

Sugestões de resposta – Exercícios do laboratório 1

Exemplo 1

```
MOV A, R2 .....EA
ADD A, 32H .....25 32
MOV 32H, A .....85 E0 32
SJMP 0H .....80 F8
```

Exercício 1

```
MOV A, R2 .....EA
ADD A, 0x01 .....25 01
MOV 0x00, ACC .....85 E0 00
SJMP 0H .....80 F8
```

Exercício 2

```
MOV A, R2 .....EA
SUBB A, 0x01 .....95 01
MOV 0x00, ACC .....85 E0 00
SJMP 0H .....80 F8
```

Exercício 3 – opção 1

```
MOV A, R2 .....EA
SUBB A, 0x01 .....95 01
MOV DPTR, #0002 ...90 00 02
MOVX @DPTR, A .....F0
SJMP 0H .....80 F7
```

Exercício 3 – opção 2

```
MOV A, R2 .....EA
SUBB A, 0x01 .....95 01
MOV P2, #00 .....75 A0 00
MOV R0, #02 .....78 02
MOVX @R0, A .....F2
SJMP 0H .....80 F5
```

Sugestões de resposta – Exercícios do laboratório 2

Exemplo 1

```
INICIO    EQU 0H
          ORG INICIO
          MOV 02H, #3
VOLTA:    MOV A, R2
          ADD A, 32H
          MOV 32H, A
          JMP VOLTA
          END
```

Exemplo 2

```
RESET     EQU 0H
VETOR     EQU 60H

          ORG RESET
          MOV DPTR, #NRO
          MOV A, #0
          MOVC A, @A+DPTR
          MOV R1, A
          MOV DPTR, #DADOS
          MOV R2, #0
          MOV R0, #0
VOLTA:    MOV A, R0
          MOVC A, @A+DPTR
          ADD A, R2
          MOV R2, A
          INC R0
          DJNZ R1, VOLTA
FIM:      JMP FIM

          ORG VETOR
NRO:      DB 03H
DADOS:    DB 01H, 03H, 05H, 06H, 0AH, 0E2H
          END
```

Exercício 1

```
RESET      EQU 0H
VETOR      EQU 60H

                ORG RESET
                MOV DPTR, #NRO
                MOV A, #0
                MOVC A, @A+DPTR
                MOV R1, A
                MOV DPTR, #DADOS
                MOV R2, #0
                MOV R0, #0
VOLTA:      MOV A, R0
                MOVC A, @A+DPTR
                ADD A, R2
                MOV R2, A
                INC R0
                DJNZ R1, VOLTA
                MOV DPTR, #01H
                MOV A, R2
                MOVX @DPTR, A
FIM:        JMP FIM

                ORG VETOR
NRO:        DB 03H
DADOS:      DB 01H, 03H, 05H, 06H, 0AH, 0E2H
END
```

Exercício 2

```
RESET      EQU 0H
VETOR      EQU 60H

                ORG RESET
                MOV DPTR, #NRO
                MOV A, #0
                MOVC A, @A+DPTR
                JZ FIM
                MOV R1, A
                MOV DPTR, #DADOS
                MOV R2, #0
                MOV R0, #0
VOLTA:      MOV A, R0
                MOVC A, @A+DPTR
                ADD A, R2
                MOV R2, A
                INC R0
                DJNZ R1, VOLTA
                MOV DPTR, #01H
                MOV A, R2
                MOVX @DPTR, A
FIM:        JMP FIM

                ORG VETOR
NRO:        DB 03H
DADOS:      DB 01H, 03H, 05H, 06H, 0AH, 0E2H
END
```

Exercício 3

```
RESET      EQU 0H
VETOR      EQU 60H

                ORG RESET
                MOV DPTR, #NRO
                MOV A, #0
                MOVC A, @A+DPTR
                JZ FIM
                MOV R1, A
                MOV DPTR, #DADOS
                MOV R2, #0
                MOV R0, #0
                MOV R3, #0
VOLTA:       MOV A, R0
                MOVC A, @A+DPTR
                ADD A, R2
                MOV R2, A
                MOV A, #0
                ADDC A, R3
                MOV R3, A
                INC R0
                DJNZ R1, VOLTA
                MOV DPTR, #01H
                MOV A, R2
                MOVX @DPTR, A
                MOV DPTR, #00H
                MOV A, R3
                MOVX @DPTR, A
FIM:         JMP FIM

                ORG VETOR
NRO:         DB 07H
DADOS:       DB 01H, 03H, 05H, 06H, 0AH, 0E2H, 055H
END
```

Exercício 4

```
RESET      EQU 0H
MEMINT     EQU 50H
MEMEXT     EQU 2200H

                ORG RESET
                MOV A, #0H
                MOV R0, #0
                MOV R1, #MEMINT
                MOV DPTR, #MEMEXT
LOOP:        INC A
                MOV @R1, A
                MOVX @DPTR, A
                INC R1
                INC DPTR
                CJNE A, #05H, LOOP
END
```

Exercício 5

```
RESET      EQU 0H
MEXTINI    EQU 2100H
MEXTFIM    EQU 2300H

            ORG RESET
            MOV DPTR, #MEXTINI
            MOV A, #0A0H
LOOP1:      MOVX @DPTR, A
            INC A
            INC DPTR
            CJNE A, #0B0H, LOOP1
            MOV DPTR, #MEXTINI
LOOP2:      MOVX A, @DPTR
            MOV DPH, #23H
            MOVX @DPTR, A
            INC DPL
            MOV DPH, #21H
            MOV A, DPL
            CJNE A, #10H, LOOP2
FIM:        JMP FIM
            END
```

Sugestões de resposta – Exercícios do laboratório 3

Exemplo 1

```
CS      EQU P0.7
END0    EQU P3.3
END1    EQU P3.4

        ORG 0H
        CLR END0
        CLR END1
        SETB CS
        MOV A, #1
        CALL CONVERTE
        MOV P1, A
        JMP $
CONVERTE: INC A
        MOVC A, @A+PC
        RET
TABELA: DB 40H, 79H, 24H, 30H, 19H, 12H, 02H, 78H
        DB 00H, 10H, 08H, 03H, 46H, 21H, 06H, 0EH
        END
```

Exercício 1

```
CS      EQU P0.7
END0    EQU P3.3
END1    EQU P3.4

        ORG 0H
        CLR END0
        CLR END1
        SETB CS
VOLTA:   MOV A, P2
        CPL A
        CALL CONVERTE
        MOV P1, A
        JMP VOLTA
CONVERTE: INC A
        MOVC A, @A+PC
        RET
TABELA: DB 40H, 79H, 24H, 30H, 19H, 12H, 02H, 78H
        DB 00H, 10H, 08H, 03H, 46H, 21H, 06H, 0EH
        END
```

Exercício 2

```
CS      EQU P0.7
END0    EQU P3.3
END1    EQU P3.4

        ORG 0H
        SETB END0
        CLR END1
        SETB CS
volta0: MOV R0, #8
        MOV R1, #0
        MOV A, P2
volta:  RLC A
        JC salta
        INC R1
salta:  DJNZ R0, volta
        MOV A, R1
        CALL CONVERTE
        MOV P1, A
        JMP volta0

CONVERTE: INC A
        MOVC A, @A+PC
        RET

TABELA: DB 40H, 79H, 24H, 30H, 19H, 12H, 02H, 78H, 00H, 10H, 08H, 03H, 46H, 21H, 06H, 0EH
        END
```

Exercício 3

```
CS      EQU P0.7
END0    EQU P3.3;
END1    EQU P3.4;
ATRASO  EQU 0FEH

        ORG 0H
        CLR END0
        CLR END1
        SETB CS
VOLTA1: MOV R0, #9
VOLTA2: MOV A, R0
        CALL CONVERTE
        MOV P1, A
        MOV A, #ATRASO
        CALL DELAY
        DEC R0
        CJNE R0, #0FFH, VOLTA2
        JMP VOLTA1

DELAY:  DJNZ ACC, DELAY
        RET

CONVERTE: INC A
        MOVC A, @A+PC
        RET

TABELA: DB 40H, 79H, 24H, 30H, 19H, 12H, 02H, 78H, 00H, 10H, 08H, 03H, 46H, 21H, 06H, 0EH
        END
```

