

## U19CS076 MIT ASSIGNMENT 11-12

---

1. Write a Program for data transfer using different addressing modes.

```
dosseg
```

```
model small
```

```
.8086
```

```
.data
```

```
n1 dw 03A0h
```

```
n2 dw ?
```

```
.code
```

```
mov ax,@data
```

```
mov ds,ax
```

```
mov ax,1005h ;immediate addressing
```

```
mov bx,n1 ;direct
```

```
mov cx,ax ;register
```

```
mov ax,[bx] ;register indirect
```

```
mov ax,[bx + di] ;base index mode
```

```
mov ax,4c00h
```

```
int 21h
```

```
end
```

```

C:\TASM>debug 11q1.exe
-u
076A:0000 B86B07      MOV     AX,076B
076A:0003 8ED8        MOV     DS,AX
076A:0005 B80510      MOV     AX,1005
076A:0008 8B1E0800    MOV     BX,[0008]
076A:000C 8BC8        MOV     CX,AX
076A:000E 8B07        MOV     AX,[BX]
076A:0010 8B01        MOV     AX,[BX+DI]
076A:0012 B8004C      MOV     AX,4C00
076A:0015 CD21      INT     21
076A:0017 00A00300    ADD     [BX+SI+0003],AH
076A:001B 0000        ADD     [BX+SI],AL
076A:001D 820E000000    OR     BYTE PTR [0000],00
-g
Program terminated normally

```

2. Write Program to move data from source to destination using indirect addressing mode (Block Move without overlap).

```

dosseg
model small
.8086
.data
x db 12h,24h,36h,48h,60h
cnt db 5
ans db ?

.code
mov ax,@data
mov ds,ax
mov cl,cnt
mov ch, 00h
mov si,0000h
next: mov al,x[si]
      mov ans[si],al
      inc si
      loop next
mov ah, 4ch
int 21h
end

```

```

-u
076A:0000 B86B07      MOV     AX,076B
076A:0003 8ED8        MOV     DS,AX
076A:0005 8A0E1300     MOV     CL,[0013]
076A:0009 B500        MOV     CH,00
076A:000B BE0000     MOV     SI,0000
076A:000E 8A840E00     MOV     AL,[SI+000E]
076A:0012 88841400     MOV     [SI+0014],AL
076A:0016 46          INC     SI
076A:0017 E2F5        LOOP    000E
076A:0019 B44C        MOV     AH,4C
076A:001B CD21        INT     21
076A:001D 0012        ADD     [BP+SI],DL
076A:001F 2436        AND     AL,36
-g
Program terminated normally

```

```

-d 076b:0000
076B:0000 0E 00 88 84 14 00 46 E2-F5 B4 4C CD 21 00 12 24 .....F...L.?!..$
076B:0010 36 48 60 05 12 24 36 48-60 82 0E 00 00 00 83 6H'..$6H'.....
076B:0020 0E FF FF FF FF FF FF FF-FF FF FF FF FF FF FF .....
076B:0030 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF .....
076B:0040 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF .....
076B:0050 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF .....
076B:0060 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF .....
076B:0070 FF FF FF 26 00 FF FF FF-FF FF FF FF FF FF FF ...&.....

```

- Write a Program to move a block of data from source to destination (With overlap in either direction).

```

dosseg
model small
.8086
.data
y db 3 dup(0)
x db 10h,15h,20h,25h,50h
.code
mov ax, @data
mov ds, ax
lea si, x
lea di, y
mov cx, 0005h
up: mov al,[si]
mov [di], al
inc si

```

```

inc di
loop up
mov ax, 4c00h
int 21h
end

```

C:\TASM>debug 11q3.exe

```

-u
076A:0000 B8B07      MOV     AX,076B
076A:0003 8ED8        MOV     DS,AX
076A:0005 BE0F00     MOV     SI,000F
076A:0008 BF0C00     MOV     DI,000C
076A:000B B90500     MOV     CX,0005
076A:000E 8A04        MOV     AL,[SI]
076A:0010 8805        MOV     [DI],AL
076A:0012 46          INC     SI
076A:0013 47          INC     DI
076A:0014 E2F8        LOOP    000E
076A:0016 B8004C     MOV     AX,4C00
076A:0019 CD21        INT     21
076A:001B 0000        ADD     [BX+SI],AL
076A:001D 0000        ADD     [BX+SI],AL
076A:001F 1015        ADC     [DI],DL

```

-g

Program terminated normally

-d 076b:0000

```

076B:0000 88 05 46 47 E2 F8 B8 00-4C CD 21 00 10 15 20 25  ..FG....L.!... %
076B:0010 50 20 25 50 0E 10 00 02-00 82 0E 00 00 00 00 83  P %P.....
076B:0020 0E FF FF FF FF FF FF FF-FF FF FF FF FF FF FF  .....
076B:0030 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF  .....
076B:0040 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF  .....
076B:0050 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF  .....
076B:0060 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF  .....
076B:0070 FF FF FF 26 00 FF FF FF-FF FF FF FF FF FF FF  ...&.....

```

-q

- Write a Program to interchange two blocks of data.

dosseg

model small

.8086

.data

src db 20h,30h,10h,40h,50h

dst db 08,09,06,07,0Ah

total equ 05

```

.code
mov ax, @data
mov ds, ax
lea si,src
lea di,dst
mov cl, total

up: mov al, [si]
    mov bl, [di]
    mov [si], bl
    mov [di], al
    inc si
    inc di
    loop up
    mov ax,4c00h
    int 21h
end

```

```

C:\TASM>debug 11q4.exe
-u
076A:0000 B86B07      MOV     AX,076B
076A:0003 8ED8          MOV     DS,AX
076A:0005 BE0E00      MOV     SI,000E
076A:0008 BF1300      MOV     DI,0013
076A:000B B105          MOV     CL,05
076A:000D 8A04          MOV     AL,[SI]
076A:000F 8A1D          MOV     BL,[DI]
076A:0011 881C          MOV     [SI],BL
076A:0013 8805          MOV     [DI],AL
076A:0015 46          INC     SI
076A:0016 47          INC     DI
076A:0017 E2F4          LOOP    000D
076A:0019 B8004C      MOV     AX,4C00
076A:001C CD21          INT     21
076A:001E 2030          AND     [BX+SI],DH
-g
Program terminated normally

```

```

Program terminated normally
-d 076b:0000
076B:0000  1D 88 1C 88 05 46 47 E2-F4 B8 00 4C CD 21 08 09  ....FG....L.!...
076B:0010  06 07 0A 20 30 10 40 50-00 82 0E 00 00 00 00 83  ... 0.@P.....
076B:0020  0E FF FF FF FF FF FF FF-FF FF FF FF FF FF FF  .....
076B:0030  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF  .....
076B:0040  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF  .....
076B:0050  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF  .....
076B:0060  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF  .....
076B:0070  FF FF FF 26 00 FF FF FF-FF FF FF FF FF FF FF  ...&.....

```

## Assignment -12

\*For first three programs perform only one operation i.e. ADD or SUB.

1. Write ALP to ADD/SUB 'n' 16 bit numbers stored in consecutive memory location

Code:

dosseg

model small

.8086

.data

n db 0ah

a dw 1210h,3443h,0ab54h,4337h,5675h,0086h,4209h,0cd2h,12ffh,4500h

ans dw ?

carry db ?

.code

mov ax,@data

mov ds,ax

mov cl,n

mov ch,00h

mov si,offset a

mov ax,0h

mov dl,00

sumation: add ax,[si]

jnc next

```

inc dl
next: inc si
inc si
loop sumation

```

```

mov ans,ax
mov carry,dl
mov ax,4c00h
int 21h
end

```

C:\TASM>debug 12q1.exe

-u

```

076A:0000 B86C07      MOV     AX,076C
076A:0003 8ED8          MOV     DS,AX
076A:0005 8A0E0A00      MOV     CL,[000A]
076A:0009 B500          MOV     CH,00
076A:000B BE0B00      MOV     SI,000B
076A:000E B80000      MOV     AX,0000
076A:0011 B200          MOV     DL,00
076A:0013 0304          ADD     AX,[SI]
076A:0015 7302          JNB     0019
076A:0017 FEC2          INC     DL
076A:0019 46            INC     SI
076A:001A 46            INC     SI
076A:001B E2F6          LOOP    0013
076A:001D A31F00      MOV     [001F],AX

```

-g

Program terminated normally

Program terminated normally

-d 076c:0000

```

076C:0000 88 16 21 00 B8 00 4C CD-21 00 0A 10 12 43 34 54  ..!...L.!....C4T
076C:0010 AB 37 43 75 56 86 00 09-42 D2 0C FF 12 00 45 B3  .7CuU...B.....E.
076C:0020 32 02 FF FF FF FF FF FF-FF FF FF FF FF FF FF  2.....
076C:0030 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF  .....
076C:0040 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF  .....
076C:0050 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF  .....
076C:0060 FF FF FF 26 00 FF FF FF-FF FF FF FF FF FF FF  ...&.....
076C:0070 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF  .....

```

Sum= 02 32 B3 h

2. Write a Program to find smallest/largest number in a given array of 16 bits numbers.

```
dosseg
model small
.8086

.data
n db 0ah
a dw 0111h,0213h,1203h,1404h,5405h,4506h,9007h,6708h,6709h,560Ah
ans dw ?

.code
mov ax,@data
mov ds,ax

mov cl,n
mov ch,00h
mov si,offset a
mov ax,0h
up: cmp ax,[si]
;if ax>[si] jump to next
jnc next
;else store [si] in ax
mov ax,[si]
next: inc si
inc si
loop up
mov ans,ax

mov ax,4c00h
int 21h
end
```



```

076A:0005 8A0E0400      MOV     CL,[0004]
076A:0009 B500          MOV     CH,00
076A:000B BE0500      MOV     SI,0005
076A:000E B80000      MOV     AX,0000
076A:0011 3B04          CMP     AX,[SI]
076A:0013 7302          JNB     0017
076A:0015 8B04          MOV     AX,[SI]
076A:0017 46           INC     SI
076A:0018 46           INC     SI
076A:0019 E2F6          LOOP    0011
076A:001B A31900      MOV     [0019],AX
076A:001E B8004C      MOV     AX,4C00
-g

Program terminated normally
-d 076c:0000
076C:0000 4C CD 21 00 0A 11 01 13-02 03 12 04 14 05 54 06 L.!.....T.
076C:0010 45 07 90 08 67 09 67 0A-56 07 90 FF FF FF FF FF E...g.g.U.....
076C:0020 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF .....
076C:0030 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF .....
076C:0040 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF .....
076C:0050 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF .....
076C:0060 FF FF FF 26 00 FF FF FF-FF FF FF FF FF FF FF ...&.....
076C:0070 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF .....

```

3. Write a Program to sort 16 bits given numbers in ascending /descending order.

dosseg

model small

.8086

.data

n db 0AH

b dw 1210h,3443h,0ab54h,4337h,5675h,0086h,4209h,0cd2h,12ffh,4500h

;storing in b for reference

a dw 1210h,3443h,0ab54h,4337h,5675h,0086h,4209h,0cd2h,12ffh,4500h

temp db ?

i db ?

.code

mov ax,@data

mov ds,ax

mov ch,00h

mov cl,n

```
dec cl
mov si,offset a
mov i,00h
up1:
mov temp,cl
mov cl,n
sub cl,i
inc i
dec cl
mov bx,0000h
up2:mov ax,[si+bx]
mov dx,[si+bx+2]
cmp ax,dx
jnc next
mov [si+bx],dx
mov [si+bx+2],ax
next: inc bx
inc bx
loop up2

mov cl,temp
loop up1

mov ax,4c00h
int 21h
end
```

```

C:\TASM>debug 12q3.exe
-u
076A:0000 B86E07      MOV     AX,076E
076A:0003 8ED8        MOV     DS,AX
076A:0005 B500        MOV     CH,00
076A:0007 8A0E0800     MOV     CL,[0008]
076A:000B FEC9        DEC     CL
076A:000D BE1D00     MOV     SI,001D
076A:0010 C606320000     MOV     BYTE PTR [0032],00
076A:0015 880E3100     MOV     [0031],CL
076A:0019 8A0E0800     MOV     CL,[0008]
076A:001D 2A0E3200     SUB     CL,[0032]
-g

Program terminated normally
-d 076e:0000
076E:0000 E2 D3 B8 00 4C CD 21 00-0A 10 12 43 34 54 AB 37  ....L.!....C4T.7
076E:0010 43 75 56 86 00 09 42 D2-0C FF 12 00 45 54 AB 75  CuU...B.....ET.u
076E:0020 56 00 45 37 43 09 42 43-34 FF 12 10 12 D2 0C 86  U.E7C.BC4.....
076E:0030 00 01 09 FF FF FF FF FF-FF FF FF FF FF FF FF  .....
076E:0040 FF FF FF 26 00 FF FF FF-FF FF FF FF FF FF FF  ...&.....
076E:0050 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF  .....
076E:0060 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF  .....
076E:0070 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF  .....

```

initial :- 1210h, 3443h, 0ab54h, 4337h, 5675h, 0086h, 4209h, 0cd2h, 12ffh, 4500h  
 after Execution-after sorting in reverse order:- AB54 ,5675 ,4500, 4337 , 4209, 3443  
 ,12FF , 1210 ,0CD2 ,0086

4. Write a Program to find occurrences of a given number in a list of N numbers given through keyboard.

;U19CS066

dosseg

.model small

.8086

.data

n db ?

ans db 00h

x db ?

y db "\$"

cr equ 0dh

lf equ 0ah

msg1 db "n=\$"

msg2 db "Enter Numbers:\$"

msg3 db "Enter the number for freq: \$"

msg4 db " \$"

p db ?

print macro msg

mov ah,09h

mov dx,offset msg

int 21h

endm

read macro no

mov ah,01h

int 21h

sub al,'0'

mov bl,0ah

mul bl

mov no,al

mov ah,01h

int 21h

sub al,'0'

add no,al

endm

.code

mov ax,@data

mov ds,ax

print msg1

read n

print msg4

print msg3

read x

print msg4

print msg2

mov ch,00h

mov cl,n

up: read p

mov al,x

cmp al,p

jnz next

inc ans

next: inc p

print msg4

loop up

mov ax,4c00h

int 21h

end

```
076A:000A CD21      INT     21
076A:000C B401      MOV     AH,01
076A:000E CD21      INT     21
076A:0010 2C30      SUB     AL,30
076A:0012 B30A      MOV     BL,0A
076A:0014 F6E3      MUL     BL
076A:0016 A20200     MOV     [0002],AL
076A:0019 B401      MOV     AH,01
076A:001B CD21      INT     21
076A:001D 2C30      SUB     AL,30
076A:001F 00060200    ADD     [0002],AL
-g
n=10  Enter the number for freq: 1  Enter Numbers:2  3  1  3  4  2  1
3  4
Program terminated normally
-d 0773:0000
0773:0000 CD 21 0A 02 FA 24 6E 3D-24 45 6E 74 65 72 20 4E .!...$n=$Enter N
0773:0010 75 6D 62 65 72 73 3A 24-45 6E 74 65 72 20 74 68 umbers:$Enter th
0773:0020 65 20 6E 75 6D 62 65 72-20 66 6F 72 20 66 72 65 e number for fre
0773:0030 71 3A 20 24 20 20 24 19-FF FF FF FF FF FF FF FF q: $ $.....
0773:0040 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF .....
0773:0050 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF .....
0773:0060 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF .....
0773:0070 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF .....
```

