Samiul Karim  
201-15-3041

Answer to the question no 1.

include 'emu8086.inc'

.model small

.stack 100h

.code

main proc

print "Enter the number: "

mov ah,1

int 21h

mov bl,al

printn

cmp bl,32h

je even

cmp bl,34h

je even

cmp bl,36h

je even

cmp bl,38h

je even

jne odd

odd:

mov cx,5

mov ah,2

mov dl,31h

l1:

int 21h

inc dl

loop l1

jmp ai

even:

mov cx,4

mov ah,2

mov dl,36h

l2:

int 21h

inc dl

loop l2

jmp ai

ai:

printn

print "Enter another number: "

mov ah,1

int 21h

mov cl,al

printn

cmp cl,bl

je exit

print "1 3 5 7 9"

exit:

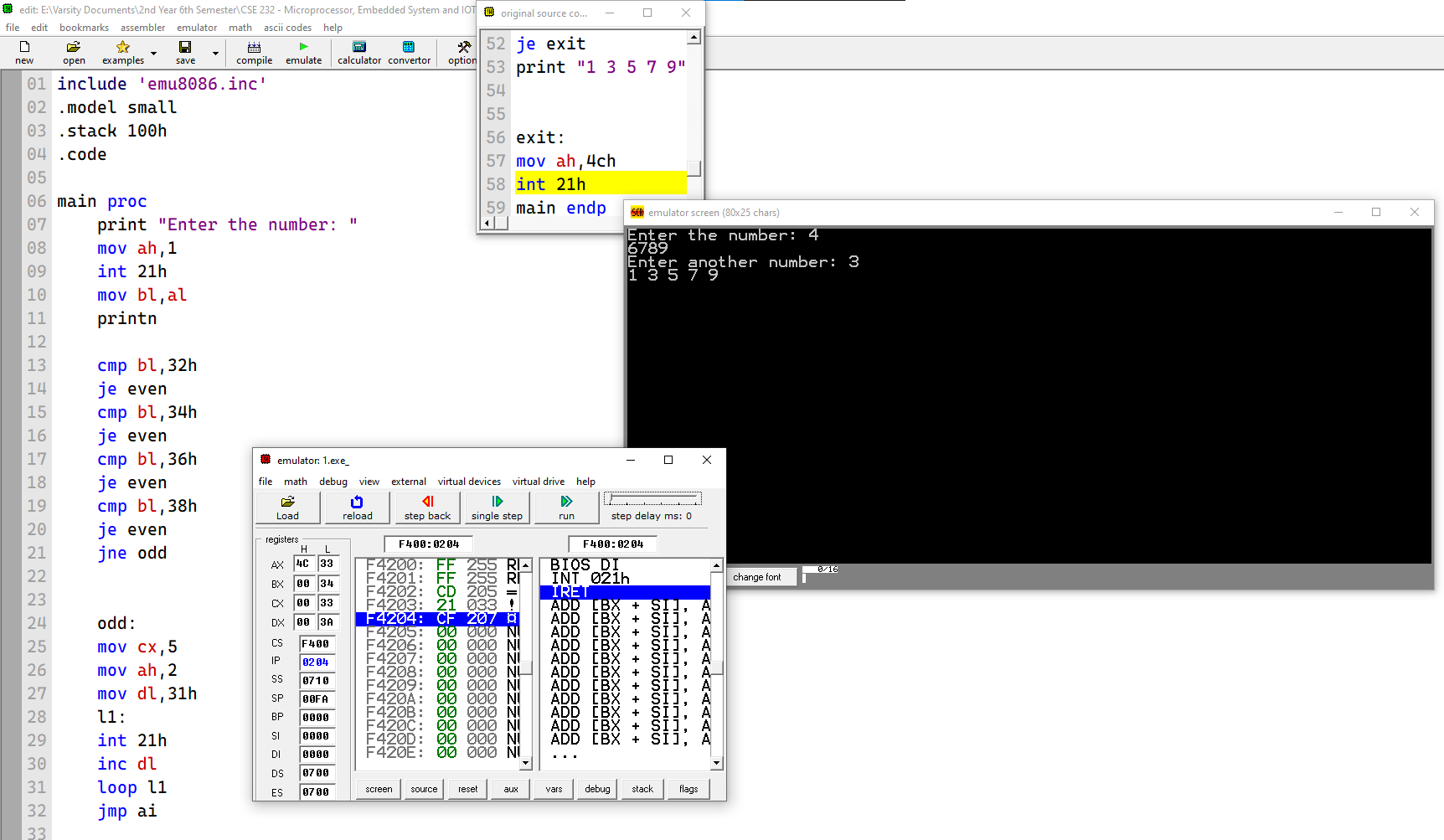
mov ah,4ch

int 21h

main endp

end main

Screenshot:



Answer to the question no 3.

include 'emu8086.inc'

