```
aIn = PORTA;
aIn &= 0x0f;
if (aIn == 0b1110)
    Col = 0;
else if (aIn == 0b1101)
    Col = 1;
else
    Col = 2
else
    Col = -1;
```
LATB = \[ \text{XXXX} \rightarrow \text{X} \]

\[
\text{int rowAct[4]} = \{ \text{0b1110} 0000 0000 \ldots \}
\]

LATB & = \text{rowAct[row]};

LATB | = \text{0x00FF} ;

row[4] = \{ 0b1110 1111 1111 1111 \ldots \}
for (row = 0; row < 4; row++) {
    LATB & = row OFFF;
    LATB l = rowAct [row];
}

if (cd != -1)
    break;
if (col == -1)
    keepIdx = -1;
else
    keepIdx = row*4 + col;

Sym = SymTable[keepIdx];
#define NOKEY 255

unsigned char scan(unsigned short int row)
{
    static unsigned char LUT[] = {NOKEY, NOKEY, NOKEY, NOKEY, NOKEY, NOKEY, NOKEY, 3,
                                  NOKEY, NOKEY, NOKEY, 2, NOKEY, 1, 0, NOKEY};

    unsigned char key;

    key = PORTA & 15;
    key = LUT[key];
    if (key == NOKEY) return key;
    key += 4*row;
    return key;
}
// KBMASK1 is used to mask off the bits in PORTB we don't use (RB12-RB0)
#define KBMASK1 0x0fff
unsigned int keyRowEnPattern[4] = {0xe000, 0xd000, 0xb000, 0x7000};

unsigned char lookupTable[16] = {1, 2, 3, 0xA, 4, 5, 6, 0xB, 7, 8, 9, 0xC, 0xe, 0, 0xf, 0xD}; // these numbers need to be further translated
     // to the actual a,b,...,g,dp pattern of the 7-seg LED
     // 0xe translates to '*' and 0xf translates to '#'

LATB = LATB & KBMASK1;
LATB |= keyRowEnPattern[row];
// not to be used directly in their lab, but students can get ideas...

void waitOneMillisecondAndCheckKeypad(void)
{
    static unsigned short int row = 0;
    static unsigned char lastKey[] = {NOKEY,NOKEY,NOKEY,NOKEY};
    static unsigned short int debounce = 0;
    unsigned char key;

    // If you haven't covered timers yet, the following two lines create a
    // delay of 1ms
    while (!IFS0bits.T1IF);
    IFS0bits.T1IF = 0;

    if (debounce)
    {
        debounce--;
        return;
    }
    row++;
    row &= 3;
    LATB = LATB & KBMASK1;
    LATB |= keyRowEnPattern[row];

    key = scan(row);
}
IFS0bits.T1IF = 0;

if (debounce)
{
    debounce--; 
    return;
}
row++; row &= 3;
LATB = LATB & KBMASK1;
LATB |= keyRowEnPattern[row];

key = scan(row);
if (key != lastKey[row])
{
    lastKey[row] = key;
    if (key != NOKEY)
    {
        // for nonstandard encoding: put( lookupTable[key]);
        put(key);
    }
    debounce = 40; // to give us 40ms debouncing delay (more than enough)
}