```
;Configuración de vectores de interrupción
            ORG     0000H           ;Dirección del origen del programa
            JMP     SETUP           ;Salta al Inicio del programa
;
            ORG     0003H           ;Dirección de interrupción externa INT0
            JMP     EXT_INT0        ;Salta a subrutina de interrupción
;
            ORG     0013H           ;Dirección de interrupción externa INT1
            JMP     EXT_INT1        ;Salta a subrutina de interrupción
;
            ORG     0023H           ;Dirección de interrupción
            JMP     UARTINT         ;Salta a la interrupción

;Definición de variables de programa
FLAG        EQU     50H             ;Reserva de dirección para el FLAG.

;Configuración de interrupción externa por flanco
SETUP:      MOV     TMOD,#01H       ;Configuramos TMOD con 1 (TMR0 - Modo 0)
16bits
            MOV     TCON,#01H       ;Configura IT0 externa por flanco
            MOV     IE,#01H         ;Configura EX0 Interrupción Externa 0
            MOV     IP,#00H         ;Des-habilita prioridad de interrupt
            SETB    EA              ;Habilita interrupción global

;Método de espera bloqueante de Flag
ESPERA1:    JNB     FLAG,ESPERA1    ;Espera a que el FLAG sea 1

;NOTA: TiempoOscilador: fXLS/Div = 12MHz/12=1us
;Valor del delay Tiempo[us]/TiempoOscilador[us]=10ms/1us=10000
;Calculando TLO y THO: 65536-10000=55536 -> D8F0
TICKTMR:    MOV     TL0,#0F0H       ;Configuramos TL0 con F0h
            MOV     TH0,#0D8H       ;Configuramos TH0 con D8h
            SETB    TR0             ;Start Timer
ESPERA:     JNB     TF0,ESPERA      ;Espera al flag TF0=1 (espera que se
desborde)
            CLR     TR0             ;Stop Timer
            CLR     TF0             ;Borra flag TF0
            RET                     ;Retorno de Subrutina

;Interrupción externa 0
EXT_INT0:   SETB    FLAG           ;Pone FLAG a TRUE al entrar en la
interrupción
            CPL     P2.2           ;Toggle en la salida P2.2
            RETI                   ;Retorno de interrupción

;Recorrido de una tabla
            MOVC    A,@A+DPTR       ;Trae el valor cargado en acum desde la
tabla
            MOV     P1,A            ;Muestra el valor del acum en el puerto

;Función de división y resto
DEC2BCD:    MOV     A,P2            ;Carga acum con el valor del ADC del puerto
P2
            MOV     B,#10D          ;Carga B con 10
            DIV     AB              ;Divide A/B=A, Resto en B
            MOV     R0,B            ;Mueve Resto B a R0

;Delay bloqueante básico
DELAY:      MOV     R7,#255         ;Carga R7 con 255
            DJNZ    R7,$            ;Decremento R7
            RET                     ;Retorna subrutina

;Definición de tabla por bytes
TABLE:      DB      00111111B       ;Numero 0 en Display de 7 segmentos
```

```
            DB      00000110B       ;Numero 1 en Display de 7 segmentos

;Definición de tabla por string
TABLA:      DB      "UTN INSPT ",0  ;Tabla de 10 datos a enviar por UART


;Función para buscar el valor mayor (guarda dato y direccion)
BUSCMAX:    MOV     A,@R0           ;Cargamos el Acumulador con el valor
apuntado por R0
            SUBB    A,VALMAX        ;Calculamos A - VALMAX
            JC      CARRY0          ;Si JC=1 (es menor), salta a Carry
            MOV     VALMAX,@R0      ;Si JC=0 (es mayor), lo guarda en VALMAX
            MOV     DIRMAX,R0       ;Al mismo tiempo guarda la dirección en
DIRMAX
CARRY0:     INC     R0              ;Incrementa valor del puntero en la tabla
            DJNZ    CONT,BUSCMAX    ;Realiza las iteraciones preestablecidas
            RET                     ;Retorno de la subrutina

;Función para buscar el valor menor (guarda dato y direccion)
BUSCMIN:    MOV     A,@R0           ;Cargamos el Acumulador con el valor
apuntado por R0
            SUBB    A,VALMIN        ;Calculamos A - VALMIN
            JNC     CARRY1          ;Si JC=0 (es mayor), salta a Carry
            MOV     VALMIN,@R0      ;Si JC=1 (es menor), lo guarda en VALMIN
            MOV     DIRMIN,R0       ;Al mismo tiempo guarda la dirección en
DIRMIN
CARRY1:     INC     R0              ;Incrementa valor del puntero en la tabla
            DJNZ    CONT,BUSCMIN    ;Realiza las iteraciones preestablecidas
            RET                     ;Retorno de la subrutina

;Función para comparar un valor
COMPARA:    MOV     A,P0            ;Cargamos el valor del puerto P0 en el
Acumulador
            CJNE    A,CONST,NOTEQ   ;Si A != CONST, si es distinto salta a NOTEQ
            SJMP    IGUAL           ;Si A = CONST, si es igual salta a IGUAL
NOTEQ:      JC      MENOR           ;Si Carry = 1, salta a MENOR
            SJMP    MAYOR           ;Si Carry = 0, salta a MAYOR
IGUAL:      SETB    P1.1            ;Enciende led igual
            CLR     P1.0            ;Apaga led mayor
            CLR     P1.2            ;Apaga led menor
            RET                     ;Retorno de subrutina
MAYOR:      SETB    P1.0            ;Enciende led mayor
            CLR     P1.2            ;Apaga led menor
            CLR     P1.1            ;Apaga led igual
            RET                     ;Retorno de subrutina
MENOR:      SETB    P1.2            ;Enciende led menor
            CLR     P1.0            ;Apaga led igual
            CLR     P1.1            ;Apaga led mayor
            RET                     ;Retorno de subrutina

;Configuración para el uso de UART en 9600bps
SETUP:      MOV     SCON, #50h      ;UART en MODO 1 (8BIT), REN=1
            ORL     TMOD, #20h      ;TIMER 1 en MODO 2
            MOV     TH1, #0FDh      ;9600bps @ 11.059MHz
            MOV     TL1, #0FDh      ;9600bps @ 11.059MHz
            SETB    ES              ;Habilitación de Interrupción Serial
            SETB    EA              ;Habilitación de Interrupciones Globales

;Interrupción por dato presente en UART
UARTINT:    JNB     RI,UARTINT      ;Espera el dato Serie
            MOV     A,SBUF          ;Mueve el dato del buffer al acumulador
            CLR     RI              ;Borra el flag de recepción
            MOV     P0,A            ;Mueve valor recibido a puerto 0
            RETI                    ;Retorno de Interrupción
```

```
;Envió de datos por UART
SEND:       MOV     SBUF,A          ;Cargo el dato en Buffer de salida Serie
EOB:        JNB     TI,EOB          ;Envía hasta el ultimo bit
            CLR     TI              ;Borra el flag de transmisión
            RET                     ;Retorno de subrutina

;Fin de un programa
            END                     ;Fin del programa
```