ATMA RAM SANATAN DHARMA COLLEGE
University of Delhi

Microprocessor

Submitted By

Kaung Khant Htut

College Roll N0. 18094

BSc(Hons) Computer Science

Third Year (5th semester)

Q1. 32 bit Binary Addition

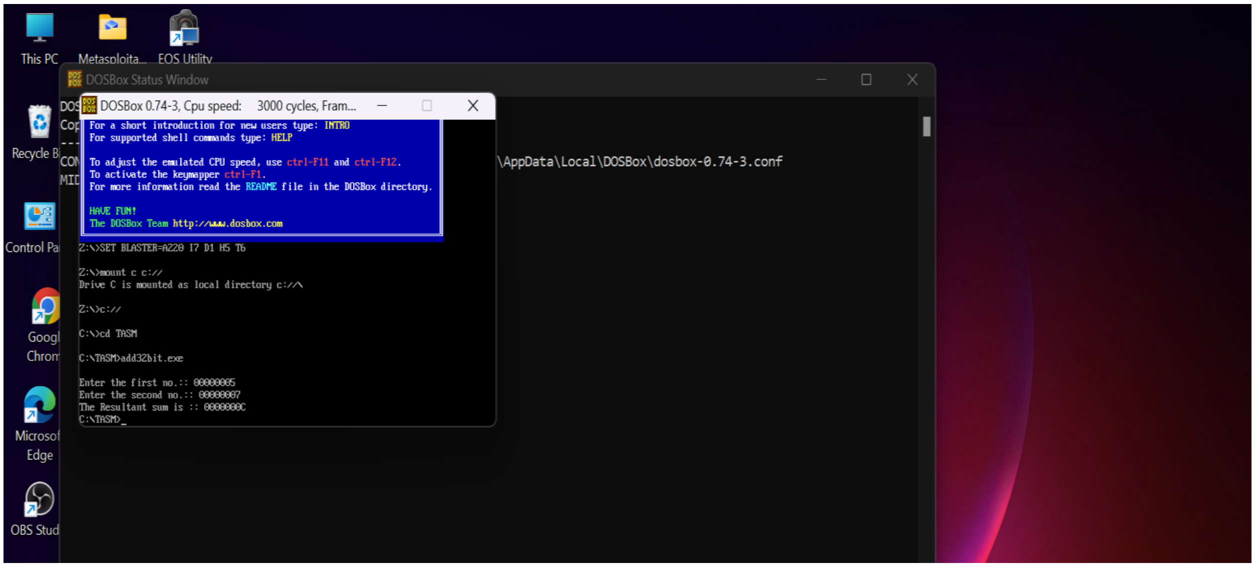
Code:

```
.model small
.stack 100H
.386
.data
data1 dd 00H
msg db 10,13,"Enter the first no.:: $"
msg1 db 10,13,"Enter the second no.:: $"
msg2 db 10,13,"The Resultant sum is :: $"
.code
.startup
MOV EBX, 00000000
MOV AH,09
MOV DX,OFFSET msg
INT 21H
MOV ECX, 8
AGAIN: MOV AH, 01
INT 21H
CMP AL, 'A'
JGE P1
SUB AL,30H
JMP P4
P1: SUB AL, 37H
```

```
P4: SHL EBX, 4
MOV AH,00
ADD EBX, EAX
LOOP AGAIN
MOV data1, EBX
MOV AH,09
MOV DX,OFFSET msg1
INT 21H
MOV ECX, 8
AGAIN2: MOV AH, 01
INT 21H
CMP AL, 'A'
JGE P2
SUB AL,30H
JMP P3
P2: SUB AL, 37H
P3: SHL EBX, 4
MOV AH,00
ADD EBX,EAX
LOOP AGAIN2
ADD EBX, data1
MOV AH,09
MOV DX,OFFSET msg2
INT 21H
MOV DX, 00
MOV ECX, 8
AGAIN3: ROL EBX, 4
MOV EDX,EBX
AND DX, 0FH
```

```
CMP DX, 09

JG L6

ADD DX,30H

JMP L7

L6: ADD DX, 37H

L7: MOV AH,02

INT 21H

LOOP AGAIN3

MOV AH, 4CH

INT 21H

end
```

Output:



```
For a short introduction for new users type: INTRO
For supported shell commands type: HELP

To adjust the emulated CPU speed, use ctrl-F11 and ctrl-F12.
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.

HAVE FUN!
The DOSBox Team http://www.dosbox.com
Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>mount c c://
Drive C is mounted as local directory c://\

Z:\>c://

C:\>cd TASM

C:\TASM>add32bit.exe

Enter the first no.:: 00000005
Enter the second no.:: 00000007
The Resultant sum is :: 0000000C
C:\TASM>_
```

```
\AppData\Local\DOSBox\dosbox-0.74-3.conf
```

Q2. Write a program for 32 bit binary subtraction

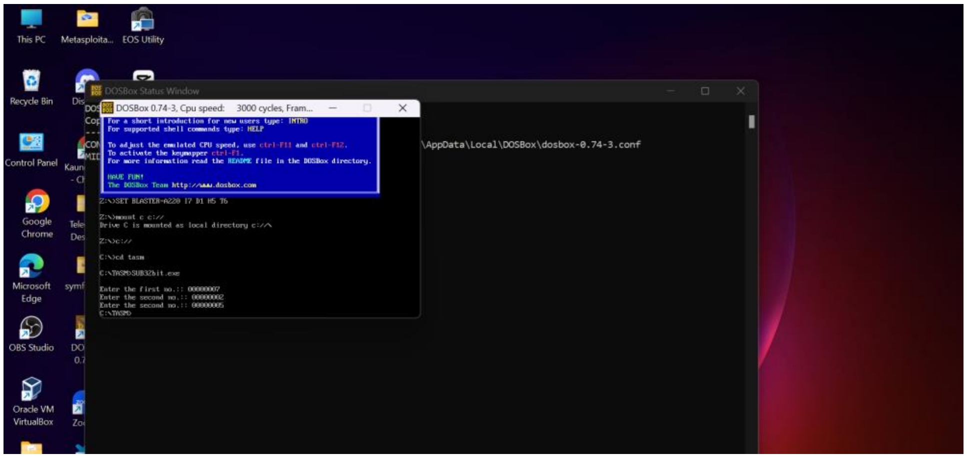
Code:

```
.model small
.stack 100h
.386
.data
data1 dd 00H

str1 db 10,13,"Enter the first no.:: $"
str2 db 10,13,"Enter the second no.:: $"
dif db 10,13,"The difference is: $"

.code
.startup
mov EBX,00000000

mov AH,09
mov DX,offset str1
int 21h

mov ECX,8
AGAIN:
MOV AH, 01
INT 21H
CMP AL, 'A'
JGE L5
SUB AL,30H
JMP L6
```

```
L5: SUB AL,37H

L6: SHL EBX,4

ADD BL,AL

LOOP AGAIN


MOV data1,EBX


mov AH,09

MOV DX,OFFSET str2

INT 21H


MOV ECX, 8

AGAIN2:

MOV AH, 01

INT 21H

CMP AL, 'A'

JGE L7

SUB AL,30H

JMP L8

L7: SUB AL,37H

L8: SHL EBX,4

ADD BL,AL

LOOP AGAIN2


SUB data1,EBX

MOV EBX,data1


mov AH,09

mov DX,offset str2
```

```
int 21h

MOV ECX, 8
AGAIN3: ROL EBX, 4
MOV EDX,EBX
AND DX, 0FH
CMP DX, 09
JG L9
ADD DL,30H
JMP L10
L9: ADD DL, 37H
L10: MOV AH,02
INT 21H
LOOP AGAIN3

  MOV AH,4CH
  INT 21H

End
```

Output:



```
\AppData\Local\DOSBox\dosbox-0.74-3.conf
```

```
For a short introduction for new users type: INTRO
For supported shell commands type: HELP

To adjust the emulated CPU speed, use ctrl-F11 and ctrl-F12.
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.

HAVE FUN!
The DOSBox Team http://www.dosbox.com
Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>mount c c://
Drive C is mounted as local directory c://\

Z:\>c://

C:\>cd tasm

C:\TASM>SUB32bit.exe

Enter the first no.:: 00000007
Enter the second no.:: 00000002
Enter the second no.:: 00000005
C:\TASM>
```

