Solution to Assignment 4
PC ISR

Question 1

```
;
; ELEC 379, Assignment 4
; Ed Casas, 98/10/12
;
; Example of IBM PC keyboard ISR
;
; The main program sets a 'count' variable to 0
; and waits until the keyboard ISR increments it
; to 2. The ISR increments the count variable
; whenever it sees the scan code for the release
; of the '1' key.
;

; 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     dx,offset msg   ; dx=offset of string
        mov     ah,09H          ; ah=print-string code
        int     21H             ; call DOS

; set count to zero

        mov     al,0
        mov     count,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 count to 2

loop:   mov     al,count
        cmp     al,2
        jnz     loop

; restore old interrupt vector and return to DOS

        cli                     ; prevent interrupts
        pop     ax              ; while they are changed
        mov     es:[kbvec+2],ax
        pop     ax
        mov     es:[kbvec],ax
        sti                     ; re-enable interrupts

        int     20h

; variables for main program

count   db      ?
msg     db     'Ed Casas, 12345678',13,10,'$'

; The (temporary) keyboard ISR

kbisr:  push    ax      ; save working register
        in      al,60H  ; get the keyboard scan code
        cmp     al,82H  ; is it a '1' key release?
        jnz     isr1    ; if not, ignore it
        mov     al,count ; otherwise, increment count
        add     al,1
        mov     count,al
isr1:   mov     al,20h  ; send EOI to PIC to re-enable
        out     20h,al  ; interrupts
        pop     ax
        iret            ; return from ISR

code    ends
        end     start
```