Solutions to Assignment 6

Question 1

```
;
; ELEC 464, Assignment 6
; Example of IBM PC keyboard ISR
; Ed Casas, 96/10/23
;

; keyboard interrupt vector location

kbvec   equ     4*(8+1)

; standard directives for DOS .com files

code    segment public
    assume  cs:code,ds:code
    org     100h
start:

; print name and student number

    mov     bx,offset msg ; offset of string
    mov     ah,02H        ; print function
print1:
    mov     dl,[bx]       ; load a character
    or      dl,dl         ; set PSW flags
    jz      print2        ; stop if zero
    int     21H           ; call DOS
    inc     bx            ; point to next
    jmp     print1        ; and loop back
print2:

; set flag to zero

    mov     al,0
    mov     flag,al

; clear ES (for access to interrupt vectors)

    mov     ax,0
    mov     es,ax

; save old interrupt vector

    mov     ax,es:[kbvec]
    push    ax
    mov     ax,es:[kbvec+2]
    push    ax

; set up new keyboard interrupt vector

    cli                   ; disable interrupts
    mov     ax,offset kbisr
    mov     es:[kbvec],ax
    mov     ax,cs
    mov     es:[kbvec+2],ax
    sti                   ; [re-]enable interrupts

; loop until ISR sets flag

loop:
    mov     al,flag
    or      al,al
    jz      loop

; restore old interrupt vector and return to DOS

    cli                   ; prevent interrupts
    pop     ax     ; during change-over
    mov     es:[kbvec+2],ax
    pop     ax
    mov     es:[kbvec],ax
    sti                   ; re-enable interrupts

    int     20h

; variables for main program

flag    db      ?
msg     db     'Ed Casas, 12345678',13,10,0

; The (temporary) keyboard ISR

kbisr:
    mov     cs:tmpax,ax  ; save working registers
    mov     ax,ds
    mov     cs:tmpds,ax

    mov     ax,cs   ; set up DS
    mov     ds,ax

    in      al,60H  ; get the keyboard scan code
    cmp     al,81H  ; is it ESC key release?
    jnz     isr1    ; if not, ignore it
    mov     al,1    ; otherwise, set flag=1
    mov     flag,al
isr1:
    mov     al,20h  ; send EOI to PIC to
    out     20h,al  ; re-enable interrupts
    mov     ax,tmp  ; restore working register
    iret            ; return from ISR

; variables for ISR

tmpax   dw      ?
tmpds   dw      ?

; stack for ISR

    dw      10 dup ?
stack:


code ends
     end     start
```