```
;----- Lab 10 Multi-Byte Key based symmetric XOR encryption
; variable and constant definitions
keyBytesRAMaddress EQU 0x40 ; symbolic constant for base address of
                                  encryption key in RAM
                EQU 0x30
                             ; variable to track length of key
keyLength
keyvalIndex
                EQU 0xe0
                             ; variable to index the keyval constant array
                                  keyvalIndex is also an alias for accumulator
; begin section from lab 9
jmp main
                           ;jump past interrupt vector table
org 0x0030
                           ; put main program at rom location 0x0030
main:
   ;----; Initialization/configuration -----;
      ; keyval variable no longer used
            keyval, #0x23 ;load the keyval variable with encryption key
      mov
             tmod, \#0x20; config timer 1 mode 2
      mov
             scon, \#0x50; config serial 8-data, 1 start, 1 stop, no parity
      mov
             th1,
                   #0xFD ; 9600 baud
      mov
      setb
                          ; start timer 1 to enable serial communication
;end section from lab 9
; In the following section load the key bytes from ROM into RAM
                           #keyBytesRAMaddress ;initialize RAM pointer
             r0,
      mov
      mov
             dptr,
                          #keyvals2
                                       ;initialize ROM pointer
             keyvalIndex,
                          #0x00
                                            ;initialize keyvalIndex
      mov
   LoadKey:
                                             ;preserve keyvalIndex variable
      push
             keyvalIndex
                                             ; load byte of key from ROM
           a, #0x00,
                          notNull
                                             ; check for null terminating character
      cjne
          jmp LoadDone
                                             ; if null is found, enter main loop
      notNull:
                                             ; put byte of key into ram
          mov
                 @r0,
                                             ; restore keyvalIndex variable
                                             ;increment keyvalIndex
                                             ;increment RAM pointer
                                             ; continue the loop
   LoadDone:
            0r0, #0x00
      mov
                                             ;append null char to string
                                             ;re-initialize RAM pointer
   ;----;
; begin section from lab 9
mainloop:
      jnb
            ri, $
                                  ; wait to receive a char
      call
                                  ; char received, get it!
            getchar
      cjne a, \#0x00, encrypt ;check for null character in string
            a, #0x00, checkKeyVal ; check for null character in string
      cjne
```

```
terminate
                                   ;terminate program if null character is recieved
; end section from lab 9
checkKeyVal:
                                        ;go to Encrypt if keyVal is not null
                  r0, #keyBytesRAMaddress
                                               ;re-initialize RAM pointer
          mov
; begin section from lab 9
Encrypt:
       xrl
              a, @r0
                             ; encrypt the character contained in the accumulator
       call
             writechar
                             ; write the encrypted character
              mainloop
       jmp
terminate:
       mov
              a, #0x00
                            ; load null character into accumulator
       call
              writechar
                             ;append the null character to text output
       sjmp
                             ;halt
;-----
                         ----;
              getchar
; subroutine receives nothing before it is called
; writes the character to the serial console
; returns a byte in the accumulator
getchar:
       mov
              a, sbuf
                            ; get serial data (char)
       clr
              ri
                             ;acknowledge data received
                             ; return from subroutine call
       ret
;-----
              writechar
; receives byte or character
; reads a character that has been received serially
;returns the c
writechar:
              sbuf, a
                           ; send data (char) serially
              ti, $
                             ; wait until data is sent
       jnb
       clr
              ti
                             ;acknowledge data has been sent
                             ; return from subroutine call
       ret
;end section from lab 9
; multibyte keys are defined below, only one will be used at a time
org 0x200
keyvals:
         db '12345678',0
keyvals2:
          db 0x23, 0x34, 0x45, 0x56, 0x67, 0x78,0x89,0x90, 0x00
end
```