Q3. Write a program for 32 bit binary division

Code:

```
model small
.stack 100H
.386

.data
DATA1 dd 00000000H
DATA2 dd 00000000H
REM dd ?
QUO dd ?
msg db 10,13,"Enter the first no.:: $"
msg1 db 10,13,"Enter the second no.:: $"
msg2 db 10,13,"The Remainder is :: $"
msg3 db 10,13,"The Quotient is :: $"

.code
.startup
```

```
MOV AH,09
MOV DX,OFFSET msg
INT 21H

MOV EBX,0
MOV CX,8

AGAIN: MOV AH,01 ;1ST NO. ENTERED
INT 21H
CMP AL,'A'
JGE L5
JMP L6
L5: SUB AL,37H
L6: SUB AL,30H
SHL EBX,4
ADD BL,AL
LOOP AGAIN

MOV DATA1, EBX

MOV AH,09
MOV DX, OFFSET msg1
INT 21H

MOV EBX, 0
MOV CX, 8

AGAIN1: MOV AH,01 ;2nd NO. ENTERED
INT 21H
```

```
CMP AL,'A'

JGE L7

SUB AL, 30H

JMP L8

L7: SUB AL, 37H

L8: SHL EBX, 4

ADD BL,AL

LOOP AGAIN1


MOV DATA2, EBX

MOV EBX,0

MOV EDX,0

MOV EAX, 0

MOV EAX, DATA1

MOV EBX,DATA2

DIV EBX

MOV REM, EDX ;REM=REMAINDER

MOV QUO, EAX ;QUO=QUOTIENT


MOV AH,09

MOV DX,OFFSET msg2

INT 21H


MOV EBX, REM

MOV CX,8


AGAIN2: ROL EBX,4

MOV DL,BL

AND DL,0FH ; to o/p the result in rem
```

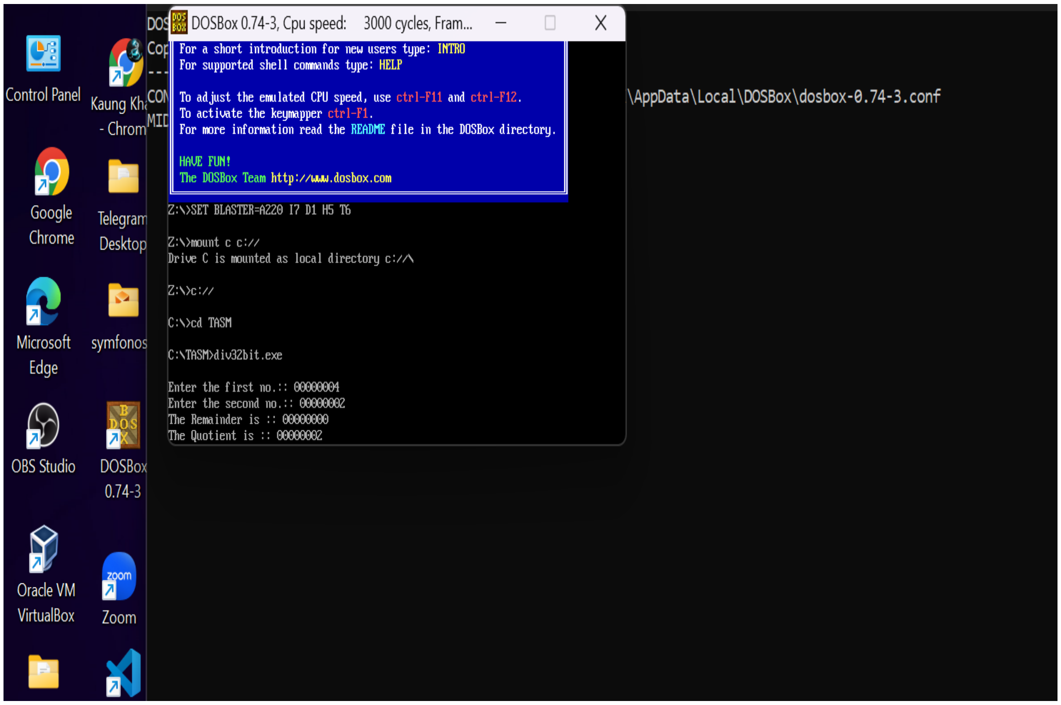
```
CMP DL,9

JBE L1

ADD DL,37H

MOV AH,02

INT 21H

JMP L2

L1: ADD DL,30H

MOV AH,02

INT 21H

L2: LOOP AGAIN2


MOV AH,09

MOV DX,OFFSET msg3

INT 21H


MOV EBX, QUO

MOV CX,8


AGAIN3: ROL EBX, 4

MOV DL,BL

AND DL,0FH ; to o/p the result in quo

CMP DL,9

JBE L3

ADD DL,37H

MOV AH,02

INT 21H

JMP L4

L3: ADD DL,30H

MOV AH,02
```

INT 21H

L4: LOOP AGAIN3

END

Output:



```
For a short introduction for new users type: INTRO
For supported shell commands type: HELP

To adjust the emulated CPU speed, use ctrl-F11 and ctrl-F12.
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.

HAVE FUN!
The DOSBox Team http://www.dosbox.com
Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>mount c c://
Drive C is mounted as local directory c://\

Z:\>c://

C:\>cd TASM

C:\TASM>div32bit.exe

Enter the first no.:: 00000004
Enter the second no.:: 00000002
The Remainder is :: 00000000
The Quotient is :: 00000002
```

```
\AppData\Local\DOSBox\dosbox-0.74-3.conf
```

Q4. Write a program for 32 bit binary multiplication

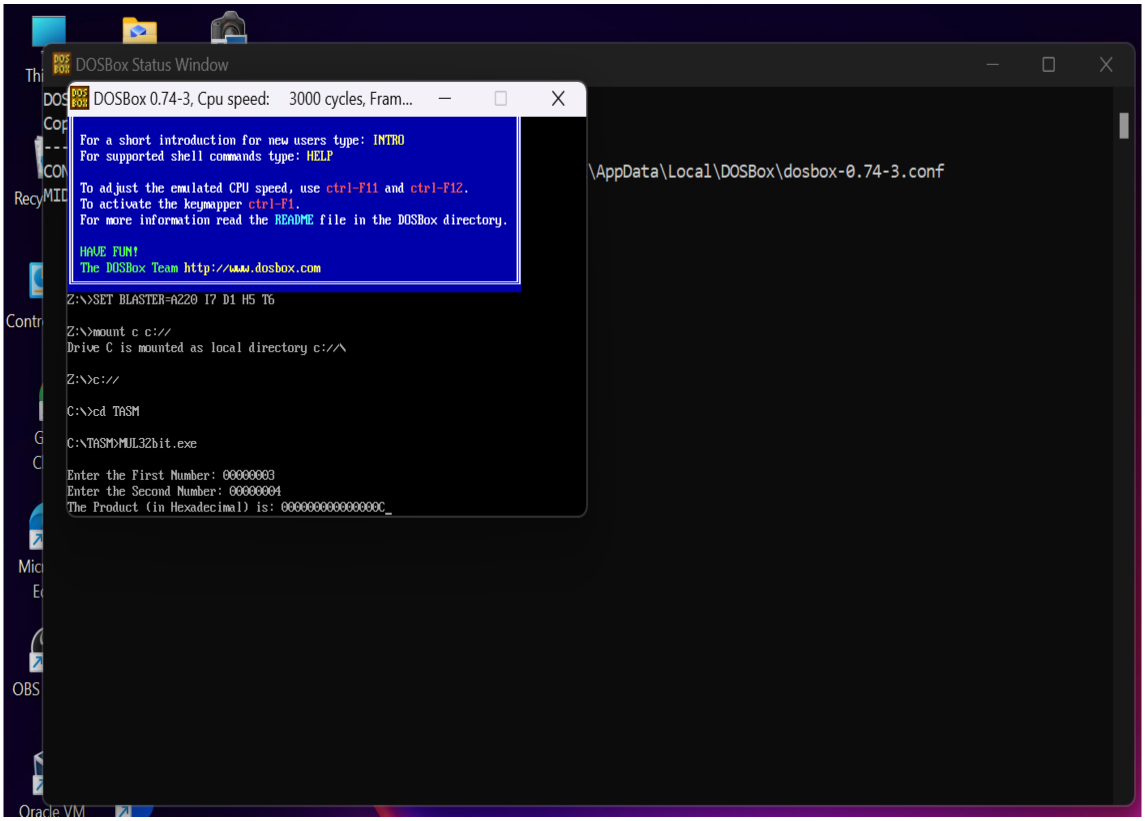
Code:

```
.model small

.stack 100H

.386


.data

DATA1 dd 00000000H

DATA2 dd 00000000H
```

```
PROD1 dd ?

PROD2 dd ?

msg db 10,13,"Enter the First Number: $"

msg1 db 10,13,"Enter the Second Number: $"

msg2 db 10,13,"The Product (in Hexadecimal) is: $"


.code

.startup

MOV AH,09

MOV DX,OFFSET msg

INT 21H


MOV EBX, 0

MOV CX, 8


AGAIN:

MOV AH, 01

INT 21H

CMP AL,'A'

JGE L5

SUB AL,30H

JMP L6

L5: SUB AL,37H

L6: SHL EBX,4

ADD BL,AL

LOOP AGAIN


MOV DATA1, EBX
MOV AH,09

MOV DX, OFFSET msg1

INT 21H


MOV EBX, 0

MOV CX, 8


AGAIN1:

MOV AH,01

INT 21H

CMP AL,'A'

JGE L7

SUB AL, 30H

JMP L8

L7: SUB AL, 37H

L8: SHL EBX, 4

ADD BL,AL

LOOP AGAIN1


MOV DATA2, EBX

MOV EBX,0

MOV EDX,0

MOV EAX,0

MOV EAX,DATA1

MOV EBX,DATA2

MUL EBX

MOV PROD1,EDX

MOV PROD2,EAX
MOV AH,09
MOV DX,OFFSET msg2
INT 21H


MOV EBX, PROD1
MOV CX, 8
AGAIN2:
        ROL EBX, 4
        MOV DL,BL
        AND DL, 0FH
        CMP DL, 9
        JBE L1
        ADD DL, 37H

        MOV AH, 02
        INT 21H
        JMP L2

        L1: ADD DL,30H
        MOV AH,02
        INT 21H
        L2: LOOP AGAIN2

MOV EBX, PROD2
MOV CX, 8
AGAIN3:
         ROL EBX, 4
         MOV DL,BL
```

```
    AND DL, 0FH
    CMP DL,9
    JBE L3
    ADD DL,37H
    MOV AH,02
    INT 21H
    JMP L4
    L3: ADD DL,30H
    MOV AH,02
    INT 21H
    L4: LOOP AGAIN3

END
```

Output:



```
\AppData\Local\DOSBox\dosbox-0.74-3.conf
```

```
For a short introduction for new users type: INTRO
For supported shell commands type: HELP

To adjust the emulated CPU speed, use ctrl-F11 and ctrl-F12.
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.

HAVE FUN!
The DOSBox Team http://www.dosbox.com
Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>mount c c://
Drive C is mounted as local directory c://\

Z:\>c://

C:\>cd TASM

C:\TASM>MUL32bit.exe

Enter the First Number: 00000003
Enter the Second Number: 00000004
The Product (in Hexadecimal) is: 000000000000000C_
```

Q5. Write a program for 32 bit BCD Addition?

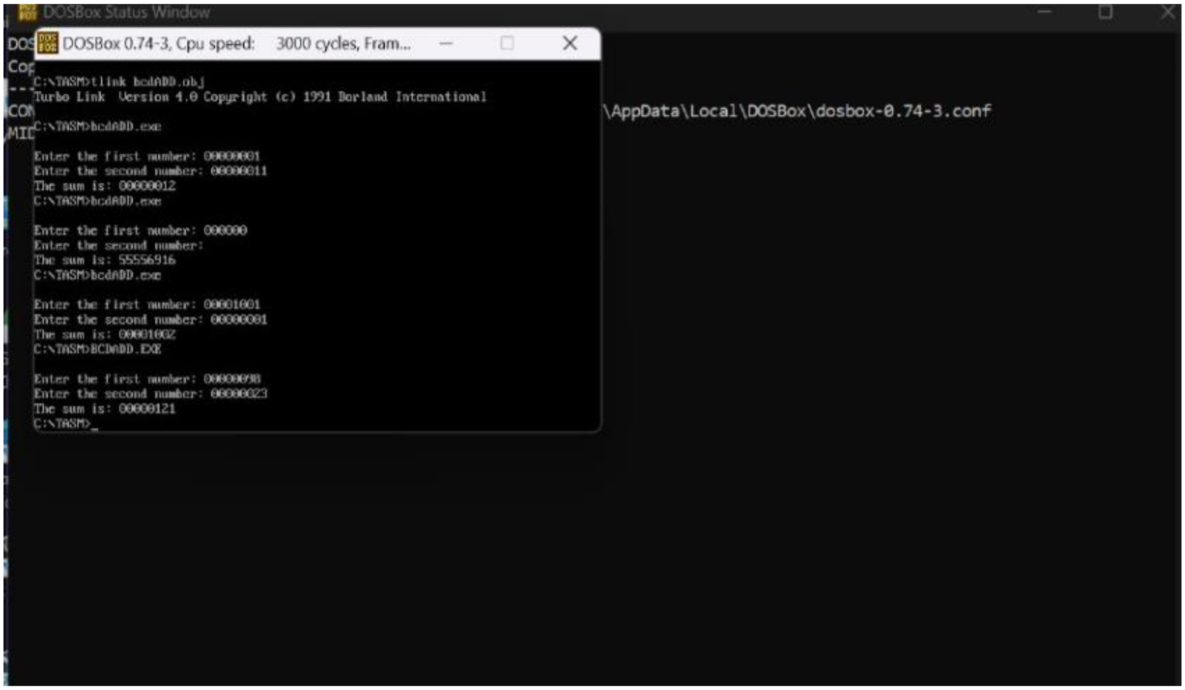
Code:

```
.model small
.stack 100h
.386
.data
num1 dd 00000000h
num2 dd 00000000h
num3 dd 00000000h
msg db 10,13,"Enter the first number: $"
msg1 db 10,13,"Enter the second number: $"
msg2 db 10,13,"The sum is: $"

.code
.startup
mov ah,09
mov dx,offset msg
int 21h

mov ebx,0
mov cx,8
again: mov ah,01
int 21h
cmp al,'A'
jge l2
sub al,30h
shl ebx,4
add bl,al
loop again
```

```
mov num1,ebx

mov ah,09
mov dx,offset msg1
int 21h

mov ebx,0
mov cx,8
again1: mov ah,01
int 21h
cmp al,'A'
jge l2
sub al,30h
shl ebx,4
add bl,al
loop again1

mov num2,ebx

mov ax,word ptr num1
mov dx,word ptr num2
add al,dl
daa
mov bl,al
mov al,ah
adc al,dh
daa
mov bh,al
mov word ptr num3,bx

mov ax,word ptr num1+2

mov dx,word ptr num2+2

adc al,dl

daa

mov bl,al

mov al,ah

adc al,dh

daa

mov bh,al

mov word ptr num3+2,bx

mov ebx,num3


mov ah,09h

mov dx,offset msg2

int 21h


jnc l6

mov ah,02h

mov dl,"1"

int 21h

l6:mov cx,8

again2:rol ebx,4

mov dl,bl

and dl,0Fh

add dl,30h

mov ah,02

int 21h

loop again2
```

```
l2:mov ah,4ch

int 21h

End
```

Result



DOSBox Status Window
DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...
\AppData\Local\DOSBox\dosbox-0.74-3.conf

```
C:\TASM>tlink bcdADD.obj
Turbo Link  Version 4.0 Copyright (c) 1991 Borland International

C:\TASM>bcdADD.exe

Enter the first number: 00000001
Enter the second number: 00000011
The sum is: 00000012
C:\TASM>bcdADD.exe

Enter the first number: 000000
Enter the second number:
The sum is: 55556916
C:\TASM>bcdADD.exe

Enter the first number: 00001001
Enter the second number: 00000001
The sum is: 00001002
C:\TASM>BCDADD.EXE

Enter the first number: 00000098
Enter the second number: 00000023
The sum is: 00000121
C:\TASM>_
```

Q6.Write a program for 32-bit BCD subtraction?

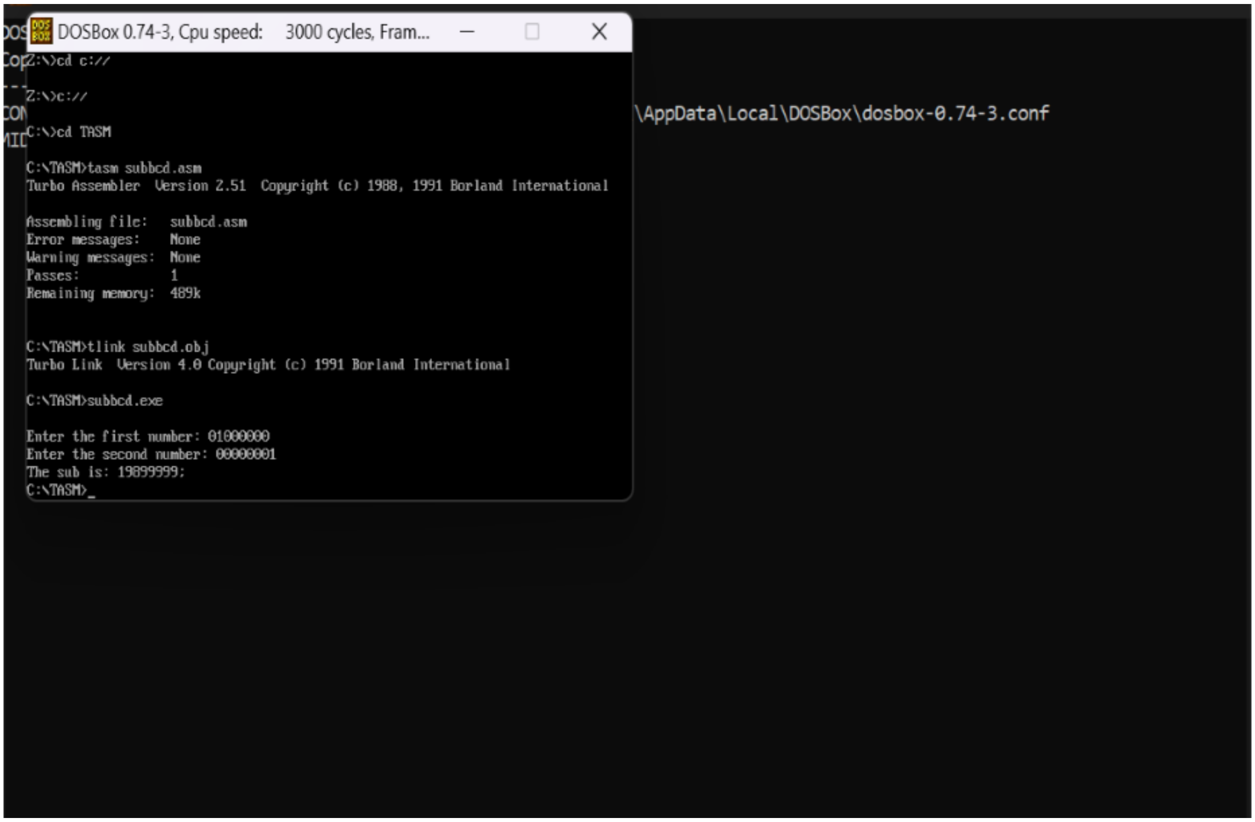
Code:

```
.model small

.stack 100h

.386

.data

num1 dd 00000000h

num2 dd 00000000h

num3 dd 00000000h

msg db 10,13,"Enter the first number: $"

msg1 db 10,13,"Enter the second number: $"
```

```
msg2 db 10,13,"The sum is: $"

.code

.startup

mov ah,09

mov dx,offset msg

int 21h


mov ebx,0

mov cx,8

again: mov ah,01

int 21h

cmp al,'A'

jge l2

sub al,30h

shl ebx,4

sub bl,al

loop again


mov num1,ebx


mov ah,09

mov dx,offset msg1

int 21h


mov ebx,0

mov cx,8

again1: mov ah,01

int 21h
```

```
cmp al,'A'

jge l2

sub al,30h

shl ebx,4

sub bl,al

loop again1


mov num2,ebx


mov ax,word ptr num1

mov dx,word ptr num2

sub al,dl

das

mov bl,al

mov al,ah

sbb al,dh

das

mov bh,al

mov word ptr num3,bx

mov ax,word ptr num1+2

mov dx,word ptr num2+2

sbb al,dl

das

mov bl,al

mov al,ah

sbb al,dh

das

mov bh,al

mov word ptr num3+2,bx
mov ebx,num3

mov ah,09h
mov dx,offset msg2
int 21h

jnc l6
mov ah,02h
mov dl,"1"
int 21h
l6:
mov cx,8
again2:
rol ebx,4
mov dl,bl
and dl,0Fh
add dl,30h
mov ah,02
int 21h
loop again2

l2:mov ah,4ch
int 21h
End
```

Result:


DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...
\AppData\Local\DOSBox\dosbox-0.74-3.conf

```
Z:\>cd c://

Z:\>c://

C:\>cd TASM

C:\TASM>tasm subbcd.asm
Turbo Assembler  Version 2.51  Copyright (c) 1988, 1991 Borland International

Assembling file:   subbcd.asm
Error messages:    None
Warning messages:  None
Passes:            1
Remaining memory:  489k


C:\TASM>tlink subbcd.obj
Turbo Link  Version 4.0 Copyright (c) 1991 Borland International

C:\TASM>subbcd.exe

Enter the first number: 01000000
Enter the second number: 00000001
The sub is: 19899999;
C:\TASM>_
```

Q7 .Write a program for Sorting.

1.Sorting descending order

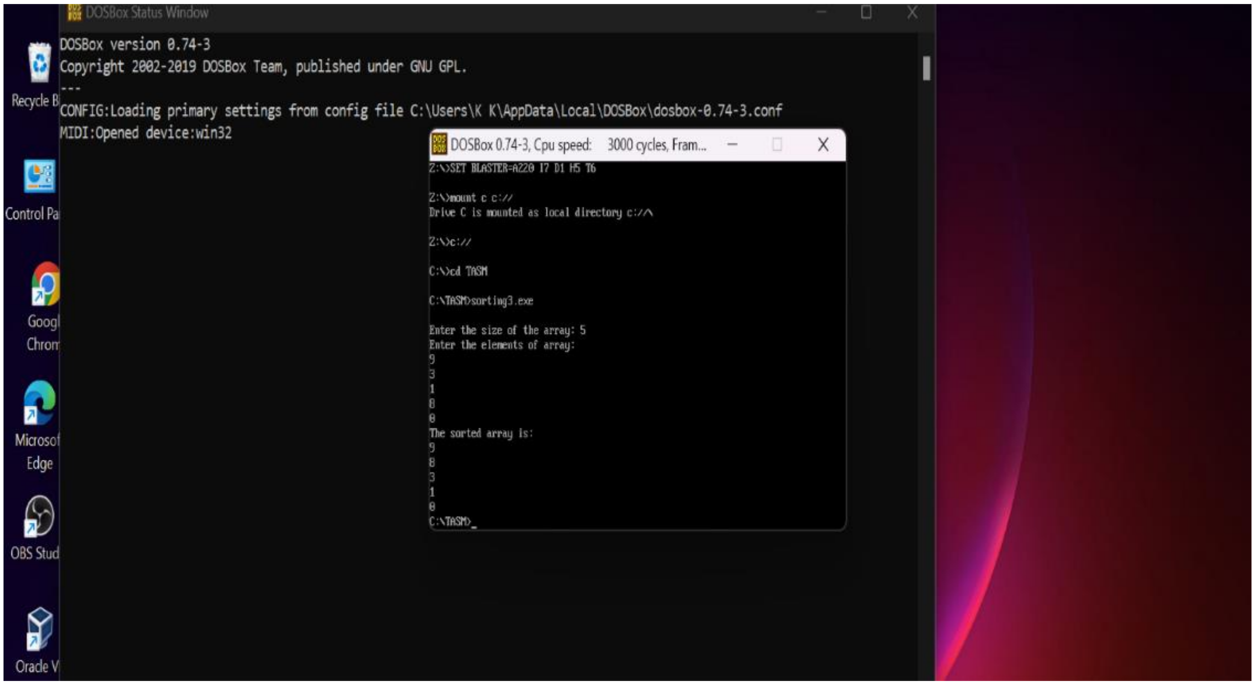
Code:

```
.model SMALL
.stack 100H
.386

.data
ARRAY dw 20 DUP (?)
DATA1 dw 0000H
NUMB dw 0000H
msg db 10,13,"Enter the size of the array: $"
msg2 db 10,13,"Enter the elements of array: $"
msg3 db 10,13,"The sorted array is:$"
.code
.startup

MOV AH, 09H
MOV DX, OFFSET msg
INT 21H
MOV AH, 01H
INT 21H

SUB AL, 30H
MOV AH, 0
MOV CX, AX
MOV DATA1, AX

MOV AH,09H
MOV DX, OFFSET msg2
INT 21H
MOV AH, 0
MOV SI, 0

MOV BX, OFFSET ARRAY

L1:     MOV DL, 0AH
MOV AH, 02H
INT 21H

MOV DX, SI
MOV AH, 01H
INT 21H

SUB AL,30H
MOV SI, DX
MOV [BX + SI], AX

INC SI
LOOP L1

MOV CX, DATA1
MOV BX, OFFSET ARRAY
MOV DI,CX
```

```
L2:     ; MOV CX, DATA1
MOV NUMB, CX
DEC NUMB

MOV CX, NUMB
MOV SI, 0

L3:     MOV AL, [BX + SI]
CMP [BX + SI + 1],AL
JL L4

L4:     INC SI
LOOP L3

DEC DI
JNZ L2

MOV CX, DATA1
MOV SI, 0
MOV BX, OFFSET ARRAY

MOV AH, 09
    MOV DX, OFFSET msg3
INT 21H

    L5:         MOV DL, 0AH ; jump onto next line
MOV AH, 02H
INT 21H
```

```
MOV DX, [BX + SI]
INC SI
ADD DL, 30H

        MOV AH, 02
INT 21H

      LOOP L5

; exit from the program
MOV AH, 4CH
    INT 21H

END
```

