```
keyval  equ 0x30                ;map keyval to RAM location 0x30

jmp main                        ;jump past interrupt vector table

org 0x0030                      ;put main program at rom location 0x0030
main:
        mov     keyval, #0x23   ;load the keyval variable with encryption key
        mov     tmod,   #0x20   ;configure timer 1 mode 2
        mov     scon,   #0x50   ;configure serial 8-data, 1 start, 1 stop, no parity
        mov     th1,    #0xFD   ;9600 baud
        setb    tr1             ;start timer 1 to enable serial communication
mainloop:
        jnb     ri, $           ;poll receive flag
        call    getchar         ;char received, get it!
        cjne    a, #0x00, encrypt   ;check for null character
            jmp     terminate   ;terminate program if null character is recieved
encrypt:
        xrl     a, keyval       ;encrypt the character contained in the accumulator
        call    writechar       ;write the encrypted character
        jmp     mainloop
terminate:
        mov     a, #0x00        ;load null character into accumulator
        call    writechar       ;write the null character
        sjmp    $               ;halt

;-----------    getchar     ----------;
;subroutine receives nothing before it is called
;reads a character from the serial input (Rx)
;returns a byte or character in the accumulator
getchar:
        mov     a, sbuf         ;get serial data (char)
        clr     ri              ;acknowledge data received
        ret                     ;return from subroutine call

;-----------    writechar       ----------;
;subroutine receives a character to be written in the accumulator
;writes a character to serial output (Tx)
;returns nothing
writechar:
        mov     sbuf, a         ;send data (char) serially
        jnb     ti, $           ;wait until data is sent
        clr     ti              ;acknowledge data has been sent
        ret                     ;return from subroutine call
end
```