(20) Question 1. Find the sequence of execution that leads to the incorrect value being displayed. Display is a function that is called from the foreground, while RTIHan is a background interrupt service routine periodically activated by the RTIF flag. If the interrupt occurs at a bad time, and time is a certain value, then Display will output an incorrect value.

Part a) Mark each line of executable C with a “1”, “2”, “3”,… symbol signifying the order of execution that yields a bad display. Also specify the value of Time before the interrupt (there is more than one).

```c
unsigned short Time;

// called from the foreground
void Display(void){
    Time++; // 0 to 59
    SCI_OutChar(Time/10+0x30); // tens
    if(Time == 60) Time=0;
    SCI_OutChar(Time%10+0x30); // ones
}

#pragma interrupt_handler RTIHan()
void RTIHan(void){
    RTIFLG = 0x80;
}
```

Part b) Rewrite the Display function to eliminate the bug.

(50) Question 2. You will write a software device driver for an input/output device. The output goes to a LCD display, and the input comes from a keypad. The I/O data are ASCII characters. The ritual should set the direction registers for ports A and B, initializing Port B to inputs. Your software should use busy-waiting (Gadfly) synchronization. All operations should be friendly. The hardware connections are as follows:

- PA7  -->  start  your software sets this high to start an I/O operation
- PA6  <--  done   this signal goes high when the I/O operation is complete
- PA5  -->  R/W    1 means input, and 0 means output
- PB7-PB0  <->  data  bidirectional, output to display, input from keypad

To output a letter to the LCD,

1) set R/W to 0
2) make Port B outputs
3) write Port B with the 8-bit data to display,
4) set start high
5) wait for done to be 1
6) set start low
7) wait for done to be 0
8) make Port B inputs again

To input a letter from the keypad,

1) set R/W to 1
2) set start high
3) wait for done to be 1
4) read input data
5) set start low
6) wait for done to be 0
Part a) Show the C code that goes in the IO.h header file.

Part b) Show the C code that goes in the IO.c implementation file.
Question 3. The overall goal is to draw a picture of the stack that exists while in the middle of the RTIHan. Starting at the top of main, hand-execute this software system; main calls function; the RTI interrupt occurs at the specified spot within function, and the RTIHan runs until the “show stack” comment. Local variables, parameters, and return values will be pushed on the stack. For each element pushed on the stack, give a general symbolic description (e.g., return pointer, old X, i1, f0) rather than its specific value (e.g., $F08A, $0000, 100, 4.)