**M.I.T. LAB Assignment – 12**

**U19CS012**

**\*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.

TASM Code:

.model small

.stack 100

.8086

.data

; Number of Elements 'n'

n dw 0005H

; Data of 5 Elements

a dw 1211H, 0A145H, 4817H, 3C12H, 12F7h

; Answer & Carry

ans dw ?

carry db ?

.code

mov ax,@data

mov ds,ax

; Intialize the Counter = n

mov cx, n

; Intialize the SI

mov si,offset a

; Intial Sum = 0

mov ax,0000h

; Intial Carry = 0

mov dl,00

up:     add ax, [si]

        jnc next    ; IF Carry Generated

        inc dl

next:   inc si

        inc si

        loop up

mov ans,ax

mov carry,dl

mov ax, 4C00H

int 21h

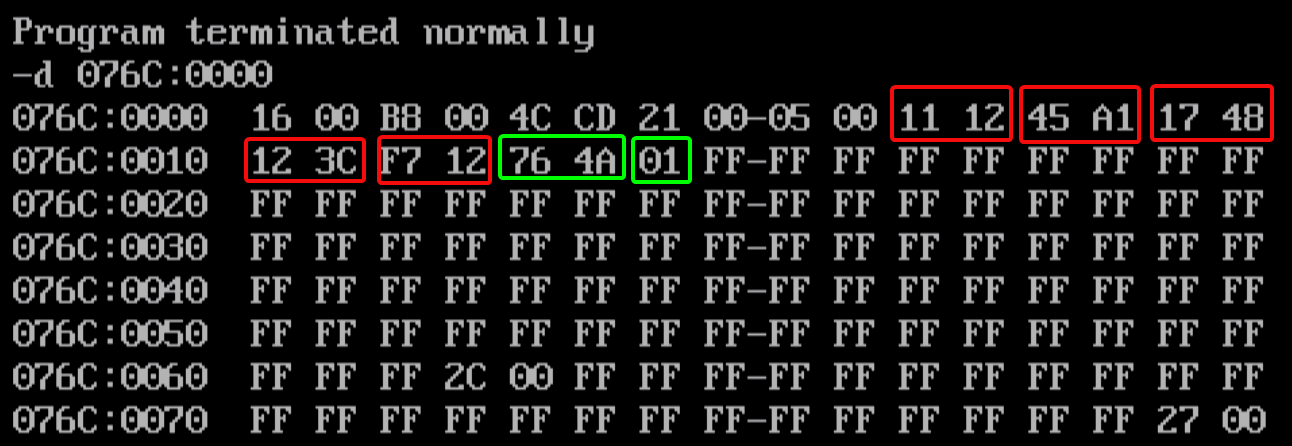
end

Input:

**Numbers:** {1211, A145, 4817, 3C12, 12F7}

Decimal = {4625 + 41285 + 18455 + 15378 + 4855 = 84,598 = (**1**4A76) H}

Output:



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

TASM Code:

.model small

.stack 100

.8086

.data

; Number Of Elements in Array

n dw 000AH

; Elements of Array

a dw 1320h, 3123h, 0EB54h, 4347h, 5605h, 1086h, 4309h, 0DC2h, 12FFh, 4500h

; Answer

ans dw ?

.code

mov ax,@data

mov ds,ax

; Intialize the Counter, Offset and Sum [Intial = 0]

mov cx, n

mov si,offset a

mov ax,0000h

up:     cmp ax,[si]     ; if (arr[i]>ans)

        jnc next

        mov ax,[si]     ; YES Update ans = arr[i]

