# TASK 1:

org 0x0100

jmp start

; Keyboard interrupt service routine

kbisr:

push ax

push es

mov ax, 0xb800

mov es, ax ; point es to video memory

in al, 0x60 ; read a char from the keyboard port

; Check if the key is a vowel

cmp al, 0x12

je vowel

cmp al, 0x16

je vowel

cmp al, 0x17

je vowel

cmp al, 0x18

je vowel

cmp al, 0x1E

je vowel

jmp consonant ; If not a vowel, check if it's a consonant

vowel:

mov byte [es:3840], 'V' ; Display 'V' at the top left

jmp done ; Leave interrupt routine

consonant:

mov byte [es:3840], 'C' ; Display 'C' at the top left

jmp done

done:

mov al, 0x20

out 0x20, al ; send EOI to PIC

pop es

pop ax

iret

start:

xor ax, ax

mov es, ax ; point es to IVT base

cli ; disable interrupts

mov word [es:9\*4], kbisr ; store offset at n4

mov [es:9\*4+2], cs ; store segment at n4+2

sti ; enable interrupts

l1:

jmp l1 ; infinite loop

# OUTPUT:

# 

# TASK 2:

org 0x0100

jmp start

; Keyboard interrupt service routine

kbisr:

push ax

push es

mov ax, 0xb800

mov es, ax ; point es to video memory

in al, 0x60 ; read a char from keyboard port

cmp al, 0x3A ; is the key CAPS Lock pressed

je CapsLockOn

cmp al, 0xBA ; is the key CAPS Lock released

je CapsLockOff

jmp nomatch ; leave interrupt routine

CapsLockOn:

mov byte [es:3842], 'O' ; Display 'ON' at top left

mov byte [es:3844], 'N'

jmp nomatch ; leave interrupt routine

CapsLockOff:

mov byte [es:3843], 'O' ; Display 'OFF' at top left

mov byte [es:3844], 'F'

mov byte [es:3846], 'F'

jmp nomatch ; leave interrupt routine

nomatch:

mov al, 0x20

out 0x20, al ; send EOI to PIC

pop es

pop ax

iret

start:

xor ax, ax

mov es, ax ; point es to IVT base

cli ; disable interrupts

mov word [es:9\*4], kbisr ; store offset at n4

mov [es:9\*4+2], cs ; store segment at n4+2

sti ; enable interrupts

l1:

jmp l1 ; infinite loop

# OUTPUT:



