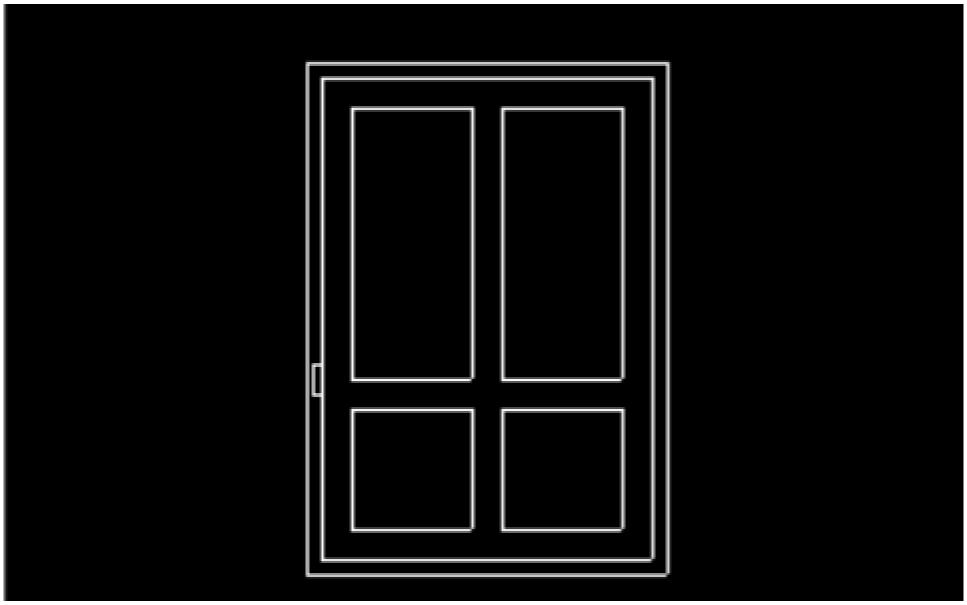
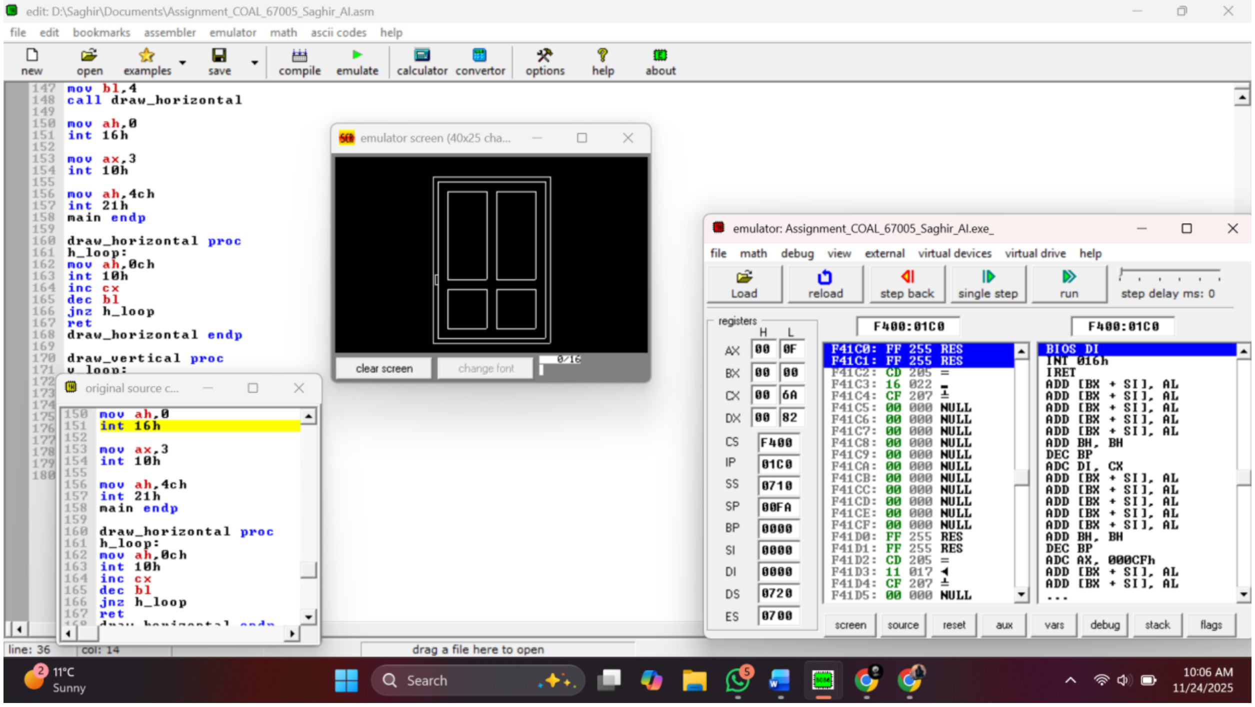
Assignment COAL – Saghir Ali – 67005 – AI

CODE OUTPUT:





edit: D:\Saghir\Documents\Assignment_COAL_67005_Saghir_Al.asm

```
mov bl,4
call draw_horizontal

mov ah,0
int 16h

mov ax,3
int 10h

mov ah,4ch
int 21h
main endp

draw_horizontal proc
h_loop:
mov ah,0ch
int 10h
inc cx
dec bl
jnz h_loop
ret
draw_horizontal endp

draw_vertical proc
v_loop:
```

original source c...

```
mov ah,0
int 16h

mov ax,3
int 10h

mov ah,4ch
int 21h
main endp

draw_horizontal proc
h_loop:
mov ah,0ch
int 10h
inc cx
dec bl
jnz h_loop
ret
```

line: 36 col: 14

emulator: Assignment_COAL_67005_Saghir_Al.exe_

registers	H	L
AX	00	0F
BX	00	00
CX	00	6A
DX	00	82
CS	F400	
IP	01C0	
SS	0710	
SP	00FA	
BP	0000	
SI	0000	
DI	0000	
DS	0720	
ES	0700	

```
F400:01C0

F41C0: FF 255 RES
F41C1: FF 255 RES
F41C2: CD 205 =
F41C3: 16 022 ▬
F41C4: CF 207 ╧
F41C5: 00 000 NULL
F41C6: 00 000 NULL
F41C7: 00 000 NULL
F41C8: 00 000 NULL
F41C9: 00 000 NULL
F41CA: 00 000 NULL
F41CB: 00 000 NULL
F41CC: 00 000 NULL
F41CD: 00 000 NULL
F41CE: 00 000 NULL
F41CF: 00 000 NULL
F41D0: FF 255 RES
F41D1: FF 255 RES
F41D2: CD 205 =
F41D3: 11 017 ◄
F41D4: CF 207 ╧
F41D5: 00 000 NULL
```

```
F400:01C0

BIOS DI
INT 016h
IRET
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD BH, BH
DEC BP
ADC DI, CX
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD BH, BH
DEC BP
ADC AX, 000CFh
ADD [BX + SI], AL
ADD [BX + SI], AL
...
```

step delay ms: 0

file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
emulator screen (40x25 cha...
clear screen change font
file math debug view external virtual devices virtual drive help
Load reload step back single step run
screen source reset aux vars debug stack flags
drag a file here to open
11°C Sunny
Search
10:06 AM
11/24/2025

CODE:

```
.model small

.stack 100h

.data

.code

main proc

mov ax,@data

mov ds,ax


mov al,13h

mov ah,0

int 10h


mov al,15
mov cx,100
mov dx,20
mov bl,120
call draw_horizontal

mov cx,100
mov dx,190
mov bl,120
call draw_horizontal

mov cx,100
mov dx,20
mov bl,170
call draw_vertical

mov cx,220
mov dx,20
mov bl,170
call draw_vertical

mov cx,105
mov dx,25
mov bl,110
call draw_horizontal

mov cx,105
mov dx,185
mov bl,110
call draw_horizontal

mov cx,105
mov dx,25
mov bl,160
call draw_vertical

mov cx,215
mov dx,25
mov bl,160
call draw_vertical

mov cx,115
mov dx,35
mov bl,40
call draw_horizontal

mov cx,115
mov dx,125
mov bl,40
call draw_horizontal

mov cx,115
mov dx,35
mov bl,90
call draw_vertical
```

```
mov cx,155
mov dx,35
mov bl,90
call draw_vertical

mov cx,165
mov dx,35
mov bl,40
call draw_horizontal

mov cx,165
mov dx,125
mov bl,40
call draw_horizontal

mov cx,165
mov dx,35
mov bl,90
call draw_vertical

mov cx,205
mov dx,35
mov bl,90
call draw_vertical

mov cx,115
mov dx,135
mov bl,40
call draw_horizontal

mov cx,115
mov dx,175
mov bl,40
call draw_horizontal

mov cx,115
mov dx,135
mov bl,40
call draw_vertical

mov cx,155
mov dx,135
mov bl,40
call draw_vertical

mov cx,165
mov dx,135
mov bl,40
call draw_horizontal

mov cx,165
mov dx,175
mov bl,40
```

```
call draw_horizontal

mov cx,165
mov dx,135
mov bl,40
call draw_vertical

mov cx,205
mov dx,135
mov bl,40
call draw_vertical

mov cx,102
mov dx,120
mov bl,10
call draw_vertical

mov cx,102
mov dx,120
mov bl,4
call draw_horizontal

mov cx,102
mov dx,130
mov bl,4
call draw_horizontal
```

```
mov ah,0
int 16h

mov ax,3
int 10h

mov ah,4ch
int 21h
main endp

draw_horizontal proc
h_loop:
mov ah,0ch
int 10h
inc cx
dec bl
jnz h_loop
ret
draw_horizontal endp

draw_vertical proc
v_loop:
mov ah,0ch
int 10h
inc dx
dec bl
jnz v_loop
```

```
ret
draw_vertical endp


end main
```