Result:



```
DOSBox version 0.74-3
Copyright 2002-2019 DOSBox Team, published under GNU GPL.
---
CONFIG:Loading primary settings from config file C:\Users\K K\AppData\Local\DOSBox\dosbox-0.74-3.conf
MIDI:Opened device:win32
```

```
Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>mount c c://
Drive C is mounted as local directory c://\

Z:\>c://

C:\>cd TASM

C:\TASM>sorting3.exe

Enter the size of the array: 5
Enter the elements of array:
9
3
1
8
0
The sorted array is:
9
8
3
1
0
C:\TASM>_
```

2.Sorting ascending order

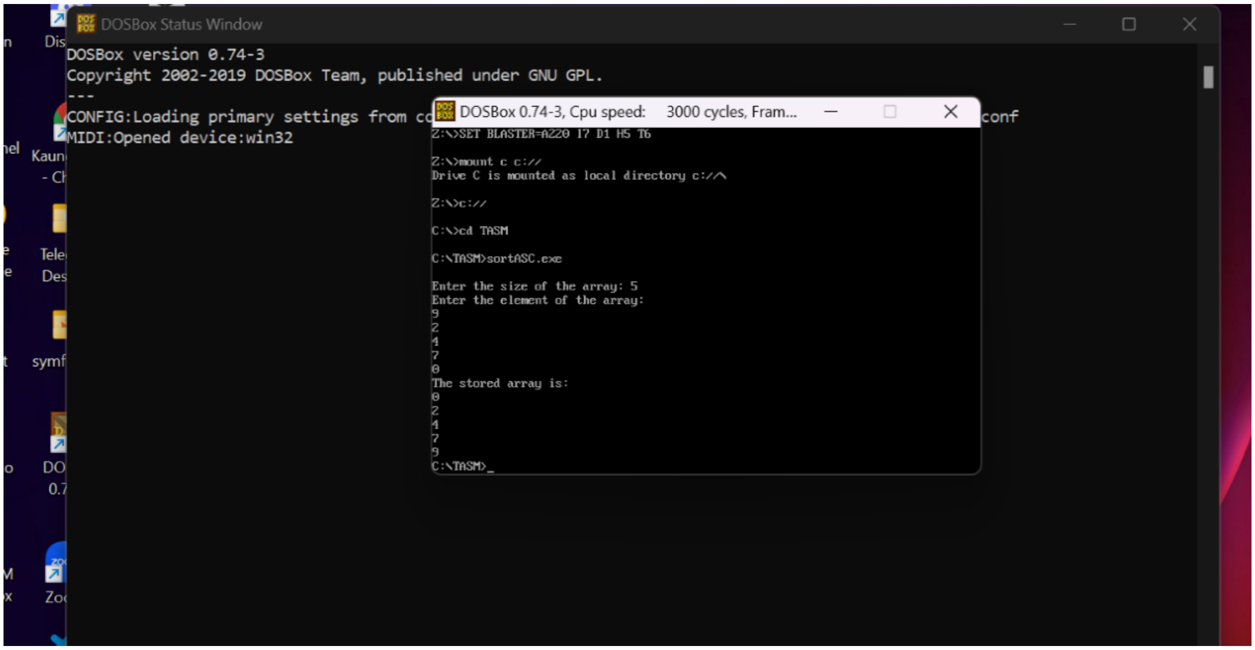
Code:

```
.model small

.stack 100H

.386

.data

ARRAY dw 20 DUP(?)

DATA1 dw 0000H

NUMB dw 0000H

msg db 10,13,"Enter the size of the array: $"

msg2 db 10,13,"Enter the element of the array: $"

msg3 db 10,13,"The stored array is: $"ta

.code

.startup

mov ah,09h

mov dx,offset msg

int 21h


mov ah,01h

int 21h


sub al,30h

mov ah,0

mov cx,ax
mov DATA1 ,ax

mov ah,09h
mov dx,offset msg2
int 21h

mov ah,0
mov si,0

mov bx,offset ARRAY


L1: mov dl,0AH
mov ah,02h
int 21h

mov dx,si
mov ah,01h
int 21h

sub al,30h

mov si,dx
mov [bx+si],ax

INC si
LOOP L1
```

```
mov cx,DATA1

mov bx,offset ARRAY

mov di,cx

L2: mov cx,DATA1
mov NUMB,cx
DEC NUMB
mov cx,NUMB
mov si,0

L3: mov al,[bx+si]
cmp [bx+si+1],al
jl L4
xchg al,[bx+si+1]
mov [bx+si],al
L4: inc si
LOOP L3

DEC DI

JNZ L2

MOV CX, DATA1
MOV SI, DATA1
dec SI
MOV BX, OFFSET ARRAY
```

```
   MOV AH,09
MOV DX, OFFSET msg3
INT 21H

   L5:     MOV DL, 0AH
MOV AH, 02H
INT 21H

MOV DX, [BX + SI]
DEC si
ADD DL, 30H

          MOV AH, 02
INT 21H

            LOOP L5

MOV AH, 4CH
   INT 21H
end
```

Result:



```
DOSBox version 0.74-3
Copyright 2002-2019 DOSBox Team, published under GNU GPL.
---
CONFIG:Loading primary settings from co                                              conf
MIDI:Opened device:win32
```

```
Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>mount c c://
Drive C is mounted as local directory c://\

Z:\>c://

C:\>cd TASM

C:\TASM>sortASC.exe

Enter the size of the array: 5
Enter the element of the array:
9
2
4
7
0
The stored array is:
0
2
4
7
9
C:\TASM>_
```

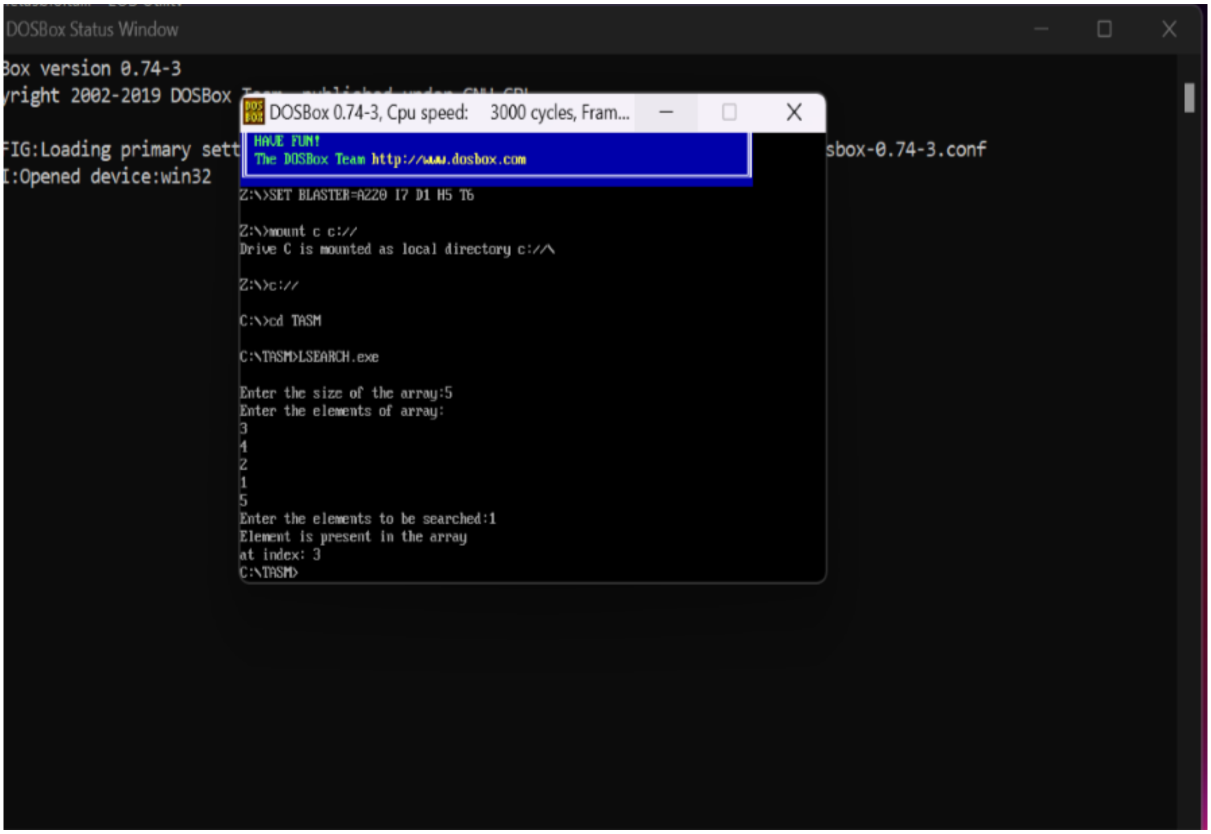
Q8. Write a program for linear search?