Occurrence of 1= 2

5. Write a Program to move a string from source to destination.

dosseg

```
.model small
```

.8086

.data

cr equ 0dh

If equ 0ah

```
len db 18h
```

```
str1 db "Hello World!This is TASM$"
```

```
str2 db "$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$"
```

.code

```
mov ax,@data
```

```
mov ds,ax
```

```
mov es,ax
```

```
mov ch,00h
```

```
mov cl,len
```

```
mov si,offset str1
```

```
mov di,offset str2
```

cld

up: movsb

loop up

mov ah,09h

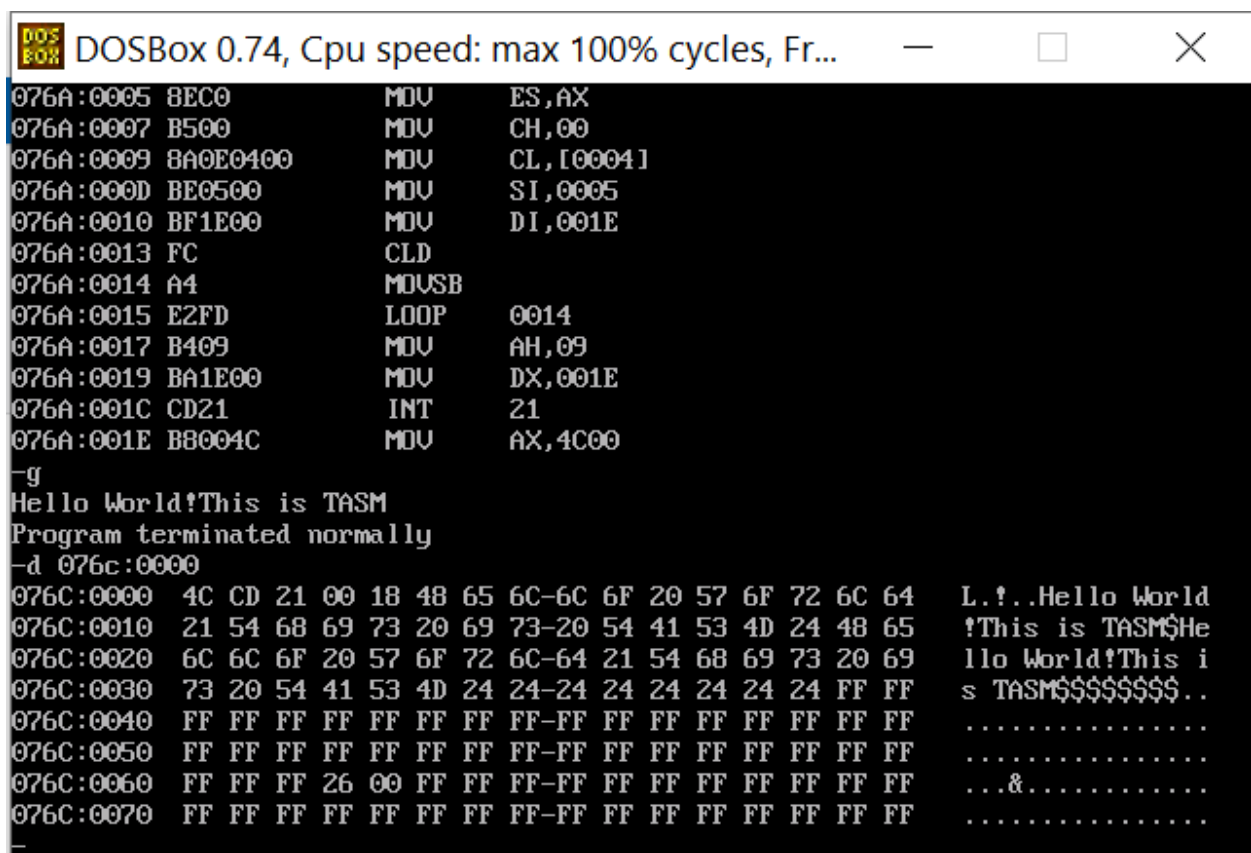
mov dx,offset str2

int 21h

mov ax,4c00h

int 21h

end



The screenshot shows a DOSBox 0.74 window with a black background and white text. The title bar reads "DOSBox 0.74, Cpu speed: max 100% cycles, Fr...". The main window displays assembly code and program output. The assembly code is as follows:

```
076A:0005 8EC0      MOV     ES,AX
076A:0007 B500      MOV     CH,00
076A:0009 8A0E0400  MOV     CL,[0004]
076A:000D BE0500      MOV     SI,0005
076A:0010 BF1E00      MOV     DI,001E
076A:0013 FC          CLD
076A:0014 A4          MOUSB
076A:0015 E2FD      LOOP    0014
076A:0017 B409      MOV     AH,09
076A:0019 BA1E00      MOV     DX,001E
076A:001C CD21      INT     21
076A:001E B8004C      MOV     AX,4C00
```

