Char Template B/chart

Purpose

To decide whether letters are upper or lower case, and convert between the two.

---

§**1. Char Is Of Case.** The following decides whether a character `c` belongs to case `case`, where 0 means lower case and 1 means upper. `c` is interpreted according to the character casing chart in "UnicodeData.i6t", which means, it will be ZSCII for the Z-machine and Unicode for Glulx.

```
[ CharIsOfCase c case
    i tab min max len par;
    if (c<'A') rfalse;
    if (case == 0) {
        if ((c >= 'a') && (c <= 'z')) rtrue;
        tab = CharCasingChart0;
    } else {
        if ((c >= 'A') && (c <= 'Z')) rtrue;
        tab = CharCasingChart1;
    }
    if (c<128) rfalse;
    while (tab-->i) {
        min = tab-->i; i++;
        len = tab-->i; i++;
        i++;
        par = 0;
        if (len<0) { par = 1; len = -len; }
        if (c < min) rfalse;
        if (c < min+len) {
            if (par) { if ((c-min) % 2 == 0) rtrue; }
            else { rtrue; }
        }
    }
    rfalse;
];
```

§**2. Char To Case.** And the following converts character `c` to the desired case, or returns it unchanged if it is not a letter with variant casings.

```
[ CharToCase c case
    i tab min max len par del f;
    if (c<'A') return c;
    if (case == 1) {
        if ((c >= 'a') && (c <= 'z')) return c-32;
        tab = CharCasingChart0;
    } else {
        if ((c >= 'A') && (c <= 'Z')) return c+32;
        tab = CharCasingChart1;
    }
    if (c<128) return c;
    while (tab-->i) {
        min = tab-->i; i++;
        len = tab-->i; i++;
        del = tab-->i; i++;
        par = 0;
        if (len<0) { par = 1; len = -len; }
        if (c < min) return c;
        if (c < min+len) {
            f = false;
            if (par) { if ((c-min) % 2 == 0) f = true; }
            else { f = true; }
            if (f) {
                if (del == UNIC_NCT) return c;
                return c+del;
            }
        }
    }
    return c;
];
```

§**3. Reversing Case.** It's convenient to provide this relatively fast routine to reverse the case of a letter since this is an operation used frequently in regular expression matching (see "RegExp.i6t").

```
#IFDEF TARGET_ZCODE;
[ IT_RevCase ch;
    if (ch<'A') return ch;
    if ((ch >= 'a') && (ch <= 'z')) return ch-'a'+'A';
    if ((ch >= 'A') && (ch <= 'Z')) return ch-'A'+'a';
    if (ch<128) return ch;
    if ((ch >= 155) && (ch <= 157)) return ch+3; ! a, o, u umlaut in ZSCII
    if ((ch >= 158) && (ch <= 160)) return ch-3; ! A, O, U umlaut
    if ((ch >= 164) && (ch <= 165)) return ch+3; ! e, i umlaut
    if ((ch >= 167) && (ch <= 168)) return ch-3; ! E, I umlaut
    if ((ch >= 169) && (ch <= 174)) return ch+6; ! a, e, i, o, u, y acute
    if ((ch >= 175) && (ch <= 180)) return ch-6; ! A, E, I, O, U, Y acute
    if ((ch >= 181) && (ch <= 185)) return ch+5; ! a, e, i, o, u grave
    if ((ch >= 186) && (ch <= 190)) return ch-5; ! A, E, I, O, U grave
    if ((ch >= 191) && (ch <= 195)) return ch+5; ! a, e, i, o, u circumflex
    if ((ch >= 196) && (ch <= 200)) return ch-5; ! A, E, I, O, U circumflex
```

```
    if (ch == 201) return 202; ! a circle
    if (ch == 202) return 201; ! A circle
    if (ch == 203) return 204; ! o slash
    if (ch == 204) return 203; ! O slash
    if ((ch >= 205) && (ch <= 207)) return ch+3; ! a, n, o tilde
    if ((ch >= 208) && (ch <= 210)) return ch-3; ! A, N, O tilde
    if (ch == 211) return 212; ! ae ligature
    if (ch == 212) return 211; ! AE ligature
    if (ch == 213) return 214; ! c cedilla
    if (ch == 214) return 213; ! C cedilla
    if (ch == 215 or 216) return ch+2; ! thorn, eth
    if (ch == 217 or 218) return ch-2; ! Thorn, Eth
    if (ch == 220) return 221; ! oe ligature
    if (ch == 221) return 220; ! OE ligature
    return ch;
];
#IFNOT;
[ IT_RevCase ch;
    if (ch<'A') return ch;
    if ((ch >= 'a') && (ch <= 'z')) return ch-'a'+'A';
    if ((ch >= 'A') && (ch <= 'Z')) return ch-'A'+'a';
    if (ch<128) return ch;
    if (CharIsOfCase(ch, 0)) return CharToCase(ch, 1);
    if (CharIsOfCase(ch, 1)) return CharToCase(ch, 0);
    return ch;
];
#ENDIF;
```

§**4. Testing.** Not actually used: simply for testing the tables.

```
[ CharTestCases case i j;
    for (i=32: i<$E0; i++) {
        if ((i>=127) && (i<155)) continue;
        print i, " - ", (char) i, " -";
        if (CharIsOfCase(i, 0)) print "  lower";
        if (CharIsOfCase(i, 1)) print "  upper";
        j = CharToCase(i, 0); if (j ~= i) print "  tolower: ", (char) j;
        j = CharToCase(i, 1); if (j ~= i) print "  toupper: ", (char) j;
        print "^";
    }
];
```