Code:

```
.model small
.stack 100H
.386
.data

ARRAY dw 20 DUP (?)
DATA1 dw 0000H
success db 10,13,"Element is present in the array $"
fail db 10,13,"Element is not present in array $"

msg db 10,13, "Enter the size of the array:$"
msg2 db 10,13,"Enter the elements of array: $"
msg3 db 10,13, "Enter the elements to be searched:$"
msg4 db 10,13,"at index: $"

.code
.startup
MOV AH,09
MOV DX,OFFSET msg
INT 21H

MOV AH,01
INT 21H
SUB AL,30H
MOV AH,0
MOV CX,AX

MOV DATA1,AX


MOV AH,09
MOV DX,OFFSET msg2
INT 21H

MOV AH,0
MOV SI,0
MOV BX,OFFSET ARRAY
L1:     MOV DL,0AH
MOV AH,02H
INT 21H

MOV DX,SI
MOV AH,01H
INT 21H

SUB AL,30H

MOV [BX+SI], AX
INC SI
LOOP L1

MOV CX,DATA1

MOV AH,09
MOV DX,OFFSET msg3
INT 21H

MOV AH,01
INT 21H
SUB AL,30H

MOV SI,0
MOV BX, OFFSET ARRAY

L2:	CMP [BX+SI], AL
JZ L3
INC SI
LOOP L2

MOV AH,09H
MOV DX,OFFSET fail
INT 21H

MOV AH, 4CH
INT 21H

L3:	MOV AH,09H
MOV DX,OFFSET success
INT 21H

MOV AH, 09H
MOV DX,OFFSET msg4
INT 21H
```

```
MOV DX,SI

ADD DX,30H

MOV AH,02

INT 21H


MOV AH,4CH

INT 21H

END
```

Result:



```
Box version 0.74-3
yright 2002-2019 DOSBox

FIG:Loading primary sett                              sbox-0.74-3.conf
I:Opened device:win32
```

```
HAVE FUN!
The DOSBox Team http://www.dosbox.com
Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>mount c c://
Drive C is mounted as local directory c://\

Z:\>c://

C:\>cd TASM

C:\TASM>LSEARCH.exe

Enter the size of the array:5
Enter the elements of array:
3
4
2
1
5
Enter the elements to be searched:1
Element is present in the array
at index: 3
C:\TASM>
```

Q9. Write a program for binary search?

Code:

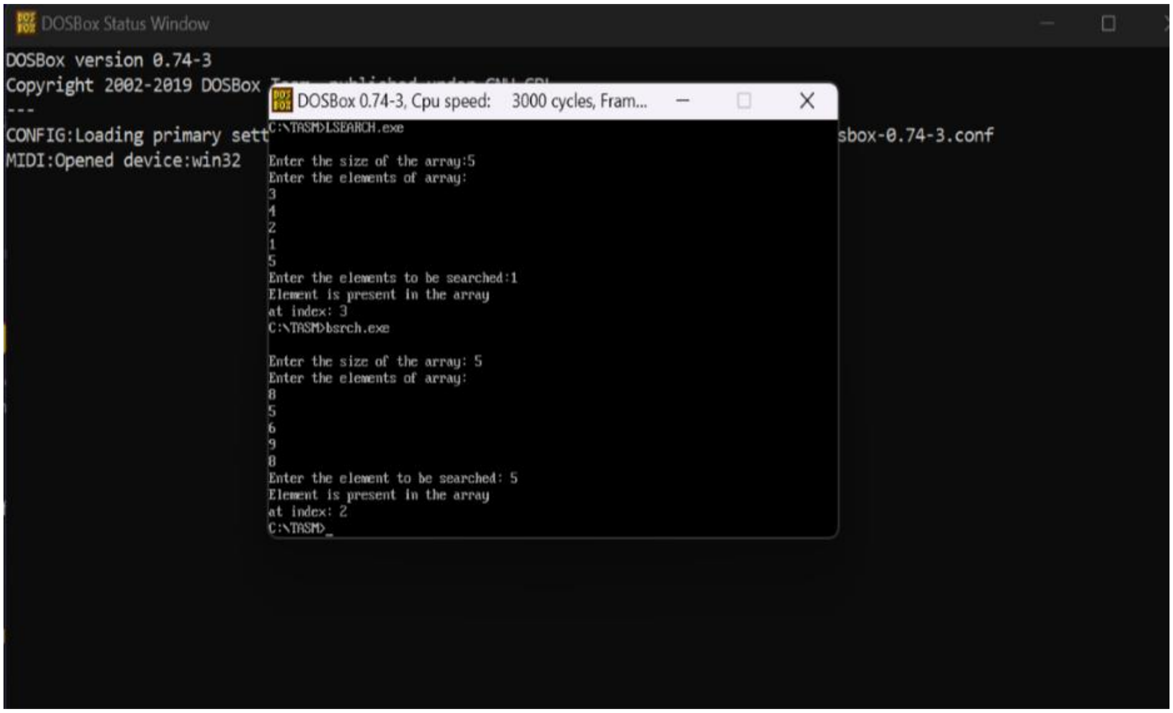
```
.model small
.stack 100H
.386
.data
ARRAY dw 20 DUP (?)
DATA1 dw 0000H
DATA2 dw 0000H

success db 10,13,"Element is present in the array $"
fail db 10,13,"Element is not present in the arary $"

msg db 10,13,"Enter the size of the array: $"
msg2 db 10,13,"Enter the elements of array: $"
msg3 db 10,13,"Enter the element to be searched: $"
msg4 db 10,13,"at index: $"
.code
.startup
;(1) for reading size of array

MOV AH,09
MOV DX,OFFSET msg
INT 21H

MOV AH,01
INT 21H
```

```
SUB AL,30H
MOV AH,0

MOV CX,AX

MOV DATA1,AX              ;DATA1 and CX contains size of Array entered by user



;(2) for reading elements in the array


MOV AH,09
MOV DX,OFFSET msg2
INT 21H


MOV AH,0
MOV SI,0
MOV BX, OFFSET ARRAY
L1:     MOV DL, 0AH    ;jump onto next line
MOV AH, 02H
INT 21H


MOV DX, SI       ;input element of the array


MOV AH, 01H
INT 21H


SUB AL,30H


MOV [BX + SI], AX
```

```
INC SI
LOOP L1

; (3) for reading element to be searched

MOV AH,09
MOV DX,OFFSET msg3
INT 21H

MOV AH,01                   ;Enter element to be searched
INT 21H
SUB AL,30H

; (4) for performing the Binary search and displaying the appropriate output
MOV DATA2,AX
MOV CX,DATA1
MOV SI,0
MOV DI, DATA1
MOV BP, 0
MOV BX, OFFSET ARRAY
MOV AX, DATA1
L2:        MOV SI, DI
ADD SI, BP
MOV AX, SI
MOV DL, 2
DIV DL
MOV AH,0
MOV DX,0
MOV SI,AX
MOV DX,DATA2
CMP [BX + SI],DL
JZ L3
CALL L4
LOOP L2

MOV AH, 09H
MOV DX,OFFSET fail ; if the element is not found
INT 21H

MOV AH, 4CH ; to forcefully terminate the program
INT 21H

L3:        MOV AH, 09H
MOV DX,OFFSET success ; if the element is found
INT 21H

MOV AH, 09H
MOV DX,OFFSET msg4
INT 21H

MOV DX,SI
ADD DX,30H
ADD DX,01

MOV AH, 02
INT 21H

MOV AH, 4CH
```

```
INT 21H

L4      PROC NEAR
CMP [BX+SI], DL
JL L6
MOV DI, SI
RET
L6:     MOV BP,SI
RET
L4      ENDP

MOV AH, 4CH
INT 21H
END
```