Below the assembly code, the program output is shown:

```
-g
Hello World!This is TASM
Program terminated normally
-d 076c:0000
076C:0000 4C CD 21 00 18 48 65 6C-6C 6F 20 57 6F 72 6C 64 L.!..Hello World
076C:0010 21 54 68 69 73 20 69 73-20 54 41 53 4D 24 48 65 !This is TASM$He
076C:0020 6C 6C 6F 20 57 6F 72 6C-64 21 54 68 69 73 20 69 llo World!This i
076C:0030 73 20 54 41 53 4D 24 24-24 24 24 24 24 24 24 24 FF FF s TASM$$$$$$$$..
076C:0040 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF ..
076C:0050 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF ..
076C:0060 FF FF FF 26 00 FF FF FF-FF FF FF FF FF FF FF FF ...&.....
076C:0070 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF ..
```

6. Write a Program to reverse a given string.

dosseg

model small

.8086



.data

cr equ 0dh

lf equ 0ah

len dw 0015h

str1 db " string reverse\$"

str2 db "\$"

.code

mov ax,@data

mov ds,ax

mov es,ax

mov cx,len

mov si,offset str1

add si,len

mov di,offset str2

cld

up: mov al,[si]

mov [di],al

inc di

dec si

loop up

```
mov ah,09h
```

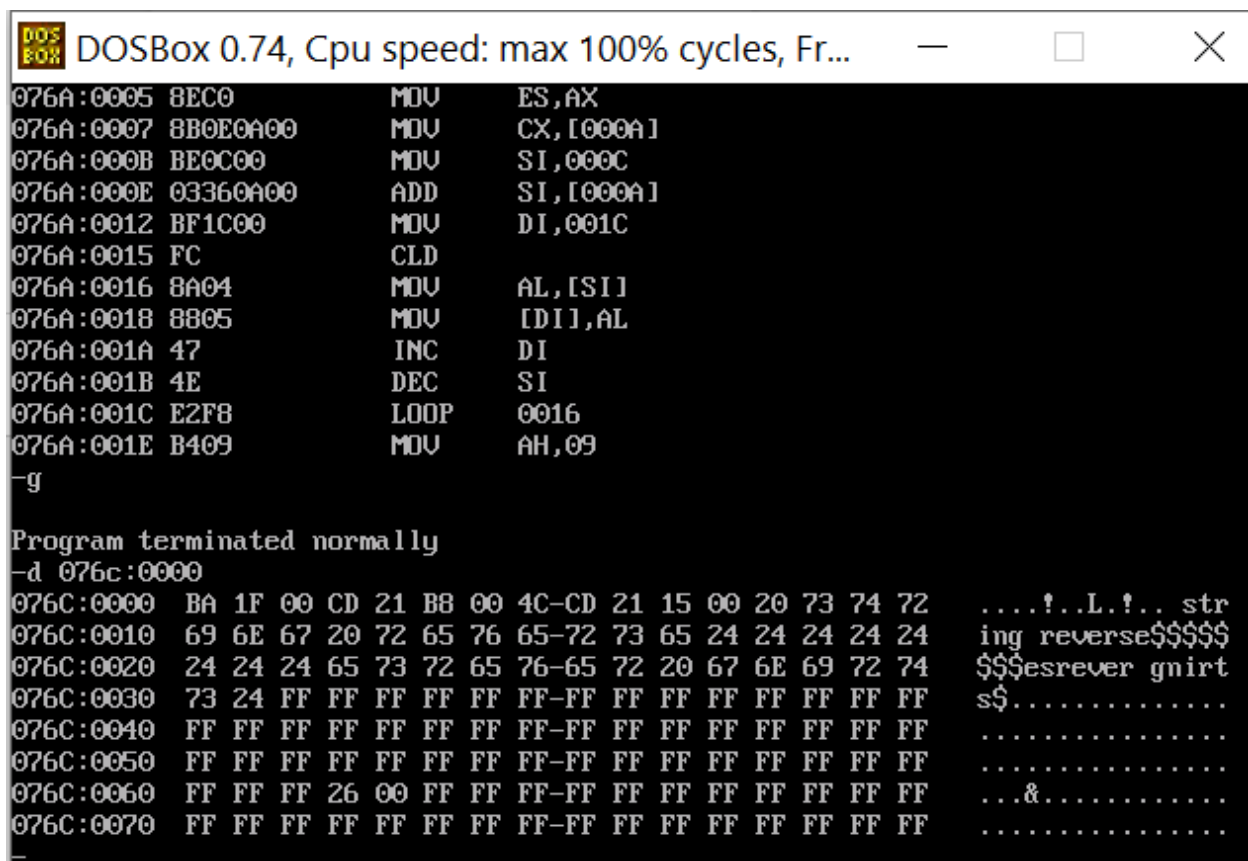
```
mov dx,offset str2+3
```

```
int 21h
```

```
mov ax,4c00h
```

```
int 21h
```

```
end
```



The screenshot shows a DOSBox window titled "DOSBox 0.74, Cpu speed: max 100% cycles, Fr...". The window contains a list of assembly instructions and their addresses, followed by a memory dump. The assembly instructions are:

```
076A:0005 8EC0      MOV     ES,AX
076A:0007 8B0E0A00   MOV     CX,[000A]
076A:000B BE0C00   MOV     SI,000C
076A:000E 03360A00   ADD     SI,[000A]
076A:0012 BF1C00   MOV     DI,001C
076A:0015 FC       CLD
076A:0016 8A04      MOV     AL,[SI]
076A:0018 8805      MOV     [DI],AL
076A:001A 47        INC     DI
076A:001B 4E        DEC     SI
076A:001C E2F8      LOOP    0016
076A:001E B409      MOV     AH,09
-g
```

Below the instructions, it says "Program terminated normally". Then, a memory dump is shown for address 076C:0000:

```
-d 076c:0000
076C:0000 BA 1F 00 CD 21 B8 00 4C-CD 21 15 00 20 73 74 72  ....!..L.!.. str
076C:0010 69 6E 67 20 72 65 76 65-72 73 65 24 24 24 24 24  ing reverse$$$$$
076C:0020 24 24 24 65 73 72 65 76-65 72 20 67 6E 69 72 74  $$$esrever gnirt
076C:0030 73 24 FF FF FF FF FF FF-FF FF FF FF FF FF FF  s$.....
076C:0040 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF  .....
076C:0050 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF  .....
076C:0060 FF FF FF 26 00 FF FF FF-FF FF FF FF FF FF FF  ...&.....
076C:0070 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF  .....
```

7. Write a Program to perform case conversion (U to L/ L to U) for a given string

```
dosseg
```

```
model small
```

```
.8086
```

```
.data
```

```
cr equ 0dh
lf equ 0ah
len dw 010h
str1 db "AbCdEfGh$"
```

```
.code
mov ax,@data
mov ds,ax
```

```
mov es,ax
mov cx,len
mov si,offset str1
cld
up: mov ax,[si]
cmp al,20
jz next
cmp al,41h
jc next
cmp al,5ah
jnc smalla
add al,20h
mov [si],al
jmp next
```

```
smalla: sub al,20h
mov [si],al
```

```
next: inc si
loop up
mov ah,09h
mov dx,offset str1
int 21h
```

```
mov ax,4c00h
int 21h
end
```

```

076A:0007 8B0E0800      MOV     CX,[0008]
076A:000B BE0A00      MOV     SI,000A
076A:000E FC          CLD
076A:000F 8B04      MOV     AX,[SI]
076A:0011 3C14      CMP     AL,14
076A:0013 7413      JZ      0028
076A:0015 3C41      CMP     AL,41
076A:0017 720F      JB      0028
076A:0019 3C5A      CMP     AL,5A
076A:001B 7307      JNB     0024
076A:001D 0420      ADD     AL,20
076A:001F 8B04      MOV     [SI],AL
-g
aBcDeFgH
Program terminated normally
-d 076d:0000
076D:0000 CD 21 B8 00 4C CD 21 00-10 00 61 42 63 44 65 46 .!...L.!...aBcDeF
076D:0010 67 48 24 DF DF DF DF DF-DF DF FF FF FF FF FF FF gH$.
076D:0020 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF
076D:0030 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF
076D:0040 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF
076D:0050 FF FF FF 26 00 FF FF FF-FF FF FF FF FF FF FF ...&...
076D:0060 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF
076D:0070 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF

```

8. Write a Program to merge two strings entered through keyboard

;Assignment 12 Q8 concat 2 strings

dosseg

model small

.8086

.data

cr equ 0dh

lf equ 0ah

len1 db ?

len2 db ?

str1 db "aaaaaaaaaaaaaaaaaaaa"

str2 db "aaaaaaaaaaaaaaaaaaaa"

final db "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa"

msg1 db "Enter String : \$"

msg3 db "final string : \$"

msg4 db "enter length : \$"

msg5 db " \$"

```
print macro msg
mov ah,09h
mov dx,offset msg
int 21h
endm
```

```
;;;;;;;;;;
```

```
read macro str
print msg4
mov ah,01h
int 21h
sub al,'0'
mov len2,al
mov cl,al
mov ch,00h
print msg5
print msg1
mov si,offset str
nextc: mov ah,01h
int 21h
mov [si],al
inc si
loop nextc
endm
```

```
;-----
```

```
listen macro str
print msg4
mov ah,01h
int 21h
sub al,'0'
mov len2,al
mov cl,al
mov ch,00h
print msg5
print msg1
mov si,offset str
next: mov ah,01h
int 21h
mov [si],al
```

```
inc si
loop next
endm
;-----
```

```
.code
mov ax,@data
mov ds,ax
```

```
mov es,ax
read str1
mov al,len2
mov len1,al
print msg5
```

```
listen str2
```

```
print msg5
```

```
cld
mov si,offset str1
mov di,offset final
mov ch,00h
mov cl,len1
rep movsb
```

```
dec di
mov si,offset str2
mov cl,len2
rep movsb
```

```
print msg3
print msg5
print final
```

```
mov ax,4c00h
int 21h
```

end

```
C:\TASM>debug 12q8.exe
-u
076A:0000 B87407      MOV     AX,0774
076A:0003 8ED8        MOV     DS,AX
076A:0005 8EC0        MOV     ES,AX
076A:0007 B409        MOV     AH,09
076A:0009 BA7D00      MOV     DX,007D
076A:000C CD21        INT     21
076A:000E B401        MOV     AH,01
076A:0010 CD21        INT     21
076A:0012 2C30        SUB     AL,30
076A:0014 A20B00      MOV     [000B],AL
076A:0017 8AC8        MOV     CL,AL
076A:0019 B500        MOV     CH,00
076A:001B B409        MOV     AH,09
076A:001D BA8D00      MOV     DX,008D
-g
enter length : 8 Enter String : helllowo enter length : 3 Enter String : bye fin
al string : helllowbye
Program terminated normally
-
```

9. Write a Program to search a character in a given string.

```
;Assignment 12 Q8 concat 2 strings
```

dosseg

model small

.8086

.data

cr equ 0dh

If equ 0ah

len1 db ?

char db ?

ans dw ?

```
str1 db "$$$$$$$$$$$$$$$$$$$$$$$$$$$$"
```

```
msg1 db "Enter String : $"
msg4 db "enter length : $"
msg2 db "enter the character to find :$"
msg5 db " $"
msg3 db "character found at :$"
```

```
;-----
```

```
print macro msg
```

```
mov ah,09h
```

```
mov dx,offset msg
```

```
int 21h
```

```
endm
```

```
;-----
```

```
read macro str
```

```
print msg4
```

```
mov ah,01h
```

```
int 21h
```

```
sub al,'0'
```

```
mov len1,al
```

```
mov cl,al
```

```
mov ch,00h
```

```
print msg5
```

```
print msg1
```

```
mov si,offset str
```

```
nextc: mov ah,01h
```



int 21h

mov [si],al

inc si

loop nextc

endm

;-----

.code

mov ax,@data

mov ds,ax

mov es,ax

print msg2

mov ah,01h

int 21h

mov char,al

read str1

print msg5

mov di,offset str1

mov cl,len1

```

mov ch,00h

mov al,char

repnz scasb

mov ans,di

sub ans,offset str1

```

```

mov ax,4c00h

```

```

int 21h

```

```

end

```

```

076A:0005 8EC0      MOV     ES,AX
076A:0007 B409      MOV     AH,09
076A:0009 BA4600   MOV     DX,0046
076A:000C CD21      INT     21
076A:000E B401      MOV     AH,01
076A:0010 CD21      INT     21
076A:0012 A20900   MOV     [0009],AL
076A:0015 B409      MOV     AH,09
076A:0017 BA3600   MOV     DX,0036
076A:001A CD21      INT     21
076A:001C B401      MOV     AH,01
076A:001E CD21      INT     21
-g
enter the character to find :wenter length : 8 Enter String : worldres
Program terminated normally
-d 0770:000
0770:0000 0C 00 B8 00 4C CD 21 00-08 77 01 00 77 6F 72 6C ....L.!...w..worl
0770:0010 64 72 65 73 24 24 24 24-24 24 24 24 24 24 24 24 dres$$$$$$$$$$$$
0770:0020 24 24 24 24 24 24 45 6E-74 65 72 20 53 74 72 69 $$$$Enter Stri
0770:0030 6E 67 20 3A 20 24 65 6E-74 65 72 20 6C 65 6E 67 ng : $enter leng
0770:0040 74 68 20 3A 20 24 65 6E-74 65 72 20 74 68 65 20 th : $enter the
0770:0050 63 68 61 72 61 63 74 65-72 20 74 6F 20 66 69 6E character to fin
0770:0060 64 20 3A 24 20 24 63 68-61 72 61 63 74 65 72 20 d :$ $character
0770:0070 66 6F 75 6E 64 20 61 74-20 3A 24 FF FF FF FF FF found at :$. ....