.model small

.code

.stack 100h

main proc

print "First Number: "

mov ah,1

int 21h

mov bl,al

printn

print "Second Number: "

int 21h

mov bh,al

printn

print "Third Number: "

int 21h

mov cl,al

printn

print "Forth Number: "

int 21h

mov ch,al

sub cl,ch

sub bh,cl

sub bl,bh

add bl,30h

printn

print "Result: "

mov ah,2

mov dl,bl

int 21h

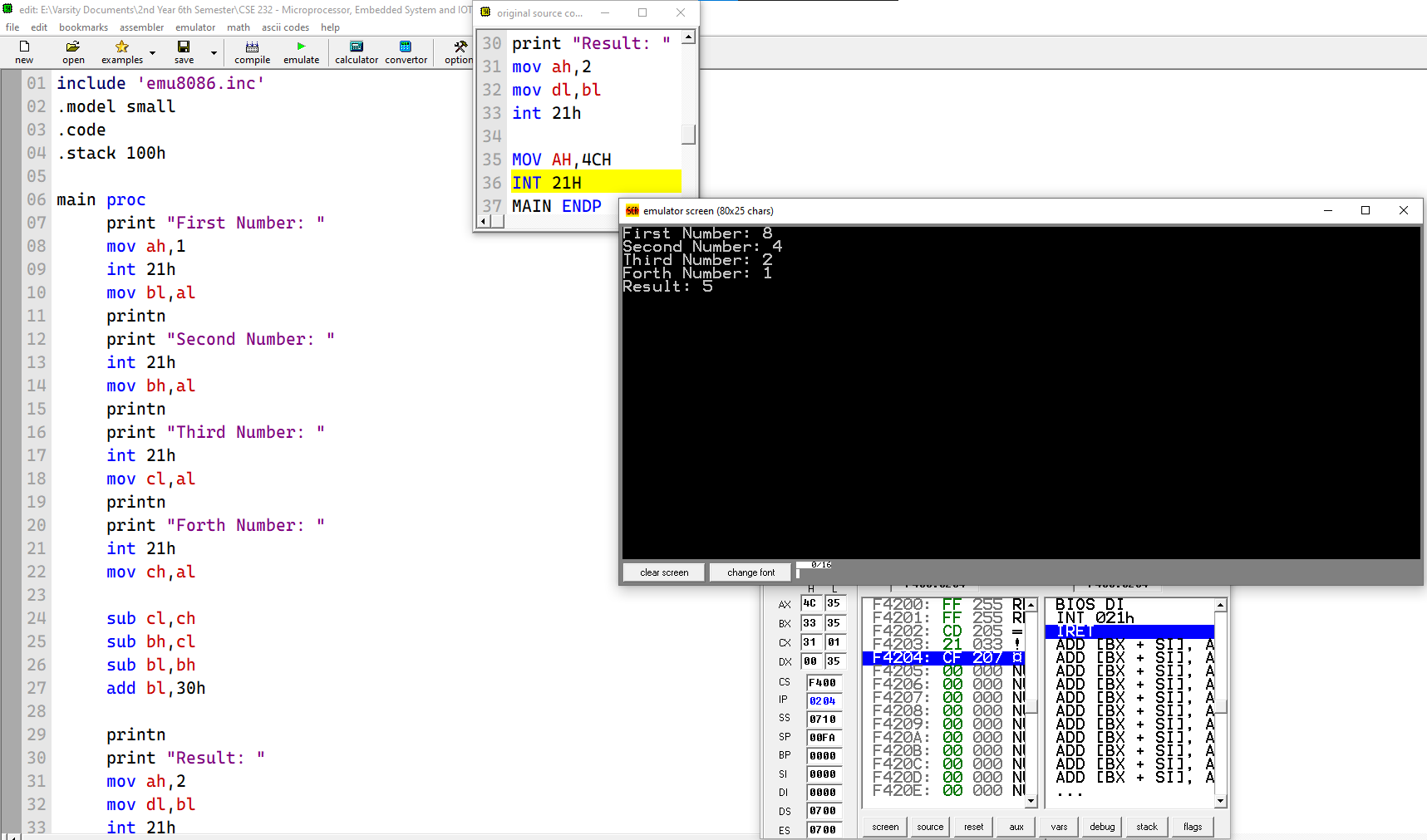
MOV AH,4CH

INT 21H

MAIN ENDP

END MAIN

Screenshot:



Answer to the question no 5.

include 'emu8086.inc'

.model small

.stack 100h

.data

msg1 db "enter the 1st digit: $"

msg2 db "enter the 2nd digit: $"

msg3 db "enter the highest number of loops of the 2nd digit: $"

var db ?

a db ?

b db ?

c db ?

.code

main proc

mov ax, @data

mov ds,ax

lea dx,msg1

mov ah,9

int 21h

mov ah,1

int 21h

sub al,48

mov a,al

printn

lea dx,msg2

mov ah,9

int 21h

mov ah,1

int 21h

sub al,48

mov b,al

printn

lea dx,msg3

mov ah,9

int 21h

mov ah,1

int 21h

sub al,48

mov c,al

printn

mov var, al

mov cx,0

mov cl,c

mov bl,c

first:

cmp bl, 0

je exit

mov ah,2

mov al,a

add al,48

mov dl,al

int 21h

mov cx,bx

second:

mov ah,2

mov al,b

add al,48

mov dl,al

int 21h

loop second

dec bl

loop first

exit:

mov ah,4ch

int 21h

main endp

end main

.model small

.stack 100h

.data

string db 100 dup(?)

msg1 db "enter a string: $"

msg2 db 0dh, 0ah, "no. of characters: $"

characters dw 0

.code

main proc

mov ax, @data

mov ds, ax

mov es, ax

mov ah, 9

lea dx, msg1

int 21h

lea di, string

mov ah, 1

read:

int 21h

cmp al, 0dh

je endofstring

stosb

jmp read

endofstring:

mov al, "$"

stosb

xor bx, bx

count:

cmp [string+bx], '$'

je exit

inc bx

inc characters

jmp count

exit:

mov ah, 9

lea dx, msg2

int 21h

mov ah, 2

mov dx, characters

add dx, 30h

int 21h

main endp

end main

Screenshot:

