Solution for Assignment 2

The fastest way to develop a working assembly program is to work from a version written in a high-level language. A suitable C solution to this assingment is:

```
/* sol2.c - C language solution */

#include <stdio.h>
#include <string.h>

main()
{
  char c, buf[10] ;
  int i, n ;

  /* read until '$', saving up to 10 vowels */

  for ( n=0 ; ( c = getchar() ) != '$' ; ) {
    if ( index("aeiouyAEIOUY", c ) && n < 10 ) {
      buf[n++] = c ;
    }
  }

  putchar ( '\n' ) ;

  /* print the vowels in upper case */

  for ( i=0 ; i<n ; i++ ) {
    putchar ( toupper ( buf[i] ) ) ;
    putchar ( ' ' ) ;
  }

  putchar ( '\n' ) ;
}
```

Once the C solution has been tested, it can be translated (by a compiler or by hand) into assembly language:

```
; sol2.asm - Solution to Assignment 2
;   for EECE 379 1999/2000 Term 2
; This proogram reads a string terminated with
; '$', saves the vowels and prints them in
; upper case with spaces between them.
; Ed Casas, February 21, 2000

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

start:

; initialization

        mov     ax,0    ; vowels in buffer = 0
        mov     n,ax

; read characters until '$', saving up to 10 vowels

inloop:
        call    getc    ; read a character
        mov     c,al    ; save it
        cmp     al,'$'  ; stop reading on a '$'
        je      indone

        call    isvwl   ; check if it's a vowel
        cmp     ax,1
        jne     skipchr ; ignore it if it's not

        mov     ax,n    ; check if buffer
        cmp     ax,10   ;    already full
        jge     skipchr ; ignore the character
                        ;    if it is

        mov     bx,offset buf
        mov     ax,n    ; save it as the n'th character
        add     bx,ax
        mov     al,c
        mov     [bx],al

        mov     ax,n
        add     ax,1    ; increment character count
        mov     n,ax

skipchr:                ; loop back
        jmp     inloop

indone:

; start a new line

        mov     al,0dh  ; print CR
        call    putc
        mov     al,0ah  ; print LF
        call    putc

; print vowels in buffer, in uppper case,
; with spaces in-between

        mov     ax,0    ; start index at 0
        mov     i,ax

outloop:
        mov     ax,i    ; check if at end
        cmp     ax,n
        jge     outdone

        mov     bx,offset buf   ; get i'th character
        add     bx,i
        mov     al,[bx]
        call    toupper ; convert to U/C and print it
        call    putc

        mov     al,' '  ; print a space
        call    putc
```

```
        mov     ax,i    ; increment index
        add     ax,1
        mov     i,ax

        jmp     outloop ; loop for next character

outdone:

; print CR/LF

        mov     al,0dh  ; print CR
        call    putc
        mov     al,0ah  ; print LF
        call    putc

; return to DOS

        int     20H

; variables

i       dw      1 dup (?)       ; array index
c       db      1 dup (?)       ; character read
buf     db      10 dup (?)      ; vowel buffer
n       dw      1 dup (?)       ; characters read

; read a character from keyboard
; the character is returned in AL
; AH is destroyed

getc:   mov     ah,1    ; DOS INT 21H function 1
        int     21h
        ret

; print a character
; character must be in AL

putc:   push    ax      ; save AX and DX
        push    dx
        mov     ah,2    ; DOS INT 21H function 2
        mov     dl,al
        int     21h
        pop     dx
        pop     ax
        ret

; determine if a character is a vowel
; input character is in al
; return ax=0 if not vowel, ax=1 if it is

vowels  db      "aeiouyAEIOUY",0

isvwl:
        push    bx      ; save BX

; search through a zero-terminated string of
; vowels for a match with al

        mov     bx,offset vowels ; point to start
                                 ;    of string
isvwl1:
        mov     ah,[bx] ; check character in string
        cmp     ah,0    ; if we've reached the zero
        je      novwl   ;    terminator it's not vowel
        cmp     al,ah   ; if the input character matches
        je      yesvwl  ;    then it is a vowel
        add     bx,1    ; point to next character
        jmp     isvwl1  ; loop back

yesvwl:
        mov     ax,1    ; return 1 if it's a vowel
        jmp     isvwl2

novwl:
        mov     ax,0    ; return 0 if it's not

isvwl2:
        pop     bx
        ret

; convert a letter to upper case
; input character in AL
; returns converted character in AL

toupper:
        cmp     al,'a'  ; if it's between 'a'
        jl      notlower
        cmp     al,'z'  ;   and 'z'
        jg      notlower
        sub     al,20h  ;   subtract 32
notlower:
        ret

code ends
        end     start
```

For an input of:

```
John Brown (12345678). The lAzY brown fOx jumps on
        the qUick dog's back$
```

the output of this program is:

```
John Brown (12345678). The lAzY brown fOx jumps on
        the qUick dog's back$
O O E A Y O O U O E
```