next:   inc si          ; NO Continue the Loop

        inc si

        loop up

mov ans,ax

mov ax,4c00h

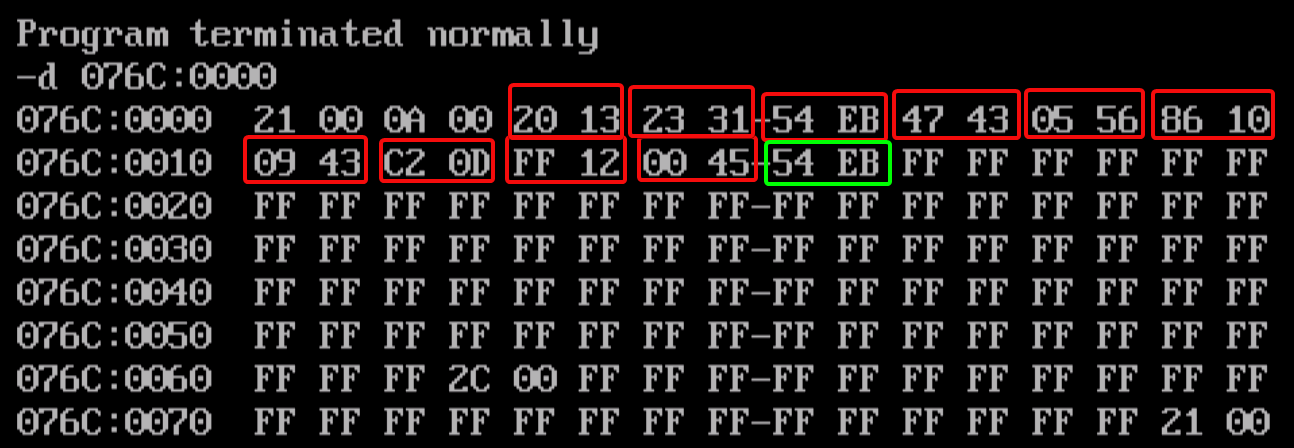
int 21h

end

Input:

{1320h, 3123h, 0EB54h, 4347h, 5605h, 1086h, 4309h, 0DC2h, 12FFh, 4500h}

Output:



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

TASM Code:

.model small

.stack 100

.8086

.data

n db 0AH

;storing in b for refernce

b dw 1410h,3443h,0db54h,4337h,5675h,0086h,4209h,0cd2h,12ffh,4500h

a dw 1410h,3443h,0db54h,4337h,5675h,0086h,4209h,0cd2h,12ffh,4500h

i db ?

tmp db ?

.code

mov ax,@data

mov ds,ax

; Initialize

mov ch,00h

mov cl,n

dec cl

mov si,offset a

mov i,00h

up1:    mov tmp,cl  ;

        mov cl,n

        sub cl,i

        inc i

        dec cl      ; cl = n-i-1

        mov bx, 0000h

up2:    mov ax, [si+bx]

        mov dx, [si+bx+2]

        cmp ax, dx  ; if (ax>=dx) goto next

        jnc next

        mov [si+bx], dx     ; Swap

        mov [si+bx+2], ax

next:   inc bx

        inc bx

        loop up2

        mov cl, tmp

        loop up1

mov ax, 4C00H

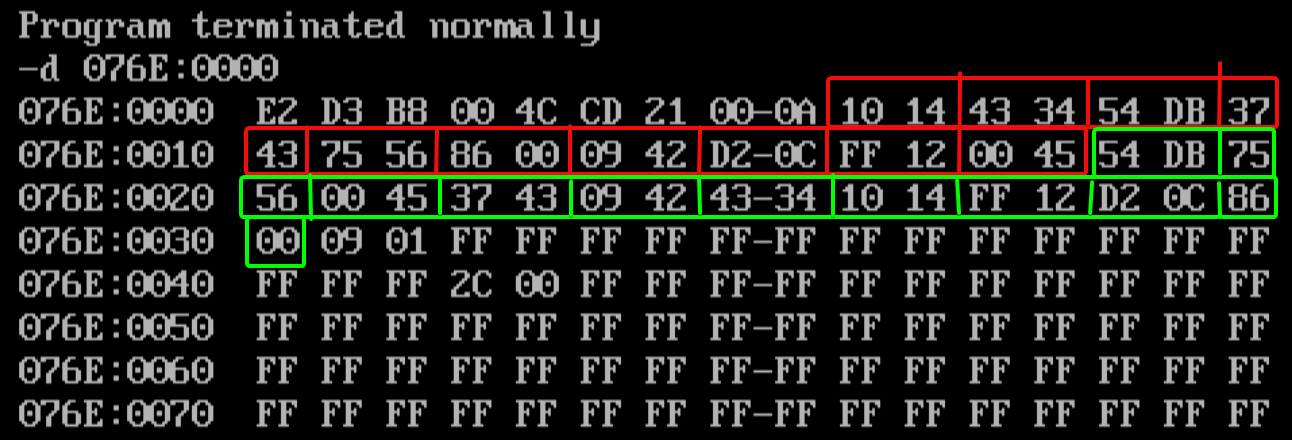
int 21h

end

Input:

{1410h,3443h,0db54h,4337h,5675h,0086h,4209h,0cd2h,12ffh,4500h}

Output:



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

TASM Code:

.model small

.stack 100

.8086

.data

n db ?

ans db 00h

x db ?

y db "$"

cr equ odh

lf equ 0ah

msg1 db "n=$"

msg2 db "Enter Numbers :$"

msg3 db "Enter the number to check its occurences :$"

msg4 db "  $"

p db ?

; PRINT MACRO

print macro msg

mov ah,09h

mov dx,offset msg

int 21h

endm

; READ MACRO

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    ; Both Equal?

    inc ans

    next: inc p ; Increment the Count

    print msg4

    loop up

mov ax,4c00h

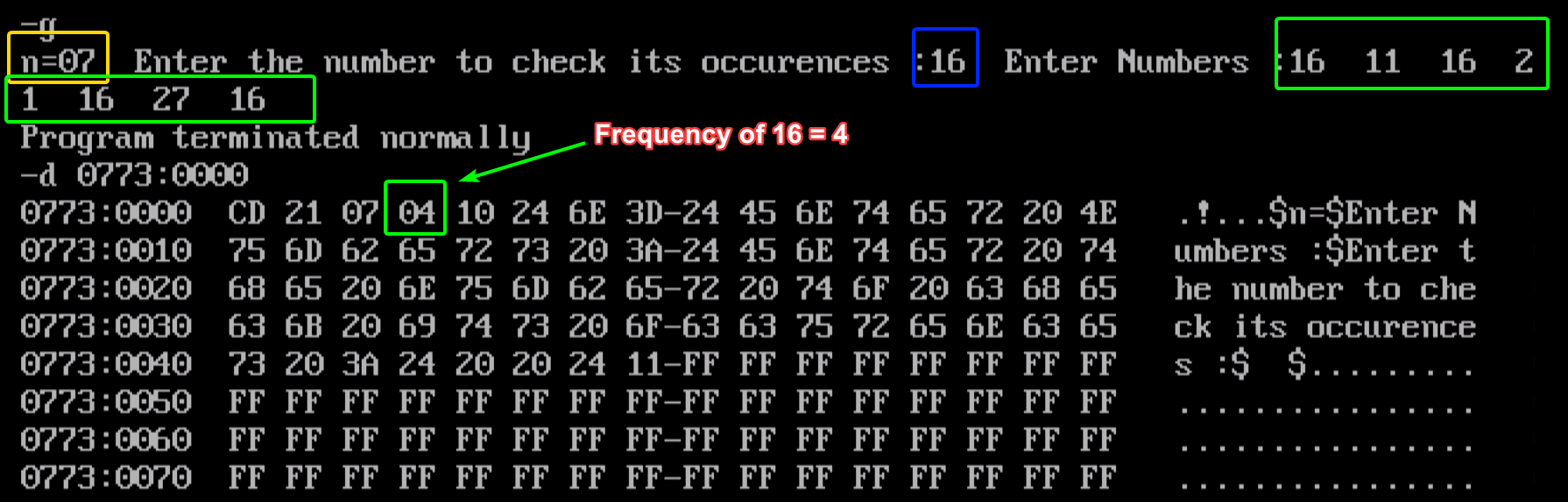
int 21h

end

Input:

{16, 11, 16, 21, 16, 27, 16} -> Frequency of 16 = (?) [(04) H]