Exercício 4

```
CS      EQU P0.7
END0    EQU P3.3
END1    EQU P3.4
ATRASSO EQU 0FEH

        ORG 0H
        CLR END0
        CLR END1
        SETB CS
VOLTA1: MOV R0, #9
VOLTA2: MOV A, R0
        CALL CONVERTE
        MOV P1, A
        MOV A, #ATRASSO
        CALL DELAY
        DEC R0
        CJNE R0, #0FFH, VOLTA2
        JMP VOLTA1

DELAY:   PUSH ACC
        PUSH PSW
        PUSH 0H
DEL:     DJNZ ACC, DEL
        POP 0H
        POP PSW
        POP ACC
        RET
CONVERTE: INC A
        MOVC A, @A+PC
        RET

TABELA: DB 40H, 79H, 24H, 30H, 19H, 12H, 02H, 78H, 00H, 10H, 08H, 03H, 46H, 21H, 06H, 0EH
        END
```


Exercício 5

```
CS      EQU P0.7

        ORG 0H
        MOV R1, #01H
        MOV R2, #80H
        MOV P1, R1
        MOV P2, R2
        MOV R6, #0F0H
        CALL DELAY
        MOV A, R1
        RL A
        MOV R1, A
        MOV P1, R1
        CLR CS
VOLTA:   CALL LED
        MOV R6, #0F0H
        CALL DELAY
        JMP VOLTA

DELAY:   DJNZ R6, DELAY
        RET

LED:     MOV P1, R1
        MOV A, R1
        RL A
        MOV R1, A
        RET
        END
```

Exercício 6

```
CS          EQU P0.7

            ORG 0H
            CLR CS
            MOV R0, #0H
            MOV R1, #0H
            SETB C
            MOV P1, R1
VOLTA:      MOV A, #02H
            CALL DELAY
            CALL LED
            JMP VOLTA

; Sub-rotina 1 - inserção de atraso
DELAY:      DJNZ ACC, DELAY
            RET

; Sub-rotina 2 - acende/apaga LEDs
LED:        MOV A, R0
            JNZ APAGA
            MOV A, R1
            SETB C
            RLC A
            JNC RETRN
            MOV A, #7FH
            CLR C
            MOV R0, #1H
            JMP RETRN
APAGA:      MOV A, R1
            CLR C
            RRC A
            JC RETRN
            MOV R0, #0H
            MOV A, #01H
RETRN:      MOV P1, A
            MOV R1, A
            RET
            END
```

Exercício 7

```
ORG 0H
CLR P0.7
MOV R0, #0H
CLR A
MOV P1, #1H
AGUARDA: JB P2.7, AGUARDA
MOV A, P2
XCHD A, @R0
MOV A, P1
JB P2.6, ESQUERDA
CALL DIREITA
JMP AGUARDA
ESQUERDA: RL A
MOV P1, A
DJNZ R0, ESQUERDA
JMP AGUARDA
DIREITA: RR A
MOV P1, A
DJNZ R0, DIREITA
RET
END
```

Exercício 8

```
CS      EQU P0.7
END0    EQU P3.3
END1    EQU P3.4

        ORG 0H
        CLR END0
        CLR END1
        SETB CS
VOLTA:   CALL TECLADO
        MOV A, R0
        CALL CONVERTE
        MOV P1, A
        JMP VOLTA

TECLADO: ORL P0, #7FH
        CLR F0
        MOV R0, #0
        ; varredura primeira linha
        CLR P0.3
        CALL COLSCAN
        JB F0, FIM
        ; varredura segunda linha
        SETB P0.3
        CLR P0.2
        CALL COLSCAN
        JB F0, FIM
        ; varredura terceira linha
        SETB P0.2
        CLR P0.1
        CALL COLSCAN
        JB F0, FIM
        ; varredura quarta linha
        SETB P0.1
        CLR P0.0
        CALL COLSCAN
        JB F0, FIM
        JMP TECLADO
FIM:     RET

COLSCAN: JNB P0.6, GOTKEY
        INC R0
        JNB P0.5, GOTKEY
        INC R0
        JNB P0.4, GOTKEY
        INC R0
        RET
GOTKEY:  SETB F0
        RET

CONVERTE: INC A
        MOVC A, @A+PC
        RET

TABELA:  DB 79H, 24H, 30H, 19H, 12H, 02H, 78H, 00H, 10H, 08H, 40H, 46H
        END
```

Sugestões de resposta – Exercícios do laboratório 4

Exemplo 1

```
RESET      EQU 0H
LTINT0     EQU 03H
STATE      EQU 20H

          ORG RESET
          JMP INICIO

          ORG LTINT0
          JMP HANDLER

INICIO:    MOV IE, #10000001B
          SETB IT0
          MOV STATE, #0H
          MOV R0, #STATE
          MOV DPTR, #TABELA
          MOV R1, #0
VOLTA:     CJNE @R0, #1, VOLTA
          MOV STATE, #0H
          MOV A, R1
          MOVC A, @A+DPTR
          MOV P1, A
          INC R1
          CJNE R1, #16, VOLTA
          JMP $

HANDLER:   MOV STATE, #1
          RETI

TABELA:    DB "Microcontrolador"
          END
```

Exercício 1

```
RESET      EQU 00H
LTINT0     EQU 03H
LTINT1     EQU 13H
STATE      EQU 20H

          ORG RESET
          JMP INICIO

          ORG LTINT0
          MOV STATE, #1H
          RETI

          ORG LTINT1
          CPL EX0
          CLR IE0
          RETI

INICIO:    MOV IE, #10000101B
          SETB IT0
          SETB IT1
          MOV STATE, #0H
          MOV R0, #STATE
          MOV DPTR, #TABELA
          MOV R1, #0
VOLTA:     CJNE @R0, #1, VOLTA
          MOV STATE, #0H
          MOV A, R1
          MOVC A, @A+DPTR
          MOV P1, A
          INC R1
          CJNE R1, #16, VOLTA
          JMP $

TABELA:    DB 'Microcontrolador'
          END
```

Exercício 2

```
LTINT0    EQU 03H
ATRASO    EQU 0FEH

          ORG 0H
          JMP INICIO

          ORG LTINT0
          CPL 0H
          RETI

INICIO:    MOV IE, #10000001B
          SETB IT0
          MOV 20H, #0H
          MOV R1, #0
VOLTA:     MOV R2, 20H
          CJNE R2, #0, VOLTA
          MOV A, R1
          MOV P1, A
          INC R1
          MOV A, #ATRASO
          CALL DELAY
          JMP VOLTA

DELAY:     DJNZ R5, DELAY
          RET
          END
```

Exercício 3

```
STATE      EQU 20H

            ORG 0H
            JMP INICIO

            ORG 03H
            JMP TRATADOR0

            ORG 13H
            JMP TRATADOR1

INICIO:     MOV IE, #10000101B
            SETB IT0
            SETB IT1
            SETB RS0
            CLR RS1
            MOV SP, #10H
            MOV STATE, #0H
            MOV R0, #STATE
            MOV DPTR, #TABELA
            MOV R1, #0
            MOV R5, #0
VOLTA:      CJNE @R0, #1, VOLTA
            MOV STATE, #0H
            MOV A, R1
            MOVC A, @A+DPTR
            MOV P1, A
            INC R1
            CJNE R1, #16, VOLTA
            MOV R1, #0
            JMP VOLTA

TRATADOR0:  MOV STATE, #1
            RETI

TRATADOR1:  PUSH DPL
            PUSH DPH
            PUSH ACC
            MOV DPH, #20H
            MOV DPL, R5
            MOV A, P1
            MOVX @DPTR, A
            INC R5
            CJNE R5, #10H, FIMINT
            MOV R5, #0
FIMINT:     POP ACC
            POP DPH
            POP DPL
            RETI

TABELA:     DB "Microcontrolador"
            END
```


Exercício 4

```
SENTIDO EQU 20H
STATE EQU 21H

ORG 0H
JMP INICIO

ORG 03H
CPL 0H
RETI

ORG 13H
CPL 8H
RETI

INICIO: MOV IE, #10000101B
        SETB IT0
        SETB IT1
        MOV SENTIDO, #0H
        MOV R0, #SENTIDO
        MOV R1, #STATE
        MOV R2, #01H
        MOV P1, R2
VOLTA:  MOV A, P1
ESPERA: CJNE @R1, #0, ESPERA
        CJNE @R0, #0, ESQ
        RR A
        JMP PORTA
ESQ:    RL A
PORTA:  MOV P1, A
DELAY:  DJNZ R6, DELAY
        JMP VOLTA
END
```

Sugestões de resposta – Exercícios do laboratório 5

Exemplo 1

```
CS      EQU P0.7
EN      EQU P1.2
RS_0    EQU 0
RS_1    EQU 00001000b
ATRASSO EQU 50H

        ORG 0H
        CLR CS
        CALL INITDSP
        ; Escreve mensagem
        MOV DPTR, #MENS
        MOV R2, #RS_1
        CALL WRT
        JMP $

MENS:    DB 13,"EEL7030 - LCD"
COMANDOS: DB 03H, 28H, 0FH, 06H

INITDSP: MOV P1, #20H
        SETB EN
        CLR EN
        CALL DELAY
        ; Nibble alto do Function Set é enviado duas vezes
        MOV R2, #RS_0
        MOV DPTR, #COMANDOS
        CALL WRT
        RET

WRT:     MOV A, #0
        MOVC A, @A+DPTR
        MOV R6, A
        MOV R1, #1H
LOOP:    MOV A, R1
        MOVC A, @A+DPTR
        MOV B, A
        ANL A, #0F0H
        ORL A, R2
        MOV P1, A
        SETB EN
        CLR EN
        MOV A, B
        SWAP A
        ANL A, #0F0H
        ORL A, R2
        MOV P1, A
        SETB EN
        CLR EN
        CALL DELAY
        INC R1
        DJNZ R6, LOOP
        RET

DELAY:   MOV R0, #ATRASSO
        DJNZ R0, $
        RET
        END
```

Exercício 1

```
EN          EQU P1.2
CS          EQU P0.7
RS_0        EQU 0
RS_1        EQU 00001000b
ATRASSO     EQU 50H
CMD_COL_07  EQU 87H

                ORG 0H
                CLR CS
                CALL INITDSP

                ; Escreve mensagem
                MOV DPTR, #MENS
                MOV R2, #RS_1
                CALL WRT
VOLTA:        MOV A, P2
                CPL A
                PUSH ACC
                ANL A, #0F0H
                SWAP A
                CALL CONVERTE
                MOV R2, #RS_1
                CALL WRT2
                POP ACC
                ANL A, #0FH
                CALL CONVERTE
                CALL WRT2
                MOV A, #CMD_COL_07
                MOV R2, #RS_0
                CALL WRT2
                JMP VOLTA

MENS:         DB 7,"Valor: "

COMANDOS: DB 03H, 28H, 0CH, 06H

INITDSP:      MOV P1, #20H
                SETB EN
                CLR EN
                CALL DELAY
                ; Nibble alto do Function Set é enviado duas vezes
                MOV R2, #RS_0
                MOV DPTR, #COMANDOS
                CALL WRT
                RET

WRT:          MOV A, #0
                MOVC A, @A+DPTR
                MOV R6, A
                MOV R1, #1H
LOOP:         MOV A, R1
                MOVC A, @A+DPTR
                CALL WRT2
                INC R1
                DJNZ R6, LOOP
                RET

WRT2:         MOV B, A
                ANL A, #0F0H
                ORL A, R2
                MOV P1, A
                SETB EN
```

```

        CLR EN
        MOV A, B
        SWAP A
        ANL A, #0F0H
        ORL A, R2
        MOV P1, A
        SETB EN
        CLR EN
        CALL DELAY
        RET

CONVERTE: INC A
          MOVC A, @A+PC
          RET

TABELA:  DB 30H, 31H, 32H, 33H, 34H, 35H, 36H, 37H
          DB 38H, 39H, 41H, 42H, 43H, 44H, 45H, 46H

DELAY:   MOV R0, #ATRASSO
          DJNZ R0, $
          RET

END

```

Exercício 2

```
EN          EQU P1.2
CS          EQU P0.7
RS_0       EQU 0
RS_1       EQU 00001000b
ATRASSO    EQU 50H
POS_CURSOR EQU 90H
DSP_SHIFT_LEFT EQU 18H

        ORG 0H
        CLR CS
        CALL INITDSP
        ; Escreve mensagem
        MOV DPTR, #MENS
        MOV R2, #RS_1
        CALL WRT
        MOV R2, #RS_0
        MOV A, #POS_CURSOR
        CALL WRT2
        MOV DPTR, #MENS
        MOV R2, #RS_1
        CALL WRT
        CALL DELAY2
        MOV R2, #0H
VOLTA:    MOV A, #DSP_SHIFT_LEFT
        CALL WRT2
        CALL DELAY2
        JMP VOLTA

MENS:     DB 14,"EEL7030 - 8051"

COMANDOS: DB 03H, 28H, 0CH, 06H

INITDSP: MOV P1, #20H
        SETB EN
        CLR EN
        CALL DELAY
        ; Nibble alto do Function Set é enviado duas vezes
        MOV R2, #RS_0
        MOV DPTR, #COMANDOS
        CALL WRT
        RET

WRT:      MOV A, #0
        MOVC A, @A+DPTR
        MOV R6, A
        MOV R1, #1H
LOOP:     MOV A, R1
        MOVC A, @A+DPTR
        CALL WRT2
        INC R1
        DJNZ R6, LOOP
        RET

WRT2:     MOV B, A
        ANL A, #0F0H
        ORL A, R2
        MOV P1, A
        SETB EN
        CLR EN
        MOV A, B
        SWAP A
        ANL A, #0F0H
```

```

    ORL A, R2
    MOV P1, A
    SETB EN
    CLR EN
    CALL DELAY
    RET

DELAY:    MOV R0, #ATRASSO
          DJNZ R0, $
          RET

DELAY2:   MOV R1, #0F0H
VOLTA2:   MOV R0, #0FFH
          DJNZ R0, $
          DJNZ R1, VOLTA2
          RET

END

```

Exercício 3

```
EN      EQU P1.2
CS      EQU P0.7
RS_0    EQU 0
RS_1    EQU 00001000b
ATRASSO EQU 50H
POS_CURSOR EQU 90H
DSP_SHIFT_LEFT EQU 18H
DSP_SHIFT_RIGHT EQU 1CH

        ORG 0H
        CLR CS
        CALL INITDSP

        ; Escreve mensagem
        MOV DPTR, #MENS
        MOV R2, #RS_1
        CALL WRT
        MOV R2, #RS_0
        MOV A, #POS_CURSOR
        CALL WRT2
        MOV DPTR, #MENS
        MOV R2, #RS_1
        CALL WRT
        CALL DELAY2
        MOV R2, #RS_0
VOLTA:   MOV A, P2
        ANL A, #080H
        CJNE A, #0, RIGHT
        MOV A, #DSP_SHIFT_RIGHT
        JMP VAI
RIGHT:   MOV A, #DSP_SHIFT_LEFT
VAI:     CALL WRT2
        CALL DELAY2
        JMP VOLTA

MENS:    DB 14,"EEL7030 - 8051"
COMANDOS: DB 03H, 28H, 0CH, 06H

INITDSP: MOV P1, #20H
        SETB EN
        CLR EN
        CALL DELAY
        ; Nibble alto do Function Set é enviado duas vezes
        MOV R2, #RS_0
        MOV DPTR, #COMANDOS
        CALL WRT
        RET

WRT:     MOV A, #0
        MOVC A, @A+DPTR
        MOV R6, A
        MOV R1, #1H
LOOP:    MOV A, R1
        MOVC A, @A+DPTR
        CALL WRT2
        INC R1
        DJNZ R6, LOOP
        RET

WRT2:    MOV B, A
        ANL A, #0F0H
        ORL A, R2
```

```

    MOV P1, A
    SETB EN
    CLR EN
    MOV A, B
    SWAP A
    ANL A, #0F0H
    ORL A, R2
    MOV P1, A
    SETB EN
    CLR EN
    CALL DELAY
    RET

DELAY:    MOV R0, #ATRASSO
          DJNZ R0, $
          RET

DELAY2:   MOV R1, #0F0H
VOLTA2:   MOV R0, #0FFH
          DJNZ R0, $
          DJNZ R1, VOLTA2
          RET

END

