3 points for each 1-15:  
1 b  
2 e  
3 b  
4 c  
5 b  
6 b  
7 a  
8 b  
9 b  
10 d  
11 b  
12 e  
13 c  
14 c  
15 e  

2 points for each in 16:  
16 1. c  
17. e  
18. d  
5 b  

1 point for each in 17:  
17. 2,7,10,5,8,1,4,9,6,3  

5 points for each 18 & 19:  
18. Pseudo-code is a written description of the functionality for a particular software module. It should include the name of the module/subroutine, author, date, high-level description of functionality, and the actual steps.  
19. The use of a simulator is not great for embedded program development due to the need to interface with I/O devices.  

10 points for each 20 - 22:  
20. Hardware interrupts are asynchronous; they are not related to code that is currently executing on the processor. Software interrupts are a byproduct of code, which is executing on the processor.  
21. There were two errors in the code segment:  
a. (5 pts) pd3_1 should have been pd3_0.  
b. (5 pts) The #pragma INTERRUPT timer_isr line was missing for the timer isr.  
22. The segment of code should resemble something of this nature:  
char name[] = “My Name\0”; //my name  
int j; //loop counter  

//for each character in the name, display  
for(j=0; name[j] != NULL; j++)  
  disp_dataw(name[j]);  

23. Algorithm: (15 points)  
Transmit handshake string “Ready\n” using uart1, one character at a time  
Begin waiting for reception of ‘R’  
If Sw2 is pressed, turn off LED and return 1  
If R is received, break receive loop  
Transmit input string using uart1, one character at a time.
ECE 306 - Exam #2
Page 2 of 3

Turn on LED and return 0 when complete.

Code: (30 points)
//Method: SendInfo
//Purpose: This function is used to transmit
//      the input string, pointed to by
//      str, via the RS-232 port.
//Parameters: char *str - pointer to string to be transmitted
//Outputs: Returns 0 if transmission is successful,
//     1 otherwise
//Assumptions: The string pointed to by str must be NULL terminated.

int SendInfo(char* str) {
    char input = 'x';  // storage for input character - initialize to garbage
    char hndsk[] = "Ready?\n";  // "handshake" message

    // transmit the handshake string char by char
    // until the NULL character is found
    for(i = 0; hndsk[i] != NULL; i++) {
        ultbl = hndsk[i];  // send a character
        while(!ti_u1c1);  // wait for the transmission to complete
    } // end for loop

    // wait for an 'R' to be received or switch 2 to be pressed
    while(input != 'R') {
        while(!r1_u1c1) {  // wait for a character
            // check to see if the switch is being pressed.
            // If this is the case, exit
            if(!s_s2) {
                LED_G = LED_OFF;
                return 1;
            } // end if
        } // end while loop
    } // end while loop

    input = r1tbl;  // get the character

    // transmit the string character by character
    // until the NULL character is found
    for(i = 0; str[i] != NULL; i++) {
        ultbl = str[i];  // send a character
        while(!ti_u1c1);  // wait for the transmission to complete
    } // end for loop

    LED_G = LED_ON;  // turn on the LED

    return 0;
} // end SendInfo

Scoring:
• - 5 pts for coding standard (comments, using correct function name)
• - 5 pts for coding standard (comments, using correct function name)
• - 5 pts. for usage of uart/LED
• - 5 pts. for aborting on S2 press
• - 5 pts. for waiting for 'R' and not breaking loop on any other character
• - 5 pts. for handling of input string (i.e. transmitting until NULL character, etc)
• - 5 pts. for transmitting handshake string