```
// PIC16F877A + TECLADO MATRICIAL + LCD + KIT MICROGENIOS @ 8 MHz
// ATENCAO!!!     LIGAR APENAS O LCD + NEHUM OUTRO PERIFERICO EH NECESSARIO!!!
// VERIFICAR O FUNCIONAMENRO DO TECLADO MATRICIAL PARA FUTURAS APLICACOES!!!!!
sbit LCD_RS at RE2_bit;                    // LCD MODULE CONNECTIONS MICROGENIOS
sbit LCD_EN at RE1_bit;
sbit LCD_D4 at RD4_bit;
sbit LCD_D5 at RD5_bit;
sbit LCD_D6 at RD6_bit;
sbit LCD_D7 at RD7_bit;
sbit LCD_RS_Direction at TRISE2_bit;
sbit LCD_EN_Direction at TRISE1_bit;
sbit LCD_D4_Direction at TRISD4_bit;
sbit LCD_D5_Direction at TRISD5_bit;
sbit LCD_D6_Direction at TRISD6_bit;
sbit LCD_D7_Direction at TRISD7_bit;              // END LCD MODULE CONNECTIONS
#define KP_R1     PORTD.RD0   // LINHA 1            // START - KEYPAD SETTINGS
#define KP_R2     PORTD.RD1   // LINHA 2
#define KP_R3     PORTD.RD2   // LINHA 3
#define KP_R4     PORTD.RD3   // LINHA 4
#define KP_C1     PORTB.RB0   // COLUNA 1
#define KP_C2     PORTB.RB1   // COLUNA 2
#define KP_C3     PORTB.RB2   // COLUNA 3              // END - KEYPAD SETTINGS
// VARIABLES DECLARATIONS
unsigned char key_get;
unsigned int i = 0;
unsigned char keypad_wait(void);
unsigned char keypad_read(void);
unsigned short t = 10;                                                 // KP TIME
// KEYPAD READ FUNCTION - START
unsigned char keypad_read(void){

// START KEYPAD SCANNING PROCESS
KP_R1 = 0;      // SCAN KEYPAD ON FIRST ROW: 1, 2, 3, A
KP_R2 = 1;      KP_R3 = 1;     KP_R4 = 1;
VDelay_ms(t);
if (KP_C1 == 0) return '*';     // KEY '*' IS PRESSED
if (KP_C2 == 0) return '0';     // KEY '0' IS PRESSED
if (KP_C3 == 0) return '#';     // KEY '#' IS PRESSED
VDelay_ms(t);
KP_R1 = 1;      // SCAN KEYPAD ON SECOND ROW: 1, 2, 3, A
KP_R2 = 0;      KP_R3 = 1;     KP_R4 = 1;
VDelay_ms(t);
if (KP_C1 == 0) return '7';     // KEY '7' IS PRESSED
if (KP_C2 == 0) return '8';     // KEY '8' IS PRESSED
if (KP_C3 == 0) return '9';     // KEY '9' IS PRESSED
VDelay_ms(t);
KP_R1 = 1;      // SCAN KEYPAD ON THIRD ROW: 1, 2, 3, A
KP_R2 = 1;      KP_R3 = 0;     KP_R4 = 1;
VDelay_ms(t);
if (KP_C1 == 0) return '4';     // KEY '4' IS PRESSED
if (KP_C2 == 0) return '5';     // KEY '5' IS PRESSED
if (KP_C3 == 0) return '6';     // KEY '6' IS PRESSED
VDelay_ms(t);
KP_R1 = 1;      // SCAN KEYPAD ON FOURTH ROW: 1, 2, 3, A
KP_R2 = 1;      KP_R3 = 1;     KP_R4 = 0;
VDelay_ms(t);
if (KP_C1 == 0) return '1';     // KEY '1' IS PRESSED
if (KP_C2 == 0) return '2';     // KEY '2' IS PRESSED
if (KP_C3 == 0) return '3';     // KEY '3' IS PRESSED
VDelay_ms(t);
return 0xFF;                                    // IF NO KEY PRESS, RETURN 0xFF
}                                                 // KEYPAD READ FUNCTION - END

// MAIN FUNCTION - START
void main(){
ADCON1 = 0x06;                                                  // ALL I/O AS DIGITAL
TRISB.RB0 = 1;    TRISB.RB1 = 1;    TRISB.RB2 = 1;              // C1, C2, C3
TRISD.RD0 = 0;    TRISD.RD1 = 0;    TRISD.RD2 = 0;    TRISD.RD3 = 0;  // L1-4
// IT IS AN UNSIGNED CHAR VARIABLE USED TO STORE THE DATA GET FROM THE 4x4 KP
// INITIALIZE THE PORTC<7:4> AS OUTPUT, PORTC<3:0> AS INPUT
```

```
 71: LCD_init();
 72: Lcd_Cmd(_LCD_CURSOR_OFF);
 73: Lcd_Cmd(_LCD_CLEAR);
 74: while(1){
 75: // WHILE LOOP USED IN HERE IN ORDER THE MICRO-CONTROLLER CAN ALWAYS READ THE
 76: // DATA FROM THE KEYPAD
 77: LCD_Out(1,1,"PRESSIONAR TECLA");
 78: // DISPLAY ENTER WORD ON THE FIRST ROW OF THE 16x2 LCD
 79: LCD_cmd(_LCD_SECOND_ROW);    // GO TO 2ND ROW OF THE 16x2 LCD
 80: for(i = 0; i < 17; i++) {
 81: //THIS LOOP USED IN ORDER FOR THE LCD TO DISPLAYED 16 CHARACTERS
 82: key_get = keypad_wait();
 83: //THE VARIABLE KEY_GET WILL STORE THE DATA FROM keypad_wait() FUNCTION.
 84: Lcd_Chr_Cp(key_get);
 85: // THE LCD WILL DISPLAY THE CHARACTER OF THE KEYPAD AS YOU PRESSED THE BUTTON.
 86: // IT WILL SHOWED 16 CHARACTERS AND IF YOU PRESSED THE KEYPAD FURTHER MORE,
 87: // THE 16   CHARACTERS WILL BE CLEAR AND STARTED AGAIN FROM THE FIRST COLUMN
 88: // OF THE 2ND ROW ON THE16X2 LCD.
 89: }                                                                    // END FOR
 90: LCD_cmd(_LCD_CLEAR);                                          // CLEAR THE LCD
 91: }                                                                  // END WHILE
 92: }                                                                   // END MAIN
 93: // MAIN FUNCTION - END
 94:
 95: // KEYPAD WAIT FINCTION - START
 96: unsigned char c_pressed_key = 0xFF;                        // THE PRESSED KEY!
 97: unsigned char keypad_wait(void){
 98: c_pressed_key = 0xFF;
 99:                                                    // WAIT UNTIL THE KEY IS PRESSED!
100: do{c_pressed_key = keypad_read();}
101: while(c_pressed_key == 0xFF);
102:                                                  // WAIT UNTIL THE KEY IS RELEASED!
103: while(keypad_read() != 0xFF);
104: return c_pressed_key;
105: }                                                    // END "keypad_wait" FUNCTION
```