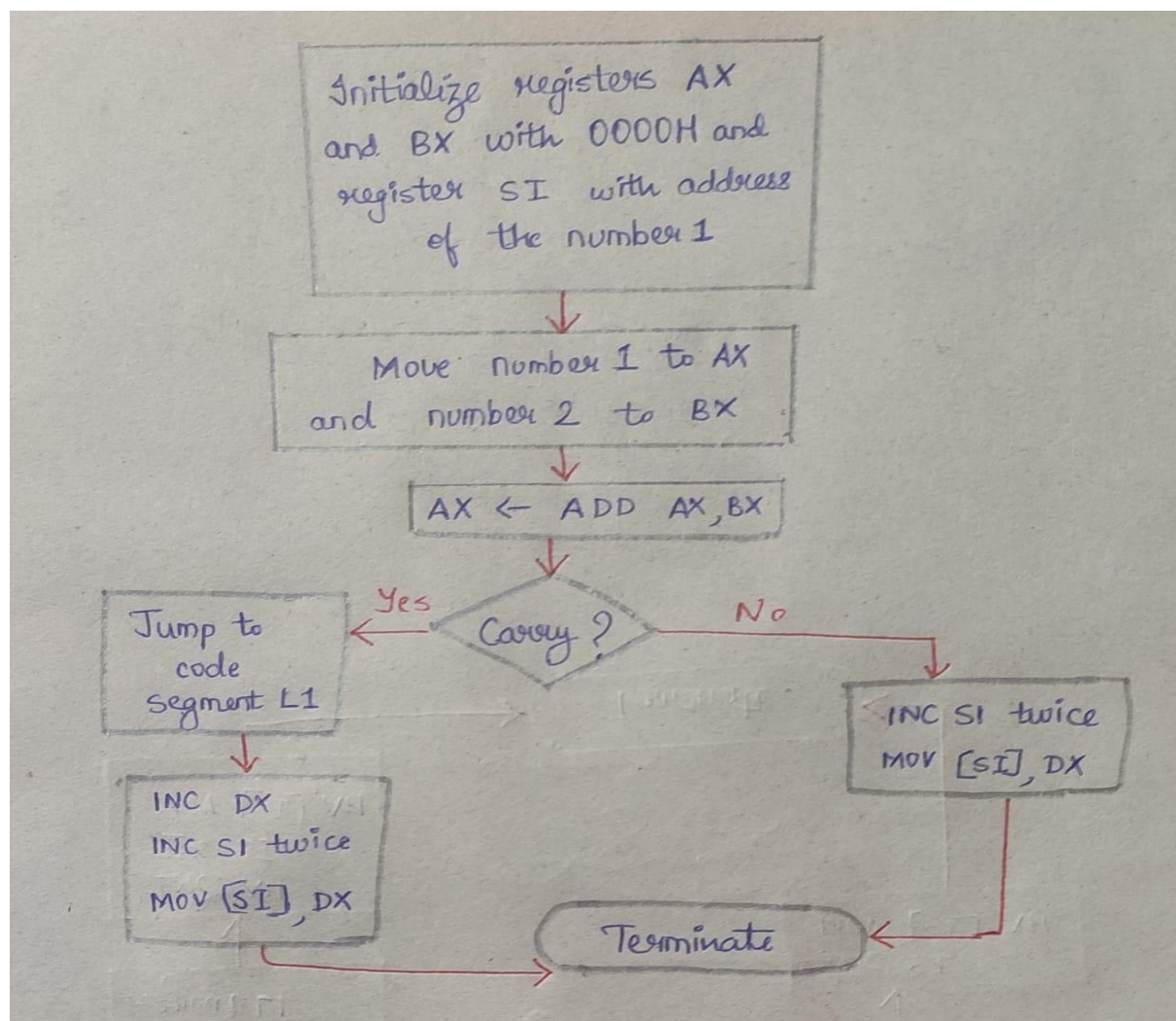
Add the values in AX and BX

IF carry exists, Increment DX

Else IF carry does not exist, Terminate

END

(Flowchart)



(ALP Code)

code segment
assume cs: code

;ANISH DESAI

;20BCE0461

start:

 mov ax,0000h

 mov bx,ax

 mov dx,ax

 mov si,3000h

 mov ax, [si]

 inc si

 inc si

 mov bx, [si]

 inc si

 inc si

 add ax,bx

 mov [si],ax

 jc l1

 inc si

 inc si

 mov [si], dx

 int 3

l1:

 inc dx

 inc si

 inc si

 mov [si],dx

 int 3

code ends

end start

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
File Edit Search View Options Help
C:\ADD16BIT.ASM
code segment
assume cs: code
;ANISH DESAI
;20BCE0461
start:
    mov ax,0000h
    mov bx,ax
    mov dx,ax
    mov si,3000h
    mov ax,[si]
    inc si
    inc si
    mov bx,[si]
    inc si
    inc si
    add ax,bx
    mov [si],ax
    jc l1
    inc si
    inc si
    mov [si],dx
    int 3

l1:
    inc dx
    inc si
    inc si
    mov [si],dx
    int 3
code ends
end start
F1=Help Line:31 Col:1
```

(MASM Output)

```
C:\>DEBUG ADD16BIT.EXE
-g=0000

AX=56B8 BX=5000 CX=0023 DX=0000 SP=0000 BP=0000 SI=3006 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=001C  NV UP EI PL NZ NA PE NC
076A:001C CC          INT     3
-e 3000
075A:3000 B8.16  06.5A  00.9A  50.7C