Output:



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

TASM Code:

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

.model small

.stack 100

.8086

.data

len db 11h  ; 11h = 17 Length of String

; Source

str1 db "this is a string $"

; Destination

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

.code

mov ax,@data

mov ds,ax

mov es,ax

mov ch,00h

mov cl,len  ; Initialize Counter

; Initialize [SI] & [DI]

mov si,offset str1

mov di,offset str2

; Clear Direction Flag

cld

up: movsb

loop up

; Print the Answer

mov ah,09h

mov dx,offset str2

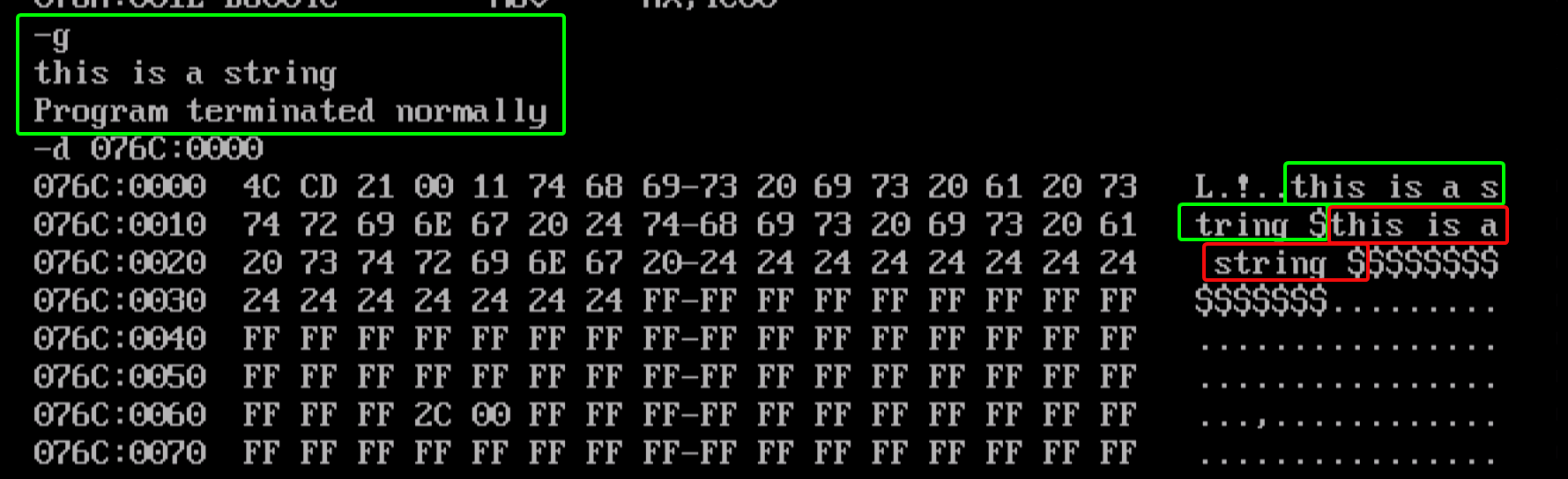
int 21h

mov ax,4c00h

int 21h

end

Output:



6. Write a Program to reverse a given string.

TASM Code:

;Q-(6) Write a Program to reverse a given string.

.model small

.stack 100

.8086

.data

; Length of String

len dw 000CH

; Source String

str1 db " ecnalubma $"

; Reverse String

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

; Print the Reverse String

mov ah,09h

mov dx,offset str2+3

int 21h

mov ax,4c00h

int 21h

end

Input:

String : {“ ecnalubma $”}

Output:



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

TASM Code:

.model small

.stack 100

.8086

.data

len dw 0024h

str1 db "ab CD ef GH ij KLMN opqr STUV wxyz $"

.code

mov ax,@data

mov ds,ax

mov es,ax

mov cx,len

mov si,offset str1

cld

up: mov al,[si]

    cmp al, 20   ; ASCII OF ' ' = 32 = 20h

    jz next

    cmp al, 41h  ; ASCII OF 'A' = 65 = 41h

    jc next

    cmp al,5Ah

    jnc smalla

    add al,20h  ; Add 32 to Get Lower Case

    mov [si],al

    jmp next

    smalla: sub al, 20h     ; Subtract 32 to Get Upper Case

            mov [si], al

    next: inc si

    loop up

; Print the Answer

mov ah,09h

mov dx,offset str1

int 21h

mov ax,4c00h

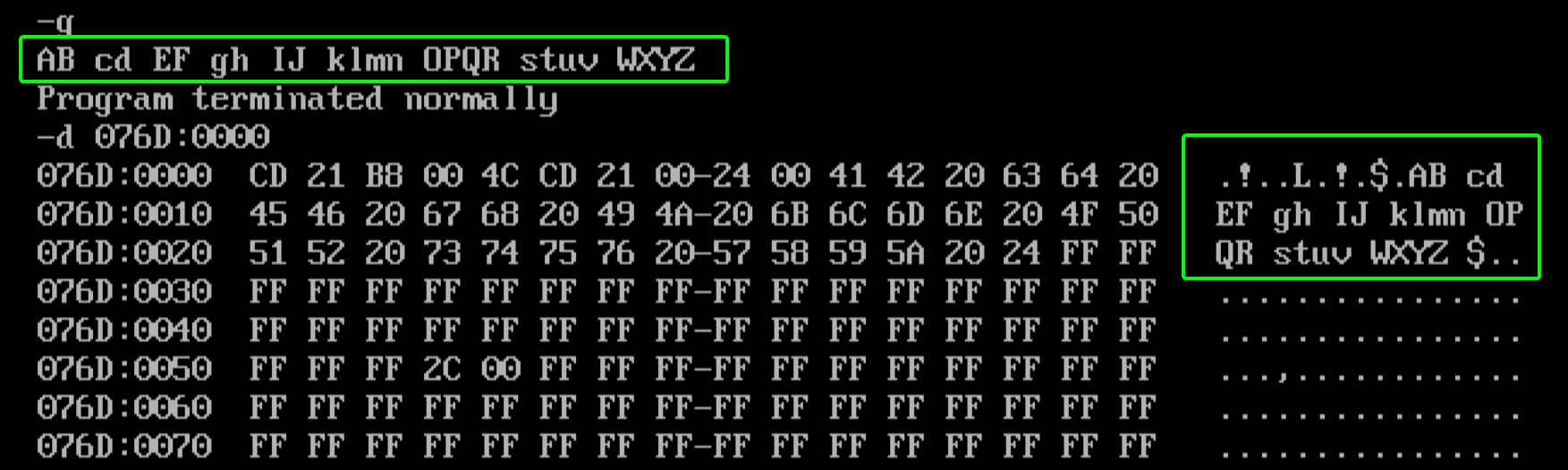
int 21h

end

Input:

{"ab CD ef GH ij KLMN opqr STUV wxyz $"}

Output:



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

TASM Code:

.model small

.stack 100

.8086

.data

len1 db ?

len2 db ?

str1 db "$$$$$$$$$$$$$$$$$$$$$$"

str2 db "$$$$$$$$$$$$$$$$$$$$$$"

final db "$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$"

msg1 db "Enter String : $"

msg3 db "Final String : $"

msg4 db "Enter Length : $"

msg5 db " $"

; PRINT MACRO

print macro msg

mov ah,09h

mov dx,offset msg

int 21h

endm

; READ MACRO

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 [Take Length of String & String Input]