Result:



```
DOSBox version 0.74-3
Copyright 2002-2019 DOSBox
---
CONFIG:Loading primary sett                                    sbox-0.74-3.conf
MIDI:Opened device:win32
```

```
C:\TASM>LSEARCH.exe

Enter the size of the array:5
Enter the elements of array:
3
4
2
1
5
Enter the elements to be searched:1
Element is present in the array
at index: 3
C:\TASM>bsrch.exe

Enter the size of the array: 5
Enter the elements of array:
8
5
6
9
8
Enter the element to be searched: 5
Element is present in the array
at index: 2
C:\TASM>_
```

Q10. Write a program to add two arrays

Code:

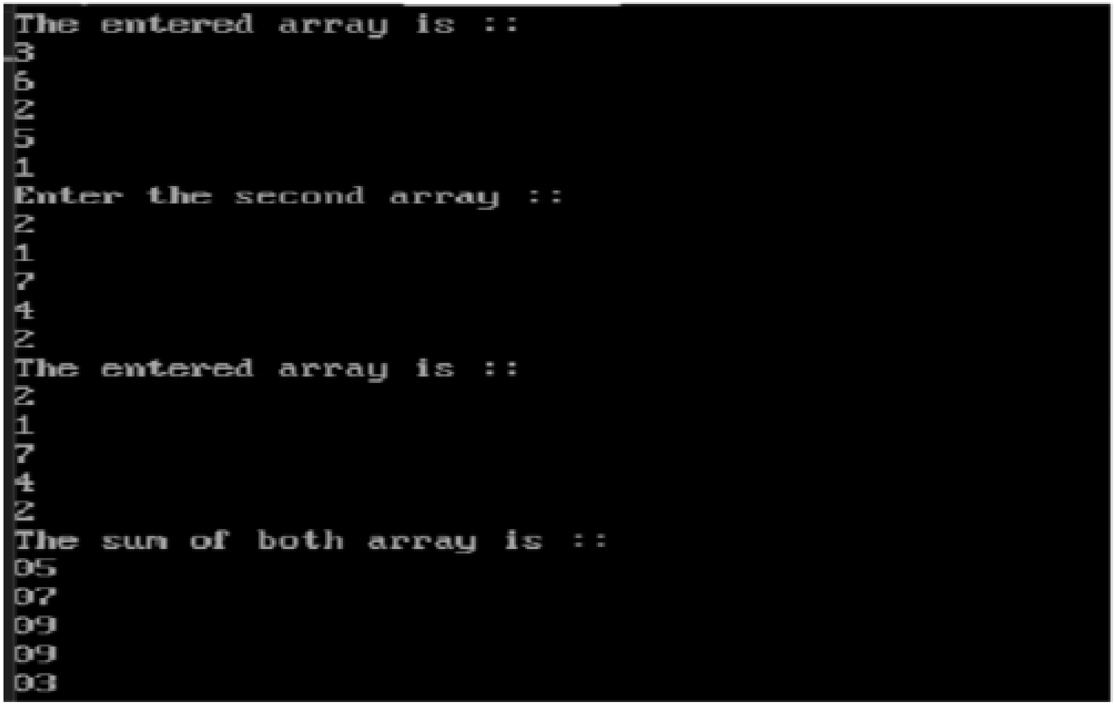
```
.model small

.386

.data

A1 DB 20 DUP (?)

A2 DB 20 DUP (?)

DATA1 dw 0000H

DATA2 DW 0000H

msg db 10,13,"Enter the size of the arrays :: $"

msg2 db 10,13,"Enter the first array :: $"

msg3 db 10,13,"The entered array is :: $"

msg4 db 10,13, "Enter the second array ::$"

msg5 db 10,13, "The sum of both array is ::$"

.code

.startup

MOV AH,09

MOV DX,OFFSET msg

INT 21H

MOV CX, 2

L4: MOV AH,01

INT 21H

CMP AL,'A'

JGE L9

SUB AL,30H

JMP L8

L9: SUB AL,37H

L8: SHL BX, 4

ADD BL, AL
```

```
LOOP L4

MOV AL, BL

MOV CL, AL

MOV AH, 0

MOV DATA1, AX

MOV CX, DATA1

MOV AH,09

MOV DX,OFFSET msg2

INT 21H


MOV AH,0

MOV CX, DATA1

LEA SI, A1

L1: MOV DL, 0AH ; jump onto next line

MOV AH, 02H

INT 21H

MOV AH, 01H

INT 21H

SUB AL,30H

MOV [SI], AL

INC SI

LOOP L1

MOV AH, 09H

MOV DX, OFFSET MSG3

INT 21H

MOV CX, DATA1

LEA SI, A1

L2:mov ah, 02h

mov dl, 0ah
```

```
int 21h
MOV DL, [SI]
ADD DL, 30h
MOV AH, 02
INT 21H
INC SI
LOOP L2
MOV CX, DATA1
MOV AH,09
MOV DX,OFFSET msg4
INT 21H
MOV AH,0
LEA DI, A2
L3: MOV DL, 0AH ; jump onto next line
MOV AH, 02H
INT 21H
MOV AH, 01H
INT 21H
SUB AL,30H
MOV [DI], AL
INC DI
LOOP L3
MOV AH, 09H
MOV DX, OFFSET MSG3
INT 21H

MOV CX, DATA1
LEA DI, A2
L14:mov ah, 02h
```

```
mov dl, 0ah

int 21h

mov dl, 0dh

int 21h

MOV DX, [DI]

ADD DL, 30h

MOV AH, 02

INT 21H

INC DI

LOOP L14

LEA SI, A1

LEA DI, A2

MOV CX, DATA1

ADDA: MOV AL, [SI]

ADD AL, [DI]

MOV [SI], AL

INC DI

INC SI

LOOP ADDA

MOV AH, 09H

MOV DX, OFFSET MSG5

INT 21H

MOV CX, DATA1

LEA SI, A1

L5:mov ah, 02h

mov dl, 0ah

int 21h

MOV DATA2, CX

MOV CX, 2
MOV BL, [SI]
ADDA1: ROL BL, 4
MOV DL, BL
AND DL, 0FH
CMP Dl, 9
JA L6
ADD DL, 30h
JMP L7
L6: ADD DL, 37H
L7: MOV AH, 02
INT 21H
LOOP ADDA1
MOV CX, DATA2
INC SI
LOOP L5

.EXIT
END
```

Resuilt:



```
The entered array is ::
3
6
2
5
1
Enter the second array ::
2
1
7
4
2
The entered array is ::
2
1
7
4
2
The sum of both array is ::
05
07
09
09
03
```

Q11. Write a program to subtract two arrays

Code:

```
.model small
.386
.data
A1 DB 20 DUP (?)
A2 DB 20 DUP (?)
DATA1 dw 0000H
DATA2 DW 0000H
msg db 10,13,"Enter the size of the arrays :: $"
msg2 db 10,13,"Enter the first array :: $"
msg3 db 10,13,"The entered array is :: $"
msg4 db 10,13, "Enter the second array ::$"
msg5 db 10,13, "The difference of both array is ::$"
```

```
.code

.startup

;Display msg

MOV AH,09

MOV DX,OFFSET msg

INT 21H

;Input the size of the array in DATA1

MOV CX, 2

L4: MOV AH,01

INT 21H

CMP AL,'A'

JGE L9

SUB AL,30H

JMP L8

L9: SUB AL,37H

L8: SHL BX, 4

ADD BL, AL

LOOP L4

MOV AL, BL

MOV CL, AL

MOV AH, 0

MOV DATA1, AX

MOV CX, DATA1

;Display msg2

MOV AH,09

MOV DX,OFFSET msg2

INT 21H

MOV AH,0

;Input the first array
```