-g=0000

AX=D6B0 BX=7C9A CX=0023 DX=0000 SP=0000 BP=0000 SI=3006 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=001C  NV UP EI PL NZ NA PE NC
076A:001C CC          INT     3
```

(Result)

As we can see from the above MASM screen, we have entered two 16-bit numbers '5A16' and '7C9A'. The sum D6B0 is stored in the register AX and since there is no carry, therefore DX is 0000.

(Question 3)

Convert a given 8-bit number to its negative value. The ALP should contain 3 different ways to achieve the goal. The ALP code file name should be of the format NEG_<YOUR NAME>.asm

(Aim)

To negate an 8-bit number in 3 different ways.

(Algorithm/Pseudocode)

```
BEGIN
AX ← 0000H
SI ← 3000H
AL ← [Store the content of the address stored in SI]
MOV number in AL to BL and use NEG
MOV number in AL to CL and find 2's complement using NOT and INC
MOV number in AL to DH and 00H to DL
SUB DL with DH and resultant negative value stored in DL
CLEAR DH
END
```

(Flowchart)

No branch instructions, thus flowchart not required.

(ALP Code)

```
code segment
assume cs:code
;ANISH DESAI
;20BCE0461
start:
mov ax,0000H
mov si,3000H
mov al,[si]

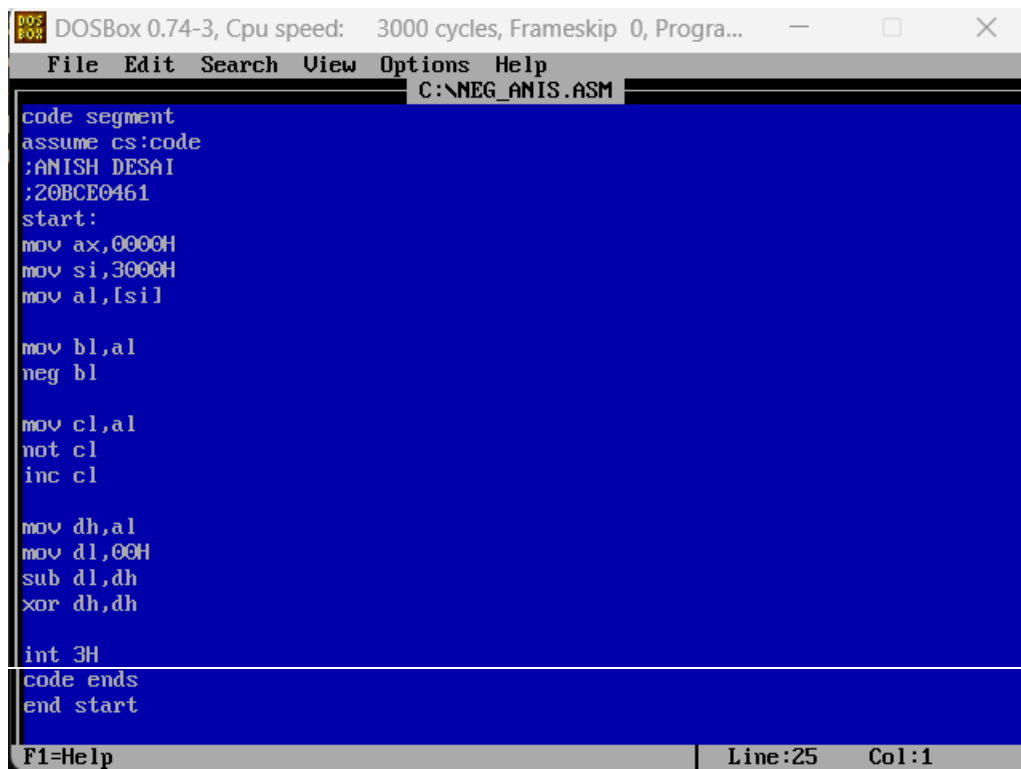
mov bl,al
neg bl

mov cl,al
not cl
```

inc cl

mov dh,al
mov dl,00H
sub dl,dh
xor dh,dh

int 3H
code ends
end start



The image shows a screenshot of a DOSBox window. The title bar reads "DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...". The menu bar includes "File", "Edit", "Search", "View", "Options", and "Help". The active window is titled "C:\NEG_ANIS.ASM". The main area displays assembly code on a blue background with white text. The code is as follows:

```
code segment
assume cs:code
;ANISH DESAI
;20BCE0461
start:
mov ax,0000H
mov si,3000H
mov al,[si]

mov bl,al
neg bl

mov cl,al
not cl
inc cl

mov dh,al
mov dl,00H
sub dl,dh
xor dh,dh

int 3H
code ends
end start
```

The status bar at the bottom shows "F1=Help" on the left, and "Line:25 Col:1" on the right.

(MASM Output)

```
C:\>DEBUG NEG_ANISH.EXE
-G=0000

AX=00B8 BX=0048 CX=0048 DX=0048 SP=0000 BP=0000 SI=3000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=001A  NU UP EI PL ZR NA PE NC
076A:001A CC          INT      3
-E 3000
075A:3000 B8.24

-G=0000

AX=0024 BX=00DC CX=00DC DX=00DC SP=0000 BP=0000 SI=3000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=001A  NU UP EI PL ZR NA PE NC
076A:001A CC          INT      3
-
```

(Result)

As we can see from the above MASM screen, we have entered an 8-bit number '24' in register 'AL'. We have computed the negative of the number stored in 'AL' in 3 different ways, using NEG in 'BL', using 2's complement in 'CL' and subtracting 'DL' with 00H and storing result in 'DL'. The negative of '24' is 'DC' and is reflected in the values stored in the three registers 'BL', 'CL' and 'DL'.