```

10. Write a Program to find occurrences of a given character in a given string through keyboard.

;Assigment 12 Q10 find occurence

dosseg

```
model small

.8086

.data

cr equ 0dh

lf equ 0ah

len1 db ?

char db ?

ans db 00h

str1 db "$$$$$$$$$$$$$$$$$$$$$$$$$$$$"

msg1 db "Enter String : $"

msg4 db "enter length : $"

msg2 db "enter the character to find :$"

msg5 db " $"

;-----

print macro msg

mov ah,09h

mov dx,offset msg

int 21h

endm

;-----

read macro str

print msg4

mov ah,01h

int 21h
```

```
sub al,'0'
mov len1,al
mov cl,al
mov ch,00h
print msg5
print msg1
mov si,offset str
nextc: mov ah,01h
int 21h
mov [si],al
inc si
loop nextc
endm
;-----
```

```
.code
mov ax,@data
mov ds,ax

mov es,ax
print msg2
mov ah,01h
int 21h
mov char,al
```

read str1

print msg5

mov di,offset str1

mov cl,len1

mov ch,00h

mov al,char

up: repnz scasb

jnz skip

inc ans

skip: inc di

loop up

mov ax,4c00h

int 21h

end

```

076A:0005 8EC0      MOV     ES,AX
076A:0007 B409      MOV     AH,09
076A:0009 BA4300    MOV     DX,0043
076A:000C CD21      INT     21
076A:000E B401      MOV     AH,01
076A:0010 CD21      INT     21
076A:0012 A20700    MOV     [0007],AL
076A:0015 B409      MOV     AH,09
076A:0017 BA3300    MOV     DX,0033
076A:001A CD21      INT     21
076A:001C B401      MOV     AH,01
076A:001E CD21      INT     21
-g
enter the character to find :aenter length : 9 Enter String : namastean
Program terminated normally
-d 0770:0000
0770:0000  F5 B8 00 4C CD 21 09 61-03 6E 61 6D 61 73 74 65  ...L.!a.namaste
0770:0010  61 6E 24 24 24 24 24 24-24 24 24 24 24 24 24 24  an$$$$$$$$$$$$$$$
0770:0020  24 24 24 45 6E 74 65 72-20 53 74 72 69 6E 67 20  $$$Enter String
0770:0030  3A 20 24 65 6E 74 65 72-20 6C 65 6E 67 74 68 20  : $enter length
0770:0040  3A 20 24 65 6E 74 65 72-20 74 68 65 20 63 68 61  : $enter the cha
0770:0050  72 61 63 74 65 72 20 74-6F 20 66 69 6E 64 20 3A  racter to find :
0770:0060  24 20 24 E8 06 F8 C3 8B-C1 F6 06 B9 6D 01 74 12  $ $......m.t.
0770:0070  3B 06 9C 74 74 23 3B 06-9E 74 74 2C 3B 06 A0 74  :..tt#:..tt;..t

```

11. Program to check whether given substring exist in a main string or not?

model small

.8086

.data

cr equ 0dh

lf equ 0ah

msg db "Enter String: \$"

msg1 db cr,lf,"Enter Substring: \$"

msg2 db cr,lf,"Found!\$"

msg3 db cr,lf,"Not Found\$"

st1 db 80 dup('\$')

st2 db 80 dup('\$')

```
print macro msg  
    mov ah,09h  
    mov dx,offset msg  
    int 21h  
endm
```

```
.code  
  
    mov ax,@data  
    mov ds,ax  
  
    print msg  
  
    mov si,offset st1  
    mov cx,0000h  
  
up:  
    mov ah,01h  
    int 21h  
    cmp al,0dh  
    je stp  
    mov [si],al  
    inc si  
    inc cx  
    jmp up  
  
stp:  
    print msg1
```

mov si,offset st2

mov bx,0000h

u:

mov ah,01h

int 21h

cmp al,0dh

je stop

mov [si],al

inc si

inc bx

jmp u

stop:

mov si,offset st1

mov di,offset st2

cmp cx,bx

jc down

mov dx,0000h

go:

mov al,[si]

cmp al,[di]

je me

mov dx,0000h

mov di,offset st2

inc si



```
loop go
jmp en
me:
inc dx
inc si
inc di
cmp dx,bx
jnz go
en:
cmp dx,bx
jnz down
print msg2
jmp e
down:
print msg3
e: mov ah,4ch
int 21h
end
```

```

076A:0005 8EC0      MOV     ES,AX
076A:0007 BF0F00    MOV     DI,000F
076A:000A BE1E00    MOV     SI,001E
076A:000D B500      MOV     CH,00
076A:000F 8A0E0C00  MOV     CL,[000C]
076A:0013 BE1E00    MOV     SI,001E
076A:0016 8A04      MOV     AL,[SI]
076A:0018 F2        REPNZ
076A:0019 AE        SCASB
076A:001A 83F900    CMP     CX,+00
076A:001D 7427      JZ      0046
076A:001F 893E4900  MOV     [0049],DI

```

-g

substring found

Program terminated normally

-d 076e:0000

```

076E:0000 8B 3E 49 00 E2 CD B8 00-4C CD 21 00 0F 04 00 48  .>I.....L.!....H
076E:0010 65 6C 6C 6F 20 45 76 65-72 79 6F 6E 65 24 6F 6E  ello Everyone$on
076E:0020 65 24 20 24 73 75 62 73-74 72 69 6E 67 20 66 6F  e$ $substring fo
076E:0030 75 6E 64 20 24 73 75 62-73 74 72 69 6E 67 20 6E  und $substring n
076E:0040 6F 74 20 66 6F 75 6E 64-24 1B 00 FF FF FF FF FF  ot found$.
076E:0050 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF  .....
076E:0060 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF  .....
076E:0070 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF  .....

```