**LAB 12**

**Question:**  
  
1. Print a string vertically in the middle of screen and then scroll it vertically downwards, such that if a letter exits the screen from bottom it reappears from the top.

**CODE:**

org 100h

.data

n db 'Hashim$'

check dw (38-1)\*2

.code

main proc

mov bp,(10-1)\*160+(38-1)\*2

mov si,bp

top1:

call print

call downshift

jmp top1

ret

Endp main

print proc

lea di,n

mov ax,0xb800

mov es,ax

mov cl,6

top:

mov ah,0x07

mov al,[di]

mov es:si,ax

ADD si,160

inc di

mov bx,(26-1)\*160+(38-1)\*2

mov ax,si

xor dx,dx

div bx

mov si,dx

cmp ax,0

je down

add si,check

down:

loop top

ret

Endp print

downshift proc

xor dx,dx

mov bx,(25-1)\*160+(38-1)\*2

cmp bp,bx

jl adding

mov bp,(0-1)\*160+(38-1)\*2

adding:

add bp,160

mov si,bp

mov ah,6

mov al,0

mov ch,0

mov cl,0

mov dh,24

mov dl,79

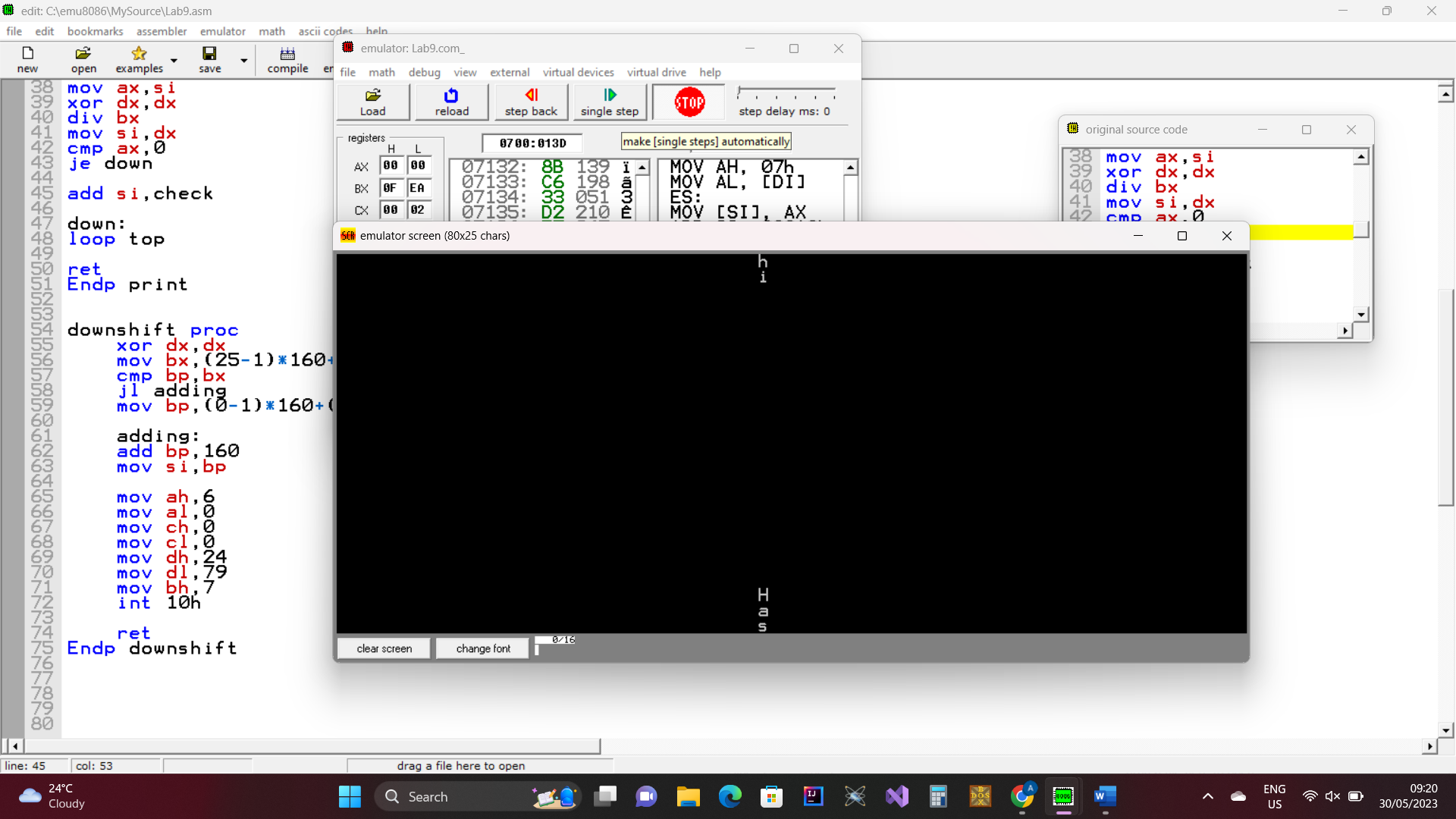
mov bh,7

int 10h

ret

Endp downshift

**OUTPUT:**



2. Hook keyboard interrupt such that the scrolling toggles on subsequent presses of button 'p'

**CODE:**

org 100h

.data

n db 'Hashim$'

check dw (38-1)\*2

flag db 0

.code

main proc

mov ax,0

mov es,ax

mov si,9\*4

cli

mov es:si, offset kbisr

mov es:si+2,cs

sti

top1:

cmp flag,1

jne skip

call print

call downshift

skip:

jmp top1

ret

Endp main

kbisr proc

push ax

in al,0x60

mov flag,0

cmp al,0x19

jne down

mov flag,1

down:

mov al,0x20

out 0x20,al

pop ax

iret

print proc

lea di,n

mov ax,0xb800

mov es,ax

mov cl,6

top:

mov ah,0x07

mov al,[di]

mov es:si,ax

ADD si,160

inc di

mov bx,(26-1)\*160+(38-1)\*2

mov ax,si

xor dx,dx

div bx

mov si,dx

cmp ax,0

je down1

add si,check

down1:

loop top

ret

Endp print

downshift proc

xor dx,dx

mov bx,(25-1)\*160+(38-1)\*2

cmp bp,bx

jl adding

mov bp,(0-1)\*160+(38-1)\*2

adding:

add bp,160

mov si,bp

mov ah,6

mov al,0

mov ch,0

mov cl,0

mov dh,24

mov dl,79

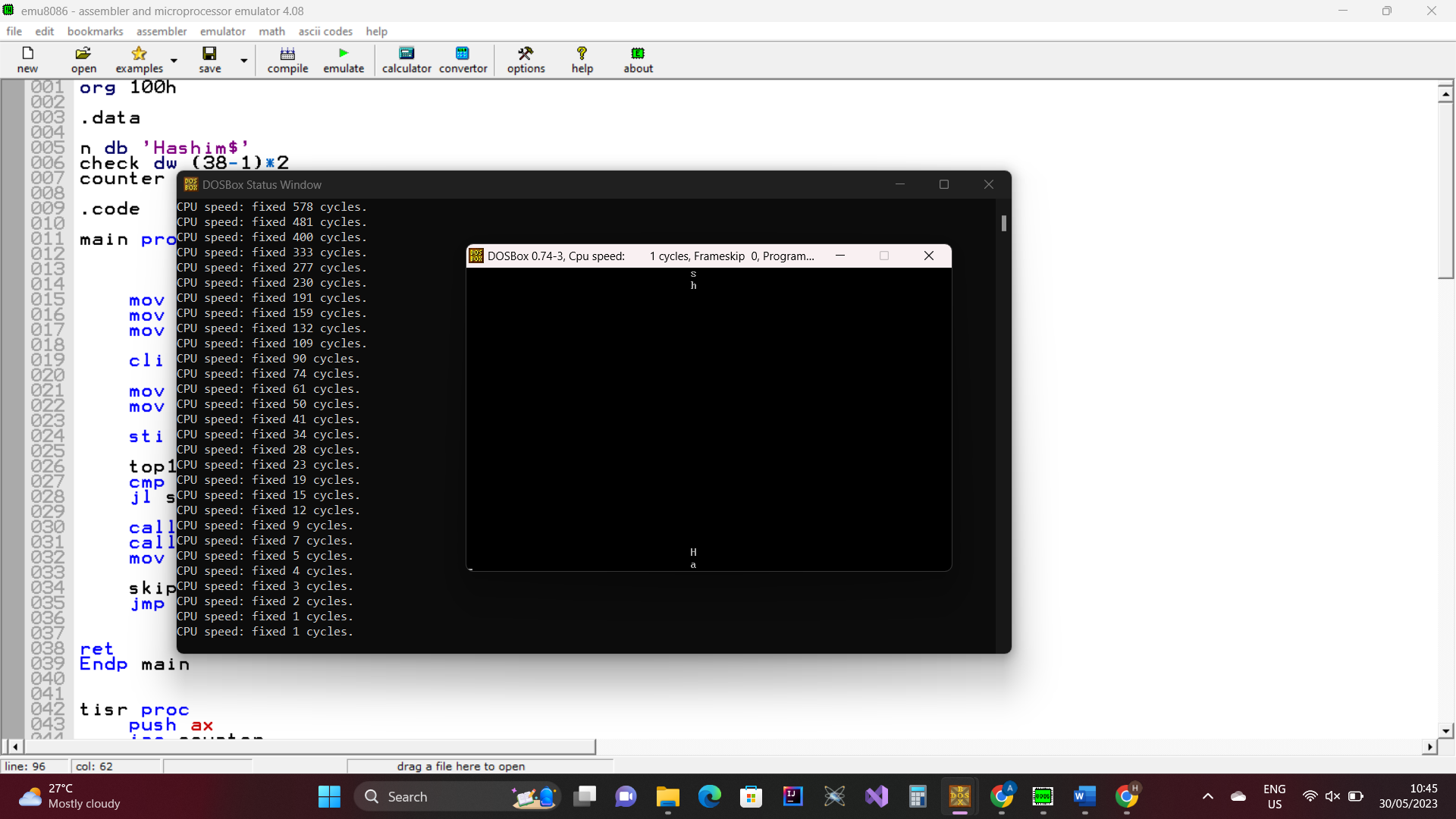
mov bh,7

int 10h

ret

Endp downshift

**OUTPUT:**



3. Hook timer interrupt in addition to KB interrupt so that each scroll of string occurs after half a second.

**CODE:**

org 100h

.data

n db 'Hashim$'

check dw (38-1)\*2

counter db 0

flag db 0

.code

main proc

mov ax,0

mov es,ax

mov si,9\*4

cli

mov es:si, offset tisr

mov es:si+2,cs

sti

mov si,(13-1)\*160+(38-1)\*2

mov bp,(13-1)\*160+(38-1)\*2

top1:

cmp counter,9

jl skip

cmp flag,1

jne skip

call print

call downshift

mov counter,0

mov flag,0

skip:

jmp top1

ret

Endp main

tisr proc

push ax

inc counter

in al,0x60

cmp al,0x19

jne down

mov flag,1

down:

mov al,0x20

out 0x20,al

pop ax

iret

print proc

mov ax,3

int 10h

lea di,n

mov ax,0xb800

mov es,ax

mov cl,6

top:

mov ah,0x07

mov al,[di]

mov es:si,ax

ADD si,160

inc di

mov bx,(26-1)\*160+(38-1)\*2

mov ax,si

xor dx,dx

div bx

mov si,dx

cmp ax,0

je down1

add si,check

down1:

loop top

ret

Endp print

downshift proc

xor dx,dx

mov bx,(25-1)\*160+(38-1)\*2

cmp bp,bx

jl adding

mov bp,(0-1)\*160+(38-1)\*2

adding:

add bp,160

mov si,bp

ret

Endp downshift

**OUTPUT:**

A picture containing text, screenshot, software, multimedia software

Description automatically generated