

U18CO018
Shubham Shekhaliya
Assignment – 11 (MIT)

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

dosseg

model small

.8086

.data

ca dw ?

sum dw ?

nums dw 7H, 4142H, 4243H, 4748H , 4544H, 5050H, 4946H, 6167H

.code

mov ax, @data

mov ds, ax

mov si, 0H

mov cx, nums[si]

inc si

inc si

mov dx, 0H

mov ax, 0H

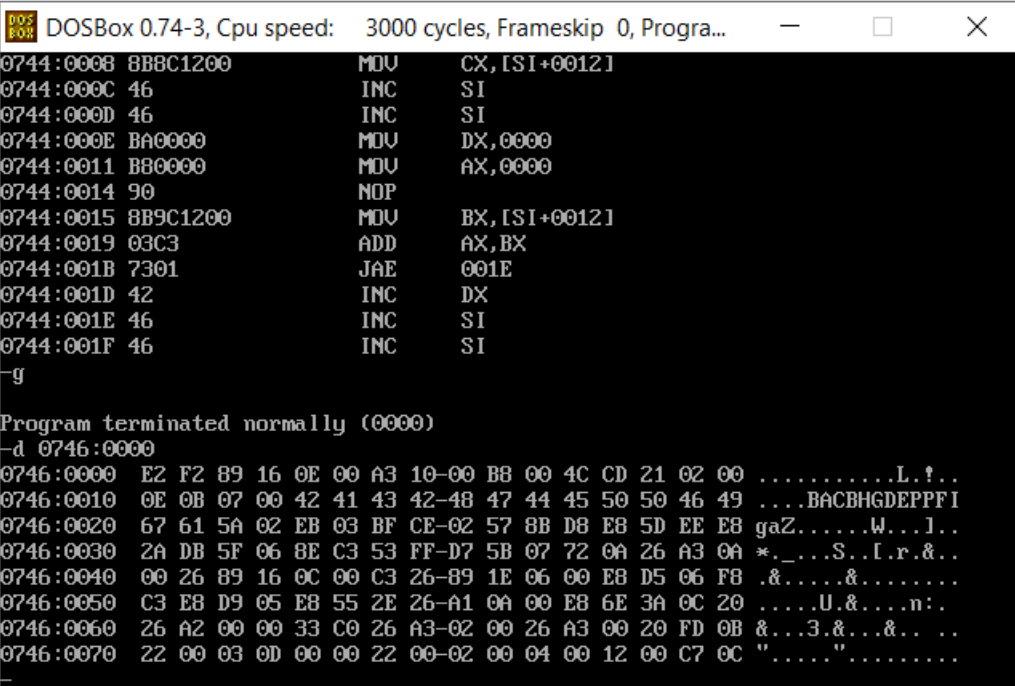
```

up: nop
mov bx, nums[si]
add ax, bx
jnc te
inc dx
te: inc si
inc si
loop up

mov ca, dx
mov sum, ax

mov ax, 4C00H
int 21H
end

```



```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
0744:0008 8B8C1200      MOV     CX,[SI+0012]
0744:000C 46             INC     SI
0744:000D 46             INC     SI
0744:000E BA0000      MOV     DX,0000
0744:0011 B80000      MOV     AX,0000
0744:0014 90            NOP
0744:0015 8B9C1200      MOV     BX,[SI+0012]
0744:0019 03C3          ADD     AX,BX
0744:001B 7301          JAE     001E
0744:001D 42            INC     DX
0744:001E 46            INC     SI
0744:001F 46            INC     SI
-g
Program terminated normally (0000)
-d 0746:0000
0746:0000 E2 F2 89 16 0E 00 A3 10-00 B8 00 4C CD 21 02 00 .....L.?!..
0746:0010 0E 0B 07 00 42 41 43 42-48 47 44 45 50 50 46 49 ...BACBHGDEPPFI
0746:0020 67 61 5A 02 EB 03 BF CE-02 57 8B D8 E8 5D EE E8 gaZ.....W...l..
0746:0030 2A DB 5F 06 8E C3 53 FF-D7 5B 07 72 0A 26 A3 0A *._...S..l.r.&..
0746:0040 00 26 89 16 0C 00 C3 26-89 1E 06 00 E8 D5 06 F8 .&.....&.....
0746:0050 C3 E8 D9 05 E8 55 2E 26-A1 0A 00 E8 6E 3A 0C 20 .....U.&....n:..
0746:0060 26 A2 00 00 33 C0 26 A3-02 00 26 A3 00 20 FD 0B &...3.&...&...
0746:0070 22 00 03 0D 00 00 22 00-02 00 04 00 12 00 C7 0C ".....".....

```

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

dosseg

model small

.8086

.data

largest dw ?

smallest dw ?

nums dw 7H, 4142H, 4243H, 4748H , 4544H, 5050H, 4946H, 6167H

.code

mov ax, @data

mov ds, ax

mov si,0

mov cx, nums[si]

inc si

inc si

mov largest, 0000H

mov smallest, 0FFFFH

up:

mov ax, nums[si]

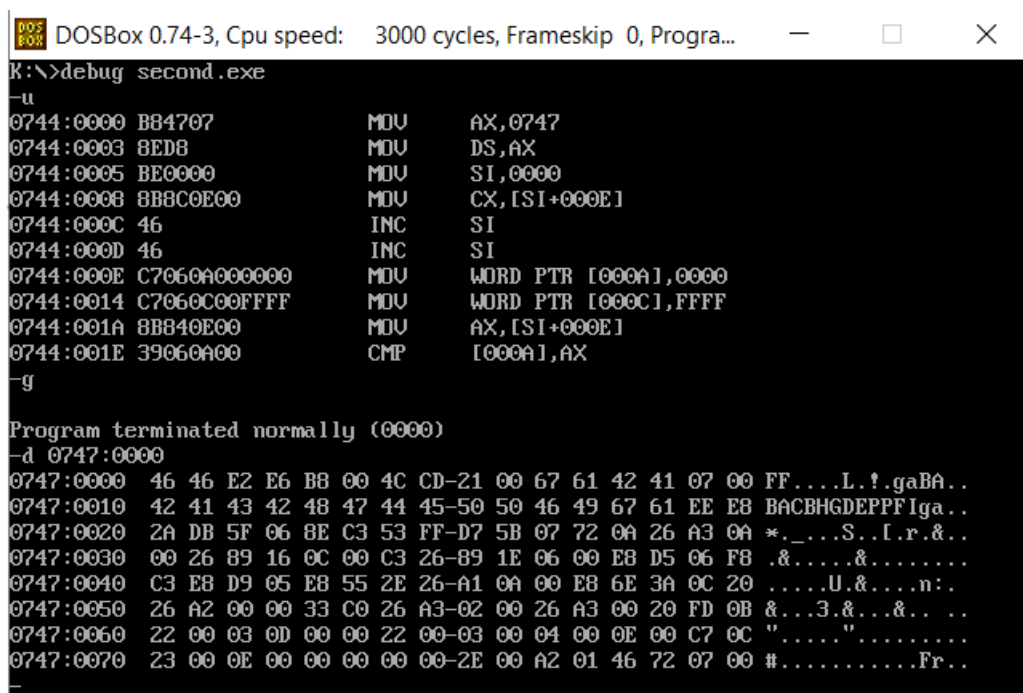
cmp largest, ax

```

jnc te1
mov largest, ax
te1: cmp smallest, ax
jc te2
mov smallest, ax
te2: inc si
inc si
loop up

mov ax, 4C00H
int 21H
end

```



DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

```

K:\>debug second.exe
-u
0744:0000 B84707      MOV     AX,0747
0744:0003 8ED8          MOV     DS,AX
0744:0005 BE0000      MOV     SI,0000
0744:0008 8B8C0E00      MOV     CX,[SI+000E]
0744:000C 46              INC     SI
0744:000D 46              INC     SI
0744:000E C7060A000000      MOV     WORD PTR [000A],0000
0744:0014 C7060C00FFFF      MOV     WORD PTR [000C],FFFF
0744:001A 8B840E00      MOV     AX,[SI+000E]
0744:001E 39060A00      CMP     [000A],AX
-g
Program terminated normally (0000)
-d 0747:0000
0747:0000 46 46 E2 E6 B8 00 4C CD-21 00 67 61 42 41 07 00 FF....L.!.gaBÂ..
0747:0010 42 41 43 42 48 47 44 45-50 50 46 49 67 61 EE E8 BACBHGDEPPFIga..
0747:0020 2A DB 5F 06 8E C3 53 FF-D7 5B 07 72 0A 26 A3 0A *._...S..[.r.&..
0747:0030 00 26 89 16 0C 00 C3 26-89 1E 06 00 E8 D5 06 F8 .&.....&.....
0747:0040 C3 E8 D9 05 E8 55 2E 26-A1 0A 00 E8 6E 3A 0C 20 .....U.&....n:..
0747:0050 26 A2 00 00 33 C0 26 A3-02 00 26 A3 00 20 FD 0B &...3.&...&...
0747:0060 22 00 03 0D 00 00 22 00-03 00 04 00 0E 00 C7 0C ".....".....
0747:0070 23 00 0E 00 00 00 00 00-2E 00 A2 01 46 72 07 00 #.....Fr..

```

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

dosseg

model small

.8086

.data

len dw 7H

nums dw 4142H, 4243H, 4748H , 4544H, 5050H, 4946H, 6167H

.code

mov ax, @data

mov ds, ax

mov cx, len

dec cx

outer:

 lea si,nums

 mov bx,0H

inner:

 inc bx

 mov ax, nums[si]

 inc si

 inc si

 cmp ax, nums[si]

 jb skip

 xchg ax, nums[si]

```
mov nums[si - 2], ax
```

```
skip:
```

```
cmp bx,cx
```

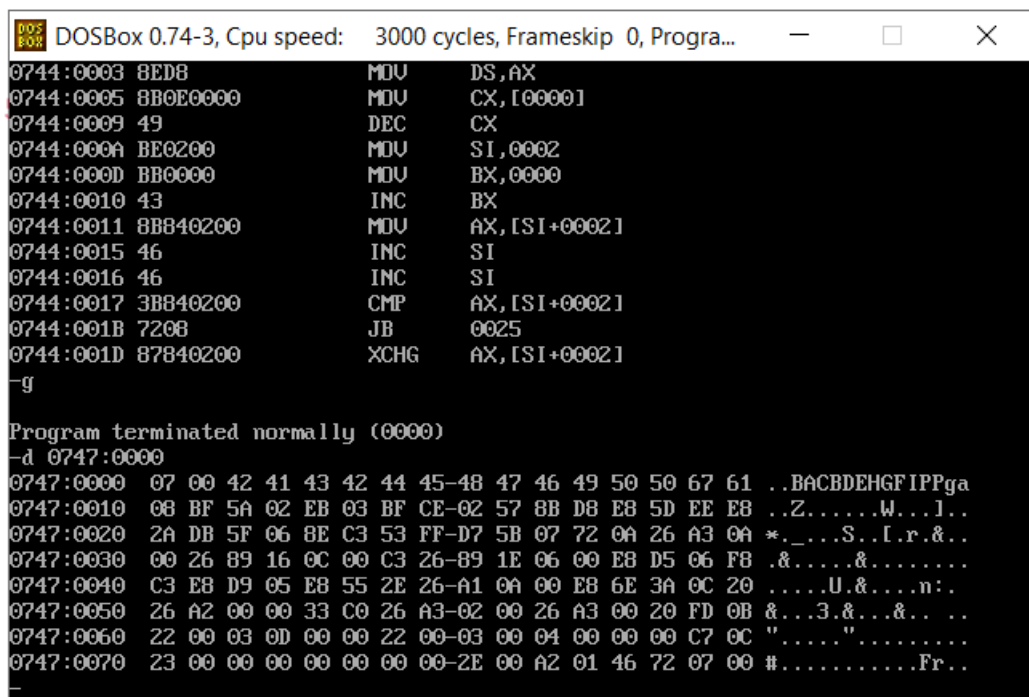
```
jl inner
```

```
LOOP outer
```

```
mov ax, 4C00H
```

```
int 21H
```

```
end
```



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
0744:0003 8ED8      MOV     DS,AX
0744:0005 8B0E0000    MOV     CX,[0000]
0744:0009 49          DEC     CX
0744:000A BE0200      MOV     SI,0002
0744:000D BB0000      MOV     BX,0000
0744:0010 43          INC     BX
0744:0011 8B840200    MOV     AX,[SI+0002]
0744:0015 46          INC     SI
0744:0016 46          INC     SI
0744:0017 3B840200    CMP     AX,[SI+0002]
0744:001B 7208        JB      0025
0744:001D 87840200    XCHG    AX,[SI+0002]
-g
Program terminated normally (0000)
-d 0747:0000
0747:0000 07 00 42 41 43 42 44 45-48 47 46 49 50 50 67 61 ..BACBDEHGFIPPga
0747:0010 08 BF 5A 02 EB 03 BF CE-02 57 8B D8 E8 5D EE E8 ..Z.....W...l..
0747:0020 2A DB 5F 06 8E C3 53 FF-D7 5B 07 72 0A 26 A3 0A *. _...S...[.r.&..
0747:0030 00 26 89 16 0C 00 C3 26-89 1E 06 00 E8 D5 06 FB .&.....&.....
0747:0040 C3 E8 D9 05 E8 55 2E 26-A1 0A 00 E8 6E 3A 0C 20 ....U.&....n:..
0747:0050 26 A2 00 00 33 C0 26 A3-02 00 26 A3 00 20 FD 0B &...3.&...&..
0747:0060 22 00 03 0D 00 00 22 00-03 00 04 00 00 00 C7 0C ".....".....
0747:0070 23 00 00 00 00 00 00 00-2E 00 A2 01 46 72 07 00 #.....Fr..
_
```

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

```
dosseg
```

```
.model small
```

.data

n db ?

list db 80 dup(0)

key db ?

numMsg db "How many digits? \$"

valMsg db "Enter the value: \$"

getKey db "Enter the key: \$"

displayMsg db "The count of the given key is: \$"

cnt db 0

printstring macro msg

 mov ah, 09h

 mov dx, offset msg

 int 21h

endm

printchar macro ch

 mov ah, 02h

 mov dl, ch

 int 21h

endm

scan2 macro num

```
mov ah, 01h
int 21h
sub al, 30h
mov bh, 0ah
mul bh
mov bh, al
mov ah, 01h
int 21h
sub al, 30h
add bh, al
mov num, bh
printchar 0dh
printchar 0ah
endm
```

```
scanarray2 macro list, n
    mov si, offset list
    mov ch, 00
    mov cl, n
up:
    printstring valMsg
    scan2 [si]
    inc si
    loop up
endm
```


print2 macro num

 printchar 0dh

 printchar 0ah

 mov al, num

 add al, 0h

 daa

 mov bl, al

 and al, 0F0h

 shr al, 04h

 add al, 30h

 printchar al

 and bl, 0Fh

 add bl, 30h

 printchar bl

endm

findOccurence macro list, n, key, cnt

 mov si, offset list

 mov ch, 00

 mov cl, n

 upa:

 mov al, [si]

 cmp key, al

 jne skipa

 inc cnt

```
    skipa: inc si
    loop upa
endm
```

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

```
printstring numMsg
scan2 n
scanarray2 list, n
```

```
printstring getKey
scan2 key
```

```
findOccurence list, n, key, cnt
printstring displayMsg
print2 cnt
```

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

```

Remaining memory: 471k

K:\>tlink fourth
Turbo Link Version 2.0 Copyright (c) 1987, 1988 Borland International
Warning: no stack

K:\>debug fourth.exe
-g
How many digits? 10
Enter the value: 11
Enter the value: 12
Enter the value: 13
Enter the value: 15
Enter the value: 14
Enter the value: 15
Enter the value: 15
Enter the value: 14
Enter the value: 18
Enter the value: 12
Enter the key: 15
The count of the given key is:
03
Program terminated normally (0000)

```

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

dosseg

.model small

.data

msg db "Enter string: \$"

st1 db 80 dup('\$')

st2 db 80 dup('\$')

printstring macro msg

mov ah, 09h

mov dx, offset msg

int 21h

endm

scanstring macro st

mov si, offset st

```
mov cx,0000h
upe: mov ah, 01h
int 21h
cmp al, 0dh
je stp
mov [si], al
inc si
inc cx
jmp upe
stp: nop
endm
```

```
move macro st1, st2, cnt
    mov si, offset st1
    mov cx, cnt
    mov di, offset st2
    upb:
    mov al, [si]
    mov [di], al
    inc si
    inc di
    loop upb
endm
```

```
.code
```

mov ax, @data

mov ds, ax

printstring msg

scanstring st1

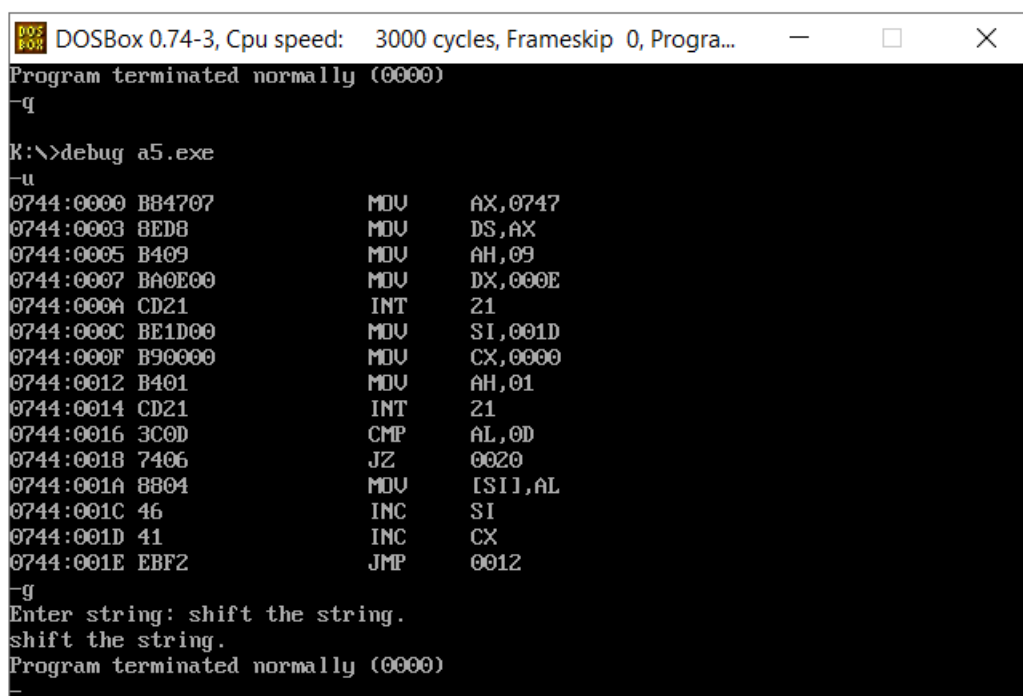
move st1, st2, cx

printstring st2

mov ax, 4c00h

int 21h

end



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
Program terminated normally (0000)
-q
K:\>debug a5.exe
-u
0744:0000 B84707      MOV     AX,0747
0744:0003 8ED8          MOV     DS,AX
0744:0005 B409          MOV     AH,09
0744:0007 BA0E00    MOV     DX,000E
0744:000A CD21          INT     21
0744:000C BE1D00    MOV     SI,001D
0744:000F B90000    MOV     CX,0000
0744:0012 B401          MOV     AH,01
0744:0014 CD21          INT     21
0744:0016 3C0D          CMP     AL,0D
0744:0018 7406          JZ      0020
0744:001A 8B04          MOV     [SI],AL
0744:001C 46            INC     SI
0744:001D 41            INC     CX
0744:001E EBF2          JMP     0012
-g
Enter string: shift the string.
shift the string.
Program terminated normally (0000)
-
```

5. Write a Program to reverse a given string.

dosseg

```
.model small
```

```
.8086
```

```
.data
```

```
msg db "Enter a string to reverse : $"
```

```
st1 db 80 dup('$')
```

```
scanstring macro str
```

```
    mov si, offset str
```

```
    mov cx, 0000h
```

```
upe:  mov ah, 01h
```

```
    int 21h
```

```
    cmp al, 0dh
```

```
    je stp
```

```
    mov [si], al
```

```
    inc si
```

```
    inc cx
```

```
    jmp upe
```

```
stp:  nop
```

```
endm
```

```
printstring macro msg
```

```
    mov ah, 09h
```

```
    mov dx, offset msg
```

```
    int 21h
```

```
endm
```

```
reverse macro st1
    mov si, offset st1
    mov di, si
    add di, cx
    dec di
    shr cx, 1
upc:  mov al, [si]
    mov bl, [di]
    mov [di], al
    mov [si], bl
    dec di
    inc si
    loop upc
endm
```

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

```
printstring msg
scanstring st1
```

```
reverse st1
```

printstring st1

mov ax, 4c00h

int 21h

end

```
Program terminated normally (0000)
-q
K:\>debug a6.exe
-u
0744:0000 B84807      MOV     AX,0748
0744:0003 8ED8        MOV     DS,AX
0744:0005 B409        MOV     AH,09
0744:0007 BA0400     MOV     DX,0004
0744:000A CD21      INT     21
0744:000C BE2100     MOV     SI,0021
0744:000F B90000     MOV     CX,0000
0744:0012 B401        MOV     AH,01
0744:0014 CD21      INT     21
0744:0016 3C0D      CMP     AL,0D
0744:0018 7406      JZ      0020
0744:001A 8804      MOV     [SI],AL
0744:001C 46         INC     SI
0744:001D 41         INC     CX
0744:001E EBF2      JMP     0012
-g
Enter a string to reverse : reverse
esrever
Program terminated normally (0000)
-
```

6. Write a Program to perform case conversion (U to L, L to U) for a given string.

dosseg

.model small

.data

msg db "Enter string: \$"

st1 db 80 dup('\$')

printstring macro msg


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

```
scanstring macro str
    mov si, offset str
    mov cx, 0000h
upe: mov ah, 01h ;scan
    int 21h
    cmp al, 0dh ;cmp with carriage return
    je stp
    mov [si], al
    inc si
    inc cx
    jmp upe
stp: nop
endm
```

```
printchar macro ch
    mov ah, 02h
    mov dl, ch
    int 21h
endm
```

```
caseconverse macro str
```

```
    mov si, offset str;
upf:
    mov al, [si]
    cmp al, 'a'
    jb checkuppercase
    cmp al, 'z'
    ja skip
    sub al, 20h
    jmp skip
uppercase:
    cmp al, 'A'
    jb skip
    cmp al, 'Z'
    ja skip
    add al, 20h
skip:
    mov [si],al
    inc si
    loop upf
endm
```

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

scanstring st1

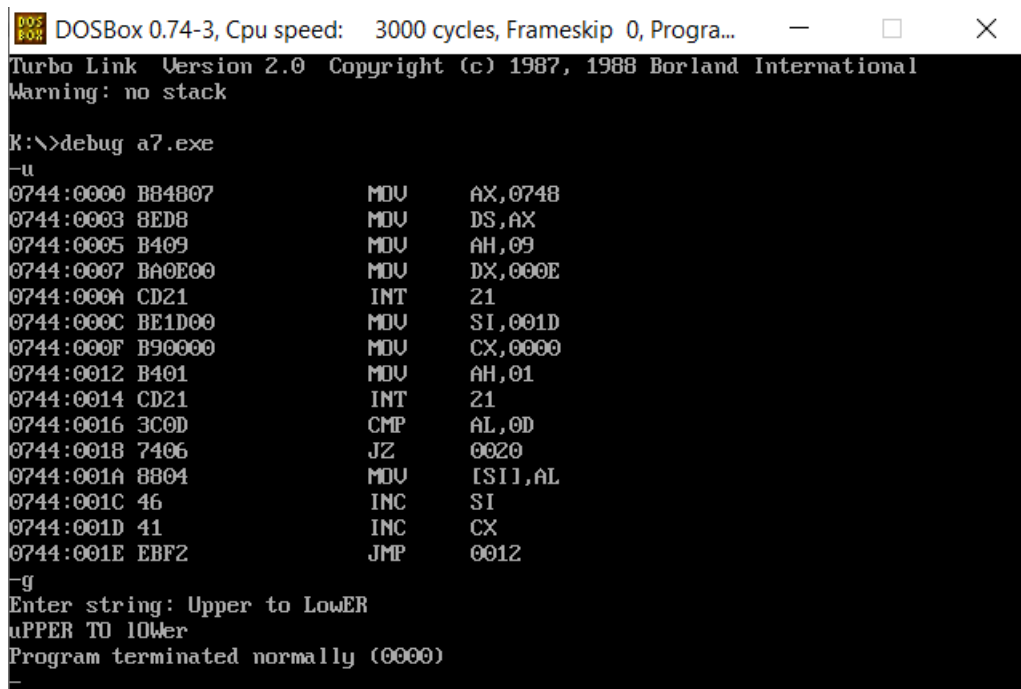
caseconverse st1

printstring st1

mov ax, 4c00h

int 21h

end



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
Turbo Link Version 2.0 Copyright (c) 1987, 1988 Borland International
Warning: no stack

K:\>debug a7.exe
-u
0744:0000 B84807      MOV     AX,0748
0744:0003 8ED8        MOV     DS,AX
0744:0005 B409        MOV     AH,09
0744:0007 BA0E00      MOV     DX,000E
0744:000A CD21        INT     21
0744:000C BE1D00      MOV     SI,001D
0744:000F B90000      MOV     CX,0000
0744:0012 B401        MOV     AH,01
0744:0014 CD21        INT     21
0744:0016 3C0D        CMP     AL,0D
0744:0018 7406        JZ      0020
0744:001A 8B04        MOV     [SI],AL
0744:001C 46          INC     SI
0744:001D 41          INC     CX
0744:001E EBF2        JMP     0012
-g
Enter string: Upper to LowER
uPPER TO lOwer
Program terminated normally (0000)
```

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

dosseg

.model small

.data

msg db "Enter a string: \$";

st1 db 80 dup('\$');

st2 db 80 dup('\$')

cnt1 dw 0

cnt2 dw 0

printstring macro msg

mov ah, 09h

mov dx, offset msg

int 21h

endm

scanstring macro str

local @@up2, @@skip1

mov si, offset str

mov cx, 0000h

@@up2: mov ah, 01h

int 21h

cmp al, 0dh

je @@skip1

mov [si], al

inc si

inc cx;

jmp @@up2

@@skip1: nop

endm

mergestring macro st1,cnt1,st2,cnt2

mov di, offset st1

add di, cnt1

mov si, offset st2

```
    mov cx, cnt2
upf:
    mov al, [si];
    mov [di], al;
    inc si;
    inc di;
    loop upf
endm
```

```
.code
```

```
mov ax, @data
mov ds, ax
```

```
printstring msg
scanstring st1
mov cnt1, cx;
```

```
printstring msg
```

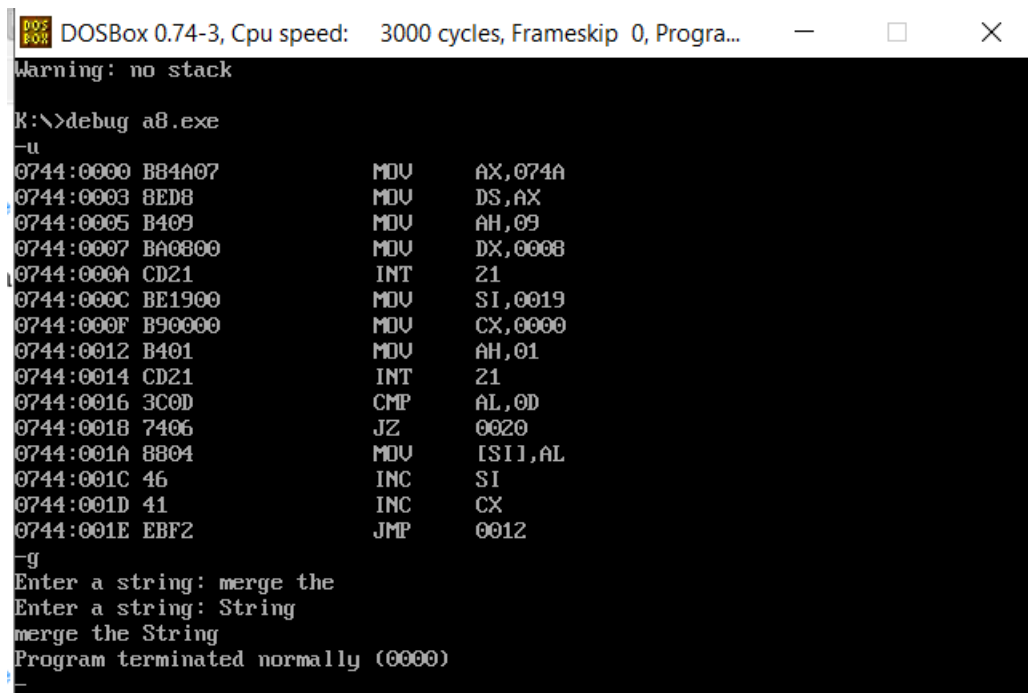
```
scanstring st2
mov cnt2, cx;
```

```
mergestring st1,cnt1,st2,cnt2
printstring st1
```

mov ax, 4c00h

int 21h

end



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
Warning: no stack
K:\>debug a8.exe
-u
0744:0000 B84A07      MOV     AX,074A
0744:0003 8ED8        MOV     DS,AX
0744:0005 B409        MOV     AH,09
0744:0007 BA0800    MOV     DX,0008
0744:000A CD21        INT     21
0744:000C BE1900    MOV     SI,0019
0744:000F B90000    MOV     CX,0000
0744:0012 B401        MOV     AH,01
0744:0014 CD21        INT     21
0744:0016 3C0D        CMP     AL,0D
0744:0018 7406        JZ      0020
0744:001A 8804        MOV     [SI],AL
0744:001C 46          INC     SI
0744:001D 41          INC     CX
0744:001E EBF2        JMP     0012
-g
Enter a string: merge the
Enter a string: String
merge the String
Program terminated normally (0000)
```

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

dosseg

.model small

.data

msg db "Enter a string: \$"

msg2 db "Enter a key: \$"

key db ?

pos db ?

st1 db 80 dup('\$')

printchar macro ch

```
    mov ah, 02h
    mov dl, ch
    int 21h
endm
```

```
print2 macro num
```

```
    printchar 0dh
    printchar 0ah
    mov al, num
    add al, 0h
    daa
    mov bl, al
    and al, 0F0h
    shr al, 04h
    add al, 30h
    printchar al
    and bl, 0Fh
    add bl, 30h
    printchar bl
endm
```

```
printstring macro msg
```

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

```
scanstring macro str
    mov si, offset str
    mov cx, 0000h
upe:  mov ah, 01h
    int 21h
    cmp al, 0dh
    je stp
    mov [si], al
    inc si
    inc cx
    jmp upe
stp: nop
endm
```

```
givepos macro st1, key, pos
    mov si, offset st1
    mov pos, 0
up:
    mov al, [si];
    cmp al, key
    je quit
    inc pos
    inc si
    loop up
```


quit: nop

endm

.code

mov ax, @data

mov ds, ax

printstring msg

scanstring st1

printstring msg2

mov ah, 01h

int 21h

mov key, al;

givepos st1, key, pos

print2 pos

mov ax, 4c00h

int 21h

end

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
-q
K:\>debug a9.exe
-u
0744:0000 B84B07      MOV     AX,074B
0744:0003 8ED8        MOV     DS,AX
0744:0005 B409        MOV     AH,09
0744:0007 BA0E00      MOV     DX,000E
0744:000A CD21        INT     21
0744:000C BE2F00      MOV     SI,002F
0744:000F B90000      MOV     CX,0000
0744:0012 B401        MOV     AH,01
0744:0014 CD21        INT     21
0744:0016 3C0D        CMP     AL,0D
0744:0018 7406        JZ      0020
0744:001A 8804        MOV     [SI],AL
0744:001C 46          INC     SI
0744:001D 41          INC     CX
0744:001E EBF2        JMP     0012
-g
Enter a string: abbacasd
Enter a key: c
04
Program terminated normally (0000)
-
```

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

dosseg

.model small

.data

msg db "Enter a string: \$"

msg2 db "Enter a key: \$"

st1 db 80 dup('\$')

key db ?

cnt db ?

n dw ?

printstring macro msg

mov ah, 09h

mov dx, offset msg

```
    int 21h  
endm
```

```
printchar macro ch  
    mov ah, 02h  
    mov dl, ch  
    int 21h  
endm
```

```
print2 macro num  
    printchar 0dh  
    printchar 0ah  
    mov al, num  
    add al, 0h  
    daa  
    mov bl, al  
    and al, 0F0h  
    shr al, 04h  
    add al, 30h  
    printchar al  
    and bl, 0Fh  
    add bl, 30h  
    printchar bl  
endm
```

```
scanstring macro str
```

```
    mov si, offset str
    mov cx, 0000
upe:   mov ah, 01h
    int 21h
    cmp al, 0dh
    je stp
    mov [si], al
    inc si
    inc cx
    jmp upe
stp:   nop
endm
```

findoccurence macro st1,n, key, cnt

```
    mov si, offset st1
    mov cx, n
    mov cnt,0000h
upd:
    mov al, [si];
    cmp al, key
    jnz skip2
    inc cnt
skip2:
    inc si
    loop upd
endm
```

.code

mov ax, @data

mov ds, ax

printstring msg

scanstring st1

mov n, cx

printstring msg2

mov ah, 01

int 21h

mov key, al

findoccurrence st1,n, key, cnt

print2 cnt

mov ah, 4ch

int 21h

end

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
Warning: no stack
K:\>debug a10.exe
-u
0744:0000 B84C07      MOV     AX,074C
0744:0003 8ED8        MOV     DS,AX
0744:0005 B409        MOV     AH,09
0744:0007 BA0400      MOV     DX,0004
0744:000A CD21        INT     21
0744:000C BE2300      MOV     SI,0023
0744:000F B90000      MOV     CX,0000
0744:0012 B401        MOV     AH,01
0744:0014 CD21        INT     21
0744:0016 3C0D        CMP     AL,0D
0744:0018 7406        JZ      0020
0744:001A 8B04        MOV     [SI],AL
0744:001C 46          INC     SI
0744:001D 41          INC     CX
0744:001E EBF2        JMP     0012
-g
Enter a string: sadbaafvdas
Enter a key: a
04
Program terminated normally (0034)
```

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

dosseg

.model small

.data

msg db "Enter a string: \$"

msg1 db "Enter a substring: \$"

msg2 db "The substring exists \$"

msg3 db "The substring doesnt exist \$"

st1 db 80 dup('\$')

st2 db 80 dup('\$')

cnt1 dw ?

cnt2 dw ?

printstring macro msg

mov ah, 09h

mov dx, offset msg

int 21h

endm

scanstring macro str

local upe, stp

mov si, offset str

mov cx, 0000h

upe: nop

mov ah, 01h

int 21h

cmp al, 0dh

je stp

mov [si], al

inc si

inc cx

jmp upe

stp: nop

endm

printchar macro ch

mov ah, 02h

mov dl, ch

int 21h

endm

print2 macro num

 printchar 0dh

 printchar 0ah

 mov al, num

 add al, 0h

 daa

 mov bl, al

 and al, 0F0h

 shr al, 04h

 add al, 30h

 printchar al

 and bl, 0Fh

 add bl, 30h

 printchar bl

endm

checkifexists macro st1,cnt1,st2,cnt2

 mov cx, cnt1

 mov si, offset st1

 mov bx, si

 up: nop

 mov si, bx

 mov di, offset st2

 up1: nop

mov al, [di]

cmp al, '\$'

je success

cmp al, [si]

jne break

inc si;

inc di

jmp up1

break:

inc bx

loop up

printstring msg3

jmp endf

success: printstring msg2

endf: nop

endm

.code

mov ax, @data

mov ds, ax

printstring msg

scanstring st1

mov cnt1, cx

printstring msg1

scanstring st2

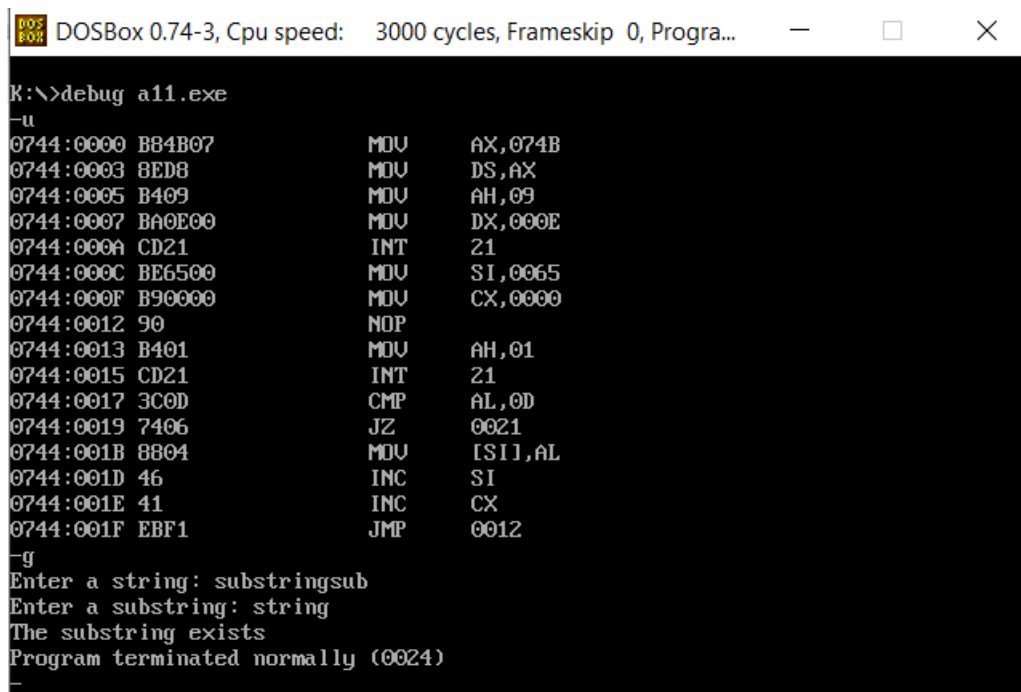
mov cnt2, cx

checkifexists st1,cnt1,st2,cnt2

mov ah, 4ch

int 21h

end



DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

```
K:\>debug a11.exe
-u
0744:0000 B84B07      MOV     AX,074B
0744:0003 BED8       MOV     DS,AX
0744:0005 B409       MOV     AH,09
0744:0007 BA0E00      MOV     DX,000E
0744:000A CD21       INT     21
0744:000C BE6500      MOV     SI,0065
0744:000F B90000      MOV     CX,0000
0744:0012 90          NOP
0744:0013 B401       MOV     AH,01
0744:0015 CD21       INT     21
0744:0017 3C0D       CMP     AL,0D
0744:0019 7406       JZ      0021
0744:001B 8804       MOV     [SI],AL
0744:001D 46          INC     SI
0744:001E 41          INC     CX
0744:001F EBF1       JMP     0012
-g
Enter a string: substringsub
Enter a substring: string
The substring exists
Program terminated normally (0024)
-
```