```


Exercício 4

```
EN      EQU P1.2
CS      EQU P0.7
;STATE  EQU 20H.0
RS_0    EQU 0
RS_1    EQU 00001000b
ATRASSO EQU 50H
POS_CURSOR0 EQU 80H
POS_CURSOR7 EQU 87H

        ORG 0H
        JMP INICIO

        ORG 13H
        ;SETB STATE
        SETB 20H.0
        RETI

INICIO:  MOV IE, #10000100b
        SETB IT1
        ;CLR STATE
        CLR 20H.0
        CLR CS
        CALL INITDSP
        ; Escreve mensagem
        MOV DPTR, #MENS
        MOV R2, #RS_1
        CALL WRT

VOLTA:   MOV P0, #01110000b
        ;JNB STATE, $
        JNB 20H.0, $
        ;CLR STATE
        CLR 20H.0
        CLR EA
        CALL KEYPAD
        CLR IE1
        SETB EA
        MOV A, R0
        CALL CONVERTE
        MOV R5, A
        MOV R2, #RS_0
        MOV A, #POS_CURSOR7
        CALL WRT2
        MOV A, R5
        MOV R2, #RS_1
        CALL WRT2
        JMP VOLTA

CONVERTE: INC A
        MOVC A, @A+PC
        RET

TABELA:  DB 31H, 32H, 33H, 34H, 35H, 36H, 37H, 38H, 39H, 41H, 30H, 43H, 45H

MENS:    DB 7, "Valor: "
COMANDOS: DB 03H, 28H, 0CH, 06H

INITDSP: MOV P1, #20H
        SETB EN
        CLR EN
        CALL DELAY
        ; Nibble alto do Function Set é enviado duas vezes
        MOV R2, #RS_0
```

```

        MOV DPTR, #COMANDOS
        CALL WRT
        RET

WRT:     MOV A, #0
        MOVC A, @A+DPTR
        MOV R6, A
        MOV R1, #1H
LOOP:    MOV A, R1
        MOVC A, @A+DPTR
        CALL WRT2
        INC R1
        DJNZ R6, LOOP
        RET

WRT2:    MOV B, A
        ANL A, #0F0H
        ORL A, R2
        MOV P1, A
        SETB EN
        CLR EN
        MOV A, B
        SWAP A
        ANL A, #0F0H
        ORL A, R2
        MOV P1, A
        SETB EN
        CLR EN
        CALL DELAY
        RET

DELAY:   MOV R0, #ATRASSO
        DJNZ R0, $
        RET

DELAY2:  MOV R1, #0F0H
VOLTA2:  MOV R0, #0FFH
        DJNZ R0, $
        DJNZ R1, VOLTA2
        RET

; Subrotina que retorna em R0 o valor pressionado
; no teclado do EDSIM51
KEYPAD: ORL P0, #7Fh
        CLR F0
        MOV R0, #0
        ; Varre primeira linha
        CLR P0.3
        CALL COLSCAN
        JB F0, FIM
        ; Varre segunda linha
        SETB P0.3
        CLR P0.2
        CALL COLSCAN
        JB F0, FIM
        ; Varre terceira linha
        SETB P0.2
        CLR P0.1
        CALL COLSCAN
        JB F0, FIM
        ; Varre quarta linha
        SETB P0.1
        CLR P0.0
        CALL COLSCAN

```

```

        JB F0, FIM
        JMP KEYPAD
FIM:    RET

; Subrotina que varre as colunas para identificar a
; qual pertence a tecla pressionada o registrador R0 é
; incrementado a cada insucesso de forma a conter o
; número da tecla pressionada
COLSCAN: JNB P0.6, GOTKEY
        INC R0
        JNB P0.5, GOTKEY
        INC R0
        JNB P0.4, GOTKEY
        INC R0
        RET
GOTKEY: SETB F0
        RET

END

```