

```

#include <sofa.h>
void reprd ( char*, double, double );
int main ()
{

#define DAS2R (4.848136811095359935899141e-6) /* arcsec to radians */
iauASTROM astrom;
iauLDBODY b[3];
double phi, elong, hm, phpa, tc, rh, wl, utc1, utc2, tail, tai2,
tt1, tt2, xp, yp, dut1, dx, dy, rc, dc, pr, pd, px, rv,
rca, dca, ri, di, eo, ra, da, aot, zot, hot, dot, rot,
aob, zob, hob, dob, rob, pvh[2][3], pvb[2][3], r[3][3],
x, y, s;

/* Site longitude, latitude (radians) and height above the geoid (m).
*/
iauAf2a ( '-', 5, 41, 54.2, &elong );
iauAf2a ( '-', 15, 57, 42.8, &phi );
hm = 625.0;

/* Ambient pressure (HPa), temperature (C) and rel. humidity (frac). */
phpa = 952.0;
tc = 18.5;
rh = 0.83;

/* Effective color (microns). */
wl = 0.55;

/* UTC date. */
if ( iauDtf2d ( "UTC", 2013, 4, 2, 23, 15, 43.55,
&utc1, &utc2 ) ) return -1;

/* TT date. */
if ( iauUtctai ( utc1, utc2, &tail, &tai2 ) ) return -1;
if ( iauTaitt ( tail, tai2, &tt1, &tt2 ) ) return -1;

/* EOPs: polar motion in radians, UT1-UTC in seconds. */
xp = 50.995e-3 * DAS2R;
yp = 376.723e-3 * DAS2R;
dut1 = 155.0675e-3;

/* Corrections to IAU 2000A CIP (radians). */
dx = 0.269e-3 * DAS2R;
dy = -0.274e-3 * DAS2R;

/* Star ICRS RA,Dec (radians). */
if ( iauTf2a ( ' ', 14, 34, 16.81183, &rc ) ) return -1;
if ( iauAf2a ( '-', 12, 31, 10.3965, &dc ) ) return -1;
reprd ( "ICRS, epoch J2000.0:", rc, dc );

/* Annual proper motion: RA/Dec derivatives, epoch J2000.0. */

```

```

pr = atan2 ( -354.45e-3 * DAS2R, cos(dc) );
pd = 595.35e-3 * DAS2R;
/* Parallax (arcsec) and recession speed (km/s). */
px = 164.99e-3;
rv = 0.0;

/* ICRS catalog to astrometric place... */
iauAtcc13 ( rc, dc, pr, pd, px, rv, tt1, tt2, &rca, &dca );
reprd ( "catalog -> astrometric:", rca, dca );

/* ...then to CIRS (geocentric observer). */
iauAtci13 ( rca, dca, 0.0, 0.0, 0.0, 0.0, tt1, tt2, &ri, &di, &eo );
reprd ( "astrometric -> CIRS:", ri, di );

/* ICRS catalog directly to CIRS (geocentric observer). */
iauAtci13 ( rc, dc, pr, pd, px, rv, tt1, tt2, &ri, &di, &eo );
reprd ( "catalog -> CIRS:", ri, di );

/* Apparent place. */
ra = iauAnp ( ri - eo );
da = di;
reprd ( "geocentric apparent:", ra, da );

/* CIRS to topocentric. */
if ( iauAtio13 ( ri, di, utc1, utc2, dut1, elong, phi, hm, xp, yp,
0.0, 0.0, 0.0, 0.0,
&aot, &zot, &hot, &dot, &rot ) ) return -1;
reprd ( "CIRS -> topocentric:", rot, dot );

/* CIRS to observed. */
if ( iauAtio13 ( ri, di, utc1, utc2, dut1, elong, phi, hm, xp, yp, phpa,
tc, rh, wl,
&aob, &zob, &hob, &dob, &rob ) ) return -1;
reprd ( "CIRS -> observed:", rob, dob );

/* ICRS to observed. */
if ( iauAtco13 ( rc, dc, pr, pd, px, rv, utc1, utc2, dut1,
elong, phi, hm, xp, yp, phpa, tc, rh, wl,
&aob, &zob, &hob, &dob, &rob, &eo ) ) return -1;
reprd ( "ICRS -> observed:", rob, dob );

/* ICRS to CIRS using some user-supplied parameters. */

/* SOFA heliocentric Earth ephemeris. */
if ( iauEpv00 ( tt1, tt2, pvh, pvh ) ) return -1;

/* JPL DE405 barycentric Earth ephemeris. */
pvb[0][0] = -0.9741704366519668;
pvb[0][1] = -0.2115201000882231;
pvb[0][2] = -0.0917583114068277;
pvb[1][0] = 0.0036436589347388;

```

```

pvb[1][1] = -0.0154287318503146;
pvb[1][2] = -0.0066892203821059;

/* IAU 2000A CIP. */
iauPnm00a ( tt1, tt2, r );
iauBpn2xy ( r, &x, &y );

/* Apply IERS corrections. */
x += dx;
y += dy;

/* SOFA CIO locator. */
s = iauS06 ( tt1, tt2, x, y );

/* Populate the context. */
iauApci ( tt1, tt2, pvb, pvh[0], x, y, s, &astrom );

/* Carry out the transformation and report the results. */
iauAtciq ( rc, dc, pr, pd, px, rv, &astrom, &ri, &di );
reprd ( "ICRS -> CIRS (JPL, IERS):", ri, di );

/* The same but with Saturn then Jupiter then Sun light deflection. */
b[0].bm = 0.00028574;
b[0].dl = 3e-10;
b[0].pv[0][0] = -7.8101442680818964;
b[0].pv[0][1] = -5.6095668114887358;
b[0].pv[0][2] = -1.9807981923749924;
b[0].pv[1][0] = 0.0030723248971152;
b[0].pv[1][1] = -0.0040699547707598;
b[0].pv[1][2] = -0.0018133584165345;
b[1].bm = 0.00095435;
b[1].dl = 3e-9;
b[1].pv[0][0] = 0.7380987962351833;
b[1].pv[0][1] = 4.6365869247538951;
b[1].pv[0][2] = 1.9693136030111202;
b[1].pv[1][0] = -0.0075581692172088;
b[1].pv[1][1] = 0.0012691372216750;
b[1].pv[1][2] = 0.0007279990012801;
b[2].bm = 1.0;
b[2].dl = 6e-6;
b[2].pv[0][0] = -0.0007121743770509;
b[2].pv[0][1] = -0.0023047830339257;
b[2].pv[0][2] = -0.0010586596574639;
b[2].pv[1][0] = 0.0000062923521264;
b[2].pv[1][1] = -0.0000003308883872;
b[2].pv[1][2] = -0.0000002964866231;
iauAtciqn ( rc, dc, pr, pd, px, rv, &astrom, 3, b, &ri, &di );
reprd ( "ICRS -> CIRS (+ planets):", ri, di );

/* CIRS to ICRS (astrometric). */
iauAticqn ( ri, di, &astrom, 3, b, &rca, &dca );

```

```

reprd ( "CIRS -> astrometric:", rca, dca );
return 0;
}
#include <stdio.h>
void reprd ( char* s, double ra, double dc )
{
    char pm;
    int i[4];
    printf ( "%25s", s );
    iauA2tf ( 7, ra, &pm, i );
    printf ( " %2.2d %2.2d %2.2