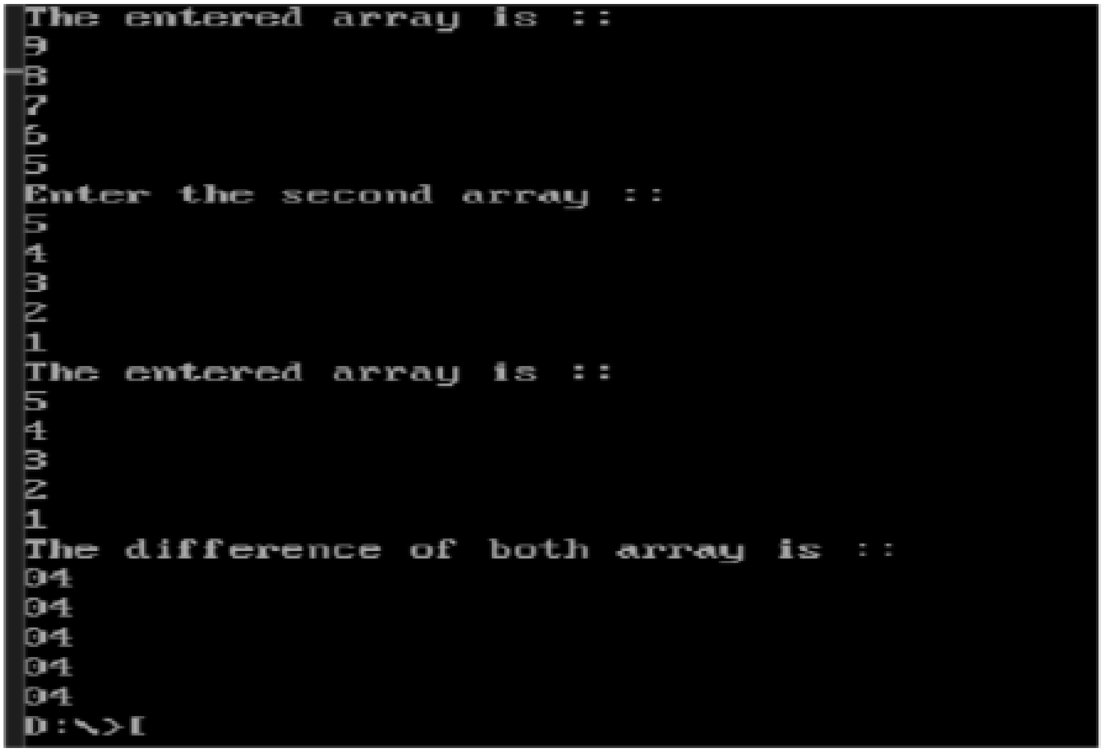
```
MOV CX, DATA1

LEA SI, A1

L1: MOV DL, 0AH ; jump onto next line

MOV AH, 02H

INT 21H

MOV AH, 01H

INT 21H

SUB AL,30H

MOV [SI], AL

INC SI

LOOP L1

;Display msg3

MOV AH, 09H

MOV DX, OFFSET MSG3

INT 21H

;Display the first array

MOV CX, DATA1

LEA SI, A1

L2:mov ah, 02h

mov dl, 0ah

int 21h

MOV DL, [SI]

ADD DL, 30h

MOV AH, 02

INT 21H

INC SI

LOOP L2

MOV CX, DATA1

;Display msg4
```

```
MOV AH,09
MOV DX,OFFSET msg4
INT 21H
MOV AH,0
;Input the second array
LEA DI, A2
L3: MOV DL, 0AH ; jump onto next line
MOV AH, 02H
INT 21H
MOV AH, 01H
INT 21H
SUB AL,30H
MOV [DI], AL
INC DI
LOOP L3
;Display msg3
MOV AH, 09H
MOV DX, OFFSET MSG3
INT 21H
;Display the second array
MOV CX, DATA1
LEA DI, A2
L14:mov ah, 02h
mov dl, 0ah
int 21h
mov dl, 0dh
int 21h
MOV DX, [DI]
ADD DL, 30h
```

```
MOV AH, 02
INT 21H
INC DI
LOOP L14
;Subtraction
LEA SI, A1
LEA DI, A2
MOV CX, DATA1
ADDA: MOV AL, [SI]
SUB AL, [DI]
MOV [SI], AL
INC DI
INC SI
LOOP ADDA
MOV AH, 09H
MOV DX, OFFSET MSG5
INT 21H
MOV CX, DATA1
LEA SI, A1
L5:mov ah, 02h
mov dl, 0ah
int 21h
MOV DATA2, CX
MOV CX, 2
MOV BL, [SI]
ADDA1: ROL BL, 4
MOV DL, BL
AND DL, 0FH
CMP Dl, 9
```

```
JA L6
ADD DL, 30h
JMP L7
L6: ADD DL, 37H
L7: MOV AH, 02
INT 21H
LOOP ADDA1
MOV CX, DATA2
INC SI
LOOP L5
.EXIT
END
```

Result:



```
The entered array is ::
9
8
7
6
5
Enter the second array ::
5
4
3
2
1
The entered array is ::
5
4
3
2
1
The difference of both array is ::
04
04
04
04
04
D:\>[
```

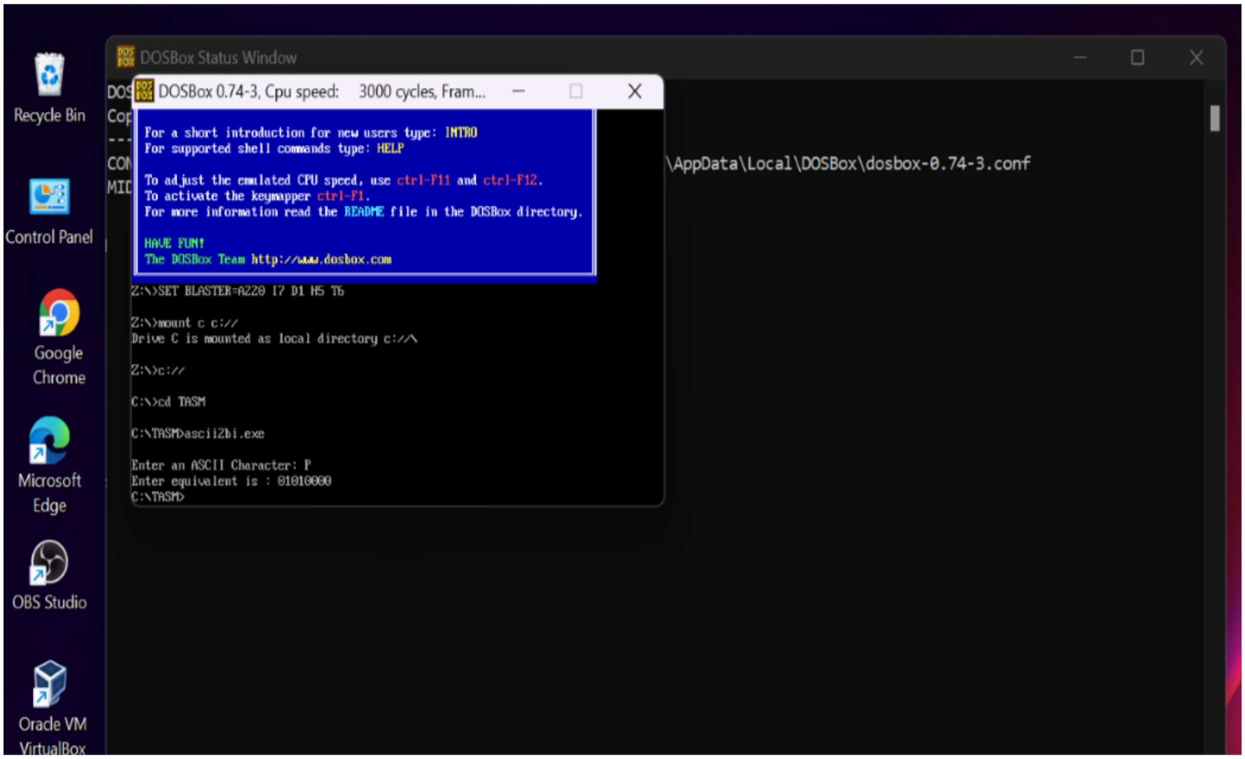
Q12. Write a program for ascii to binary conversion.

Code:

```
.model SMALL

.stack 100H


.data


intputStr db 10,13, 'Enter an ASCII Character: $'

outputStr db 10,13, 'Enter equivalent is : $'


.code

.startup


MOV DX, OFFSET intputStr

MOV AH, 09H

INT 21H


MOV AH, 01H

INT 21H


MOV BL,AL


MOV DX, OFFSET outputStr

MOV AH,09H

INT 21H

MOV CX, 8


repeat8Times:

SHL BL,1
```

```
JC printOne


MOV DL, 30H

JMP print


printOne:

MOV DL, 31H

print:

MOV AH, 02H

INT 21H

LOOP repeat8Times


MOV AH, 4CH

INT 21H

END
```

Result:



```
\AppData\Local\DOSBox\dosbox-0.74-3.conf
```

```
For a short introduction for new users type: INTRO
For supported shell commands type: HELP

To adjust the emulated CPU speed, use ctrl-F11 and ctrl-F12.
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.

HAVE FUN!
The DOSBox Team http://www.dosbox.com
Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>mount c c://
Drive C is mounted as local directory c://\

Z:\>c://

C:\>cd TASM

C:\TASM>ascii2bi.exe

Enter an ASCII Character: P
Enter equivalent is : 01010000
C:\TASM>
```