

```
unsigned short kp = 0;
```

```
// Keypad module connections
```

```
char keypadPort at PORTD;
```

```
// End Keypad module connections
```

```
// LCD module connections
```

```
sbit LCD_RS at RB4_bit;
```

```
sbit LCD_EN at RB5_bit;
```

```
sbit LCD_D4 at RB0_bit;
```

```
sbit LCD_D5 at RB1_bit;
```

```
sbit LCD_D6 at RB2_bit;
```

```
sbit LCD_D7 at RB3_bit;
```

```
sbit LCD_RS_Direction at TRISB4_bit;
```

```
sbit LCD_EN_Direction at TRISB5_bit;
```

```
sbit LCD_D4_Direction at TRISB0_bit;
```

```
sbit LCD_D5_Direction at TRISB1_bit;
```

```
sbit LCD_D6_Direction at TRISB2_bit;
```

```
sbit LCD_D7_Direction at TRISB3_bit;
```

```
// End LCD module connections
```

```
int keyPad(){
```

```
// Wait for key to be pressed and released  
do
```

```
    // kp = Keypad_Key_Press();    //
```

```
Store key code in kp variable
```

```
    kp = Keypad_Key_Click();        // Store
```

```
key code in kp variable
```

```
    while (!kp);
```

```
    // Prepare value for output, transform key  
to it's ASCII value
```

```
    switch (kp) {
```

```
        case 1: return 55; break; // 7    //
```

```
Uncomment this block for keypad4x4
```

```
        case 2: return 56; break; // 8
```

```
        case 3: return 57; break; // 9
```

```
        case 4: return 65; break; // A
```

```
        case 5: return 52; break; // 4
```

```
        case 6: return 53; break; // 5
```

```
case 7: return 54; break; // 6
case 8: return 66; break; // B
case 9: return 49; break; // 1
case 10: return 50; break; // 2
case 11: return 51; break; // 3
case 12: return 67; break; // C
case 13: return 42; break; // *
case 14: return 48; break; // 0
case 15: return 35; break; // #
case 16: return 68; break; // D
```

```
}
```

```
}
```

```
char
```

```
str[10]={'\0','\0','\0','\0','\0','\0','\0','\0','\0','\0'};
```

```
char
```

```
lcdNums[10]={48,49,50,51,52,53,54,55,56,57};
```

```
char
```

```
segNums[10]={0x3F,0x06,0x5B,0x4F,0x66,0
```

```
x6D,0x7D,0x07,0x7F,0x6F};  
int i=0;  
int cnt=0;  
char nums[2];  
void main() {  
    // Reset counter  
    Keypad_Init();           // Initialize  
    Keypad  
    ANSEL = 0;               // Configure  
    AN pins as digital I/O  
    ANSELH = 0;  
    Lcd_Init();              // Initialize LCD  
    Lcd_Cmd(_LCD_CLEAR);    //  
    Clear display  
    Lcd_Cmd(_LCD_CURSOR_OFF);  
    // Cursor off  
    TRISC=0X00;  
    PORTC=0x00;  
    // LCD_OUT(1,1,"WIRTING");  
    while(1){
```

```
while(cnt<sizeof(nums)){  
    char kp1= keypad();  
    nums[cnt]=kp1;  
    LCD_CHR(1,1,kp1);  
    cnt++;  
}  
if((nums[0]==56 && nums[1]==57)  
    || (nums[0]==57 && nums[1]==56))  
{  
    LCD_OUT(1,1,"ZEC & ALUTEYI:");  
    if(i>3)  
        i=0;  
    // max 3  
    if(i>=0 && i<=3){  
        delay_ms(500);  
        LCD_CHR(1,15,lcdNums[i]);  
        portc=segNums[i];  
    }  
}
```

```
}  
    else if((nums[0]==53 && nums[1]==56)  
        || (nums[0]==56 && nums[1]==53)  
    ){  
        LCD_OUT(1,1,"Dao & ZEC:");  
        if(i>5 )  
            {i=3;}  
        // max 3  
        if(i>=3 && i<=5){  
            delay_ms(500);  
            LCD_CHR(1,15,lcdNums[i]);  
            portc=segNums[i];  
        }  
    }  
}
```

```
    else if((nums[0]==57 && nums[1]==53)  
        || (nums[0]==53 && nums[1]==57)  
    ){  
        LCD_OUT(1,1,"ALUTEYI & DA0");  
        if(i<6 || i>8){  
            i=6;
```

```
}
```

```
else{
```

```
delay_ms(500);
```

```
LCD_CHR(1,15,lcdNums[i]);
```

```
portc=segNums[i];
```

```
}
```

```
}
```

```
else if((nums[0]==54 && nums[1]==56)
```

```
|| (nums[0]==54 && nums[1]==56)
```

```
){
```

```
LCD_OUT(1,1,"BAULU & ZEC");
```

```
if(i<7 || i>9){
```

```
i=6;
```

```
}
```

```
if(i>=7 && i<=8){
```

```
delay_ms(500);
```

```
LCD_CHR(1,15,lcdNums[i]);
```

```
portc=segNums[i];
```

```
}
```

```
}
```

```
i++;
```

```
}
```

```
}
```