TASK 1:

[org 0x0100] jmp start

x: dw 5

y: dw 10

length: dw 8 clrscr: push es push ax

push di

shl word[length],1 mov ax, 0xb800

mov es, ax

mov ax,[y] mov bx, 80; mul bx add ax,[x] shl ax,1

mov di, ax

add word[length], di

nextloc: mov word [es:di], 0x072a

add di, 2

cmp di, [length]

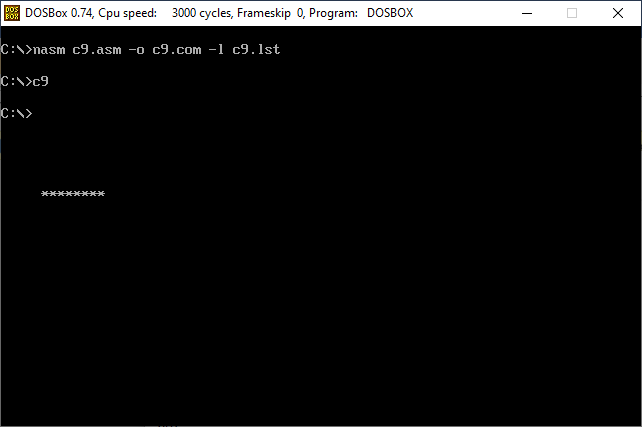
jne nextloc

pop di pop ax pop es ret

start:

call clrscr

mov ax, 0x4c00 ; terminate program int 0x21

SCS:

**TASK 2:**

[org 0x0100] jmp start

l1: db ' \* ' l2: db ' \*\*\* ' l3: db ' \*\*\*\*\* '

l4: db '\*\*\*\*\*\*\*' length: dw 7 clrscr:

push es push ax push di

mov ax, 0xb800

mov es, ax ; point es to video base mov di, 0 ; point di to top left column

nextloc: mov word [es:di], 0x0720 ; clear next char on screen add di, 2 ; move to next screen location

cmp di, 4000 ; has the whole screen cleared jne nextloc ; if no clear next position

pop di pop ax pop es ret

print: push bp mov bp, sp

mov ax, 0xb800

mov es, ax

mov di, [bp+6]

mov si, [bp+8]

mov cx, [bp+4]

mov ah, 0x07

nextchar: mov al, [si]

mov [es:di], ax

add di, 2

add si, 1

loop nextchar

pop bp

ret 4

start:

call clrscr mov bx,l1 push bx mov di,0 push di

mov bx ,[length] push bx

call print

mov bx ,l2 push bx mov di,160 push di

mov bx ,[length] push bx

call print mov bx, l3 push bx mov di,320 push di

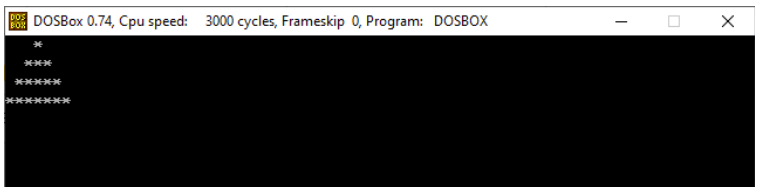
mov bx ,[length] push bx

call print

mov bx , l4 push l4 mov di,480 push di

mov bx ,[length] push bx

call print mov ah,4ch int 21h SCS:

****