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

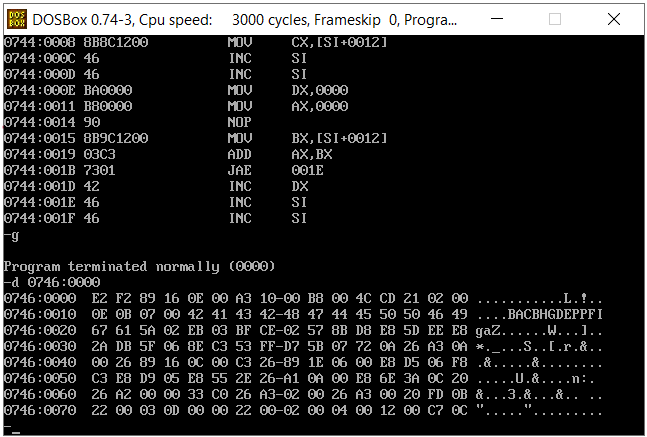
mov ca, dx

mov sum, ax

mov ax, 4C00H

int 21H

end



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

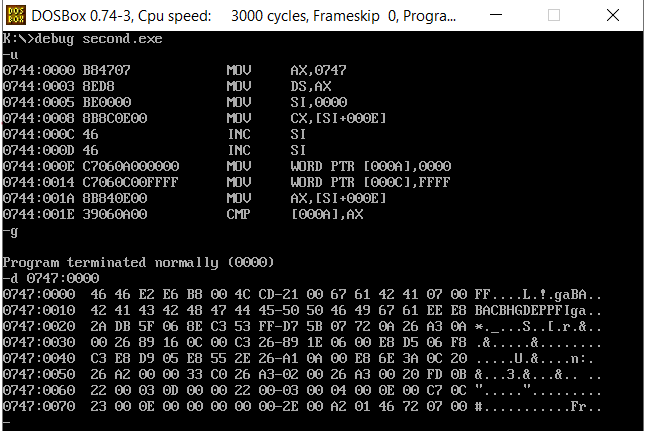
inc si

loop up

mov ax, 4C00H

int 21H

end



1. 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

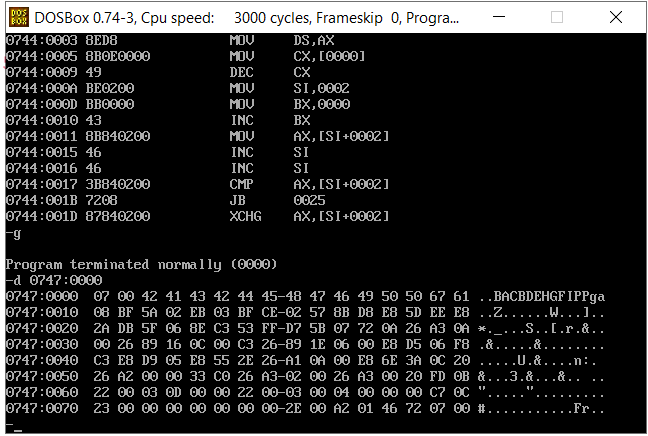
jl inner

LOOP outer

mov ax, 4C00H

int 21H

end



1. 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

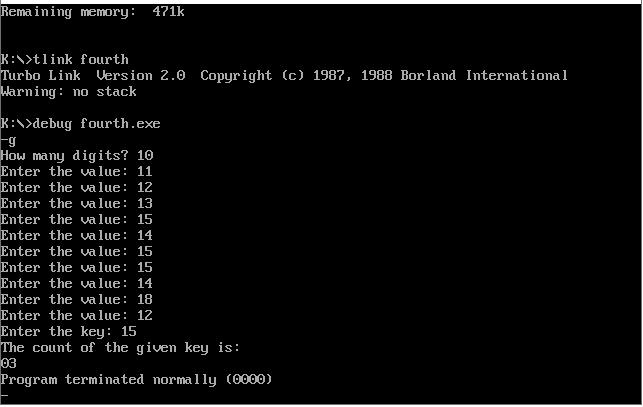
printstring displayMsg

print2 cnt

mov ax, 4c00h

int 21h

end



1. 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

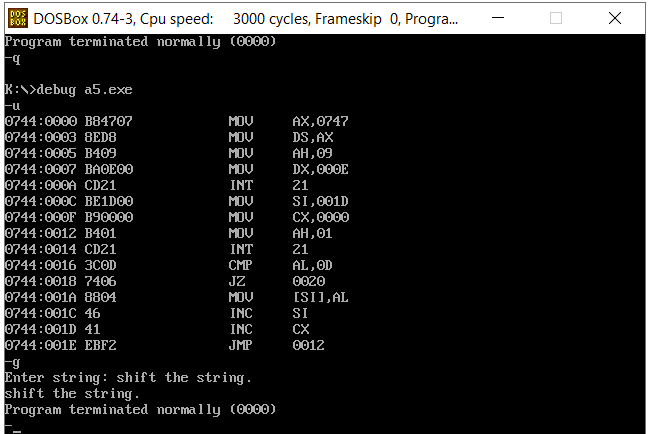
move st1, st2, cx

printstring st2

mov ax, 4c00h

int 21h

end



1. 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

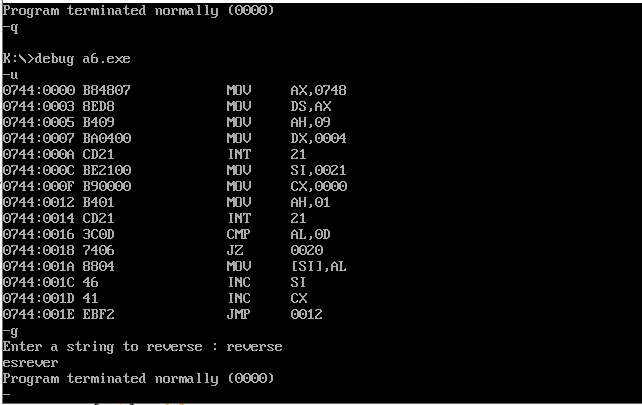
reverse st1

printstring st1

mov ax, 4c00h

int 21h

end



1. 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

checkuppercase:

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

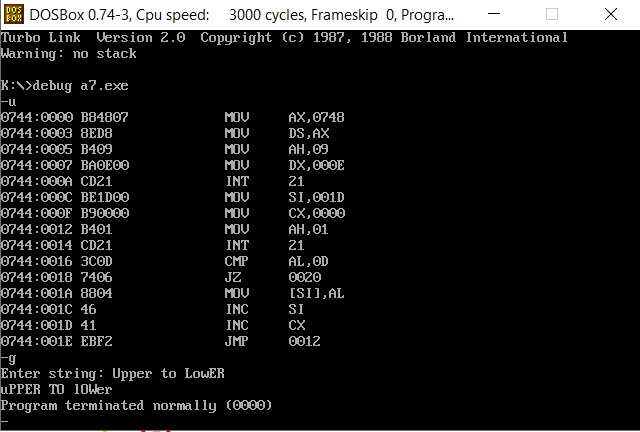
caseconverse st1

printstring st1

mov ax, 4c00h

int 21h

end



1. 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;

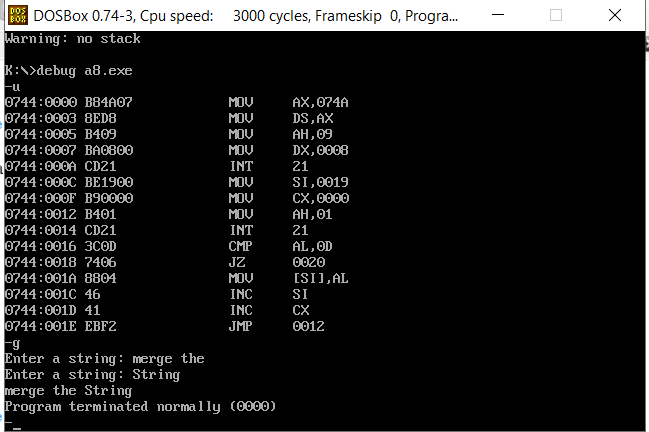
mergestring st1,cnt1,st2,cnt2

printstring st1

mov ax, 4c00h

int 21h

end



1. 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;

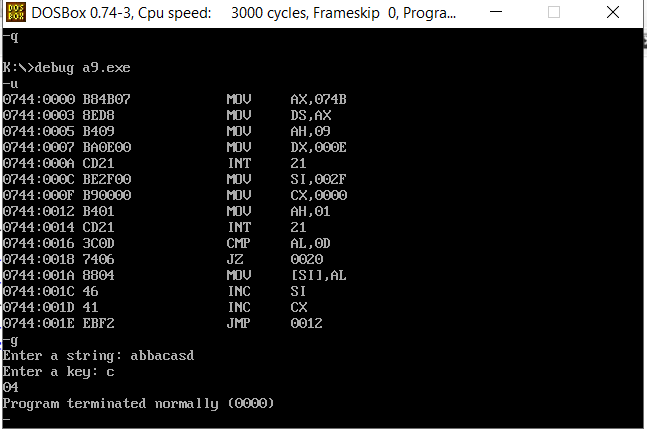
givepos st1, key, pos

print2 pos

mov ax, 4c00h

int 21h

end



1. 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

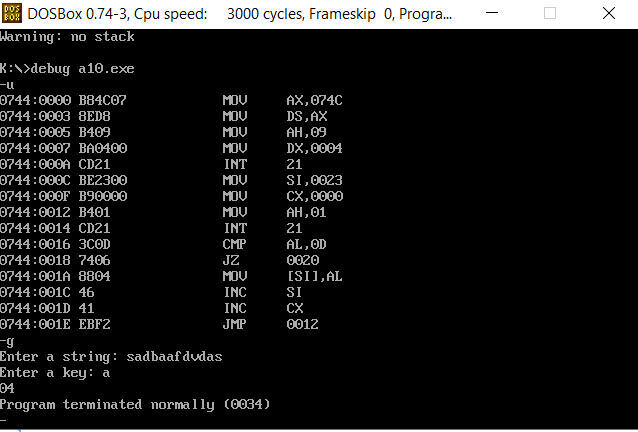
findoccurence st1,n, key, cnt

print2 cnt

mov ah, 4ch

int 21h

end



1. 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