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

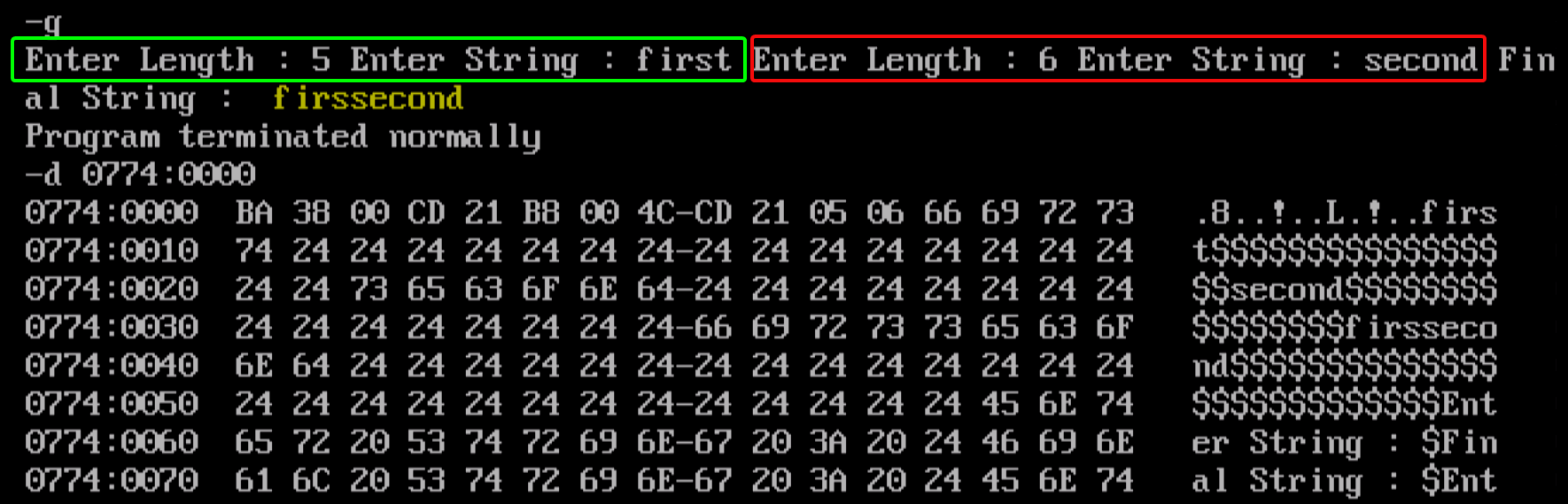
Input:

String 1 : “first”

String 2 : “second”

MergeString : “firstsecond”

Output:



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

TASM Code:

.model small

.stack 100

.8086

.data

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

print macro msg

mov ah,09h

mov dx,offset msg

int 21h

endm

; READ MACRO

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

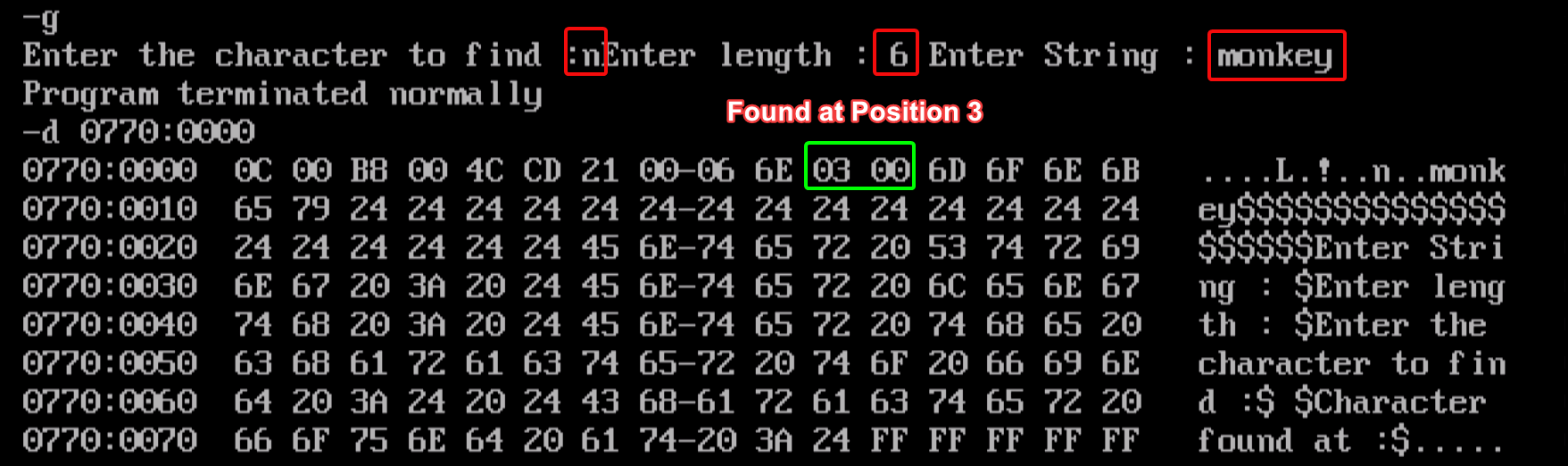
int 21h

end

Input:

‘n’ in Monkey = 3rd Place

Output:



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

TASM Code:

.model small

.stack 100

.8086

.data

;PRINT MACRO

print macro msg

mov ah,09h

mov dx,offset msg

int 21h

endm

; READ MACRO

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

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 " $"

.code

mov ax,@data

mov ds,ax

mov es,ax

print msg2  ; Enter Character

; Take Character input and Store in AL

mov ah, 01h

int 21h

mov char, al

; Take String Input

read str1

print msg5

; DI -> Pointing to Str1 [Intialize DI]

mov di,offset str1

; Intialize Counter

mov cl,len1

mov ch,00h

mov al,char

up: mov ah,[di]

    cmp ah,al   ; Compare if the Character is Same?

    jnz skip

    inc ans     ; Increment the Frequency of character

    skip: inc di

    loop up

; HLT

mov ax,4c00h

int 21h

end

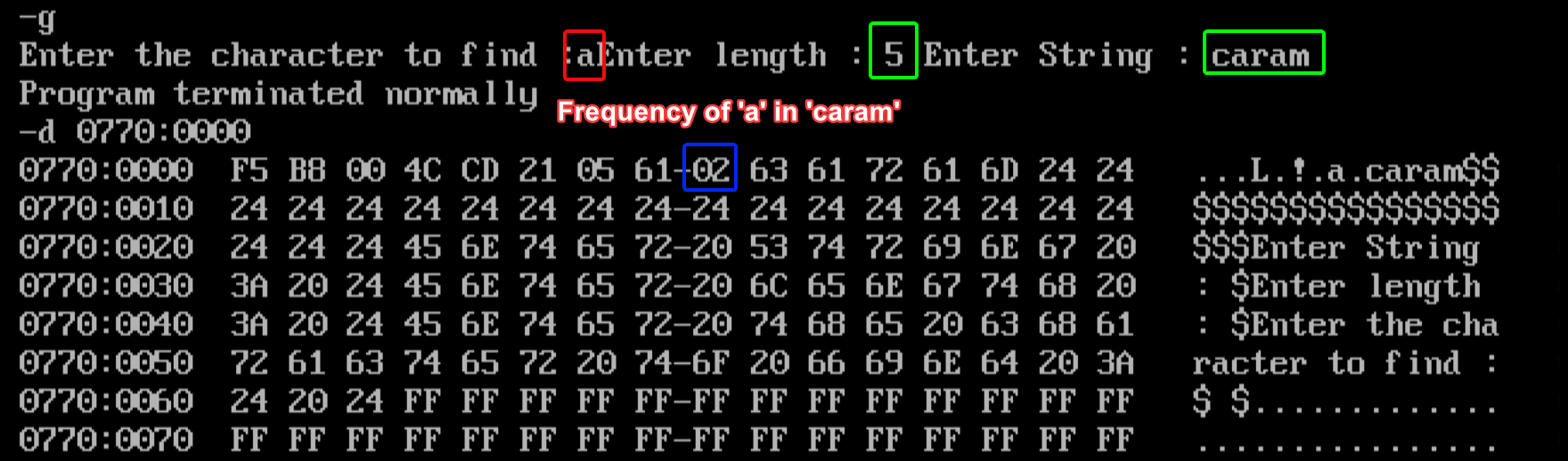
Input:

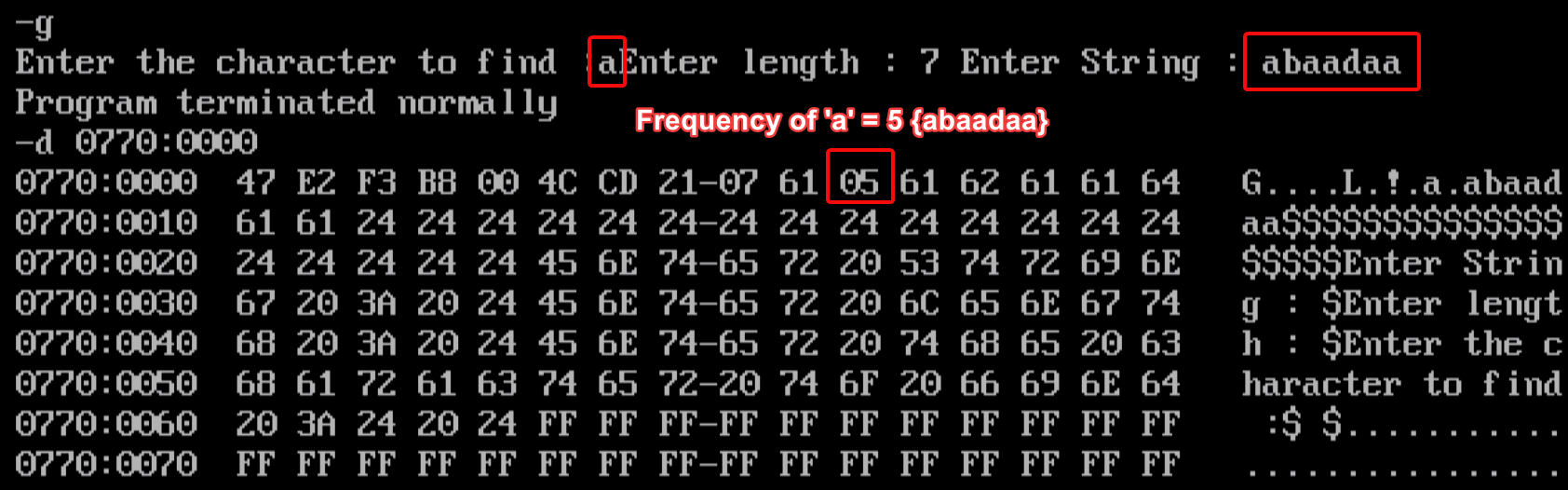
Character = ‘a’

String = ‘caram’

Frequency of ‘a’ = 2

Output:





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

TASM Code:

.model small

.stack 100

.8086

.data

len1 db 0Eh

; Change it to 06H <- Mango

len2 db 07h

ans db 00h

str1 db "assassination$"

subsr db "nation$"

; subsr db "mango$" ; NOT FOUND CASE

msg5 db " $"

msg1 db "substring found $"

msg2 db "substring not found$"

temp dw ?

; PRINT MACRO

print macro msg

mov ah,09h

mov dx,offset msg

int 21h

endm

.code

mov ax,@data

mov ds,ax

mov es,ax

mov di,offset str1

mov si,offset subsr

mov ch,00h

mov cl,len1

up: mov si,offset subsr

    mov al,[si]

    repnz scasb

    cmp cx,00h

    jz terminate

    mov temp,di

    push cx

    mov cl,len2

    dec cl

    inc si

    check:  mov al,[si]

            cmp al,[di]

            jnz skip

            inc si

            inc di

    loop check

    print msg1

    jmp terminate

    skip: pop cx

    mov di,temp

    loop up

; HLT

terminate: mov ax,4c00h

int 21h

end

Input:

String: “assassination$”

Sub-String: “nation$”

Output:



Input:

String: “assassination$”

Sub-String: “mango$”

Output:



SUBMITTED BY:

BHAGYA VINOD RANA

[***U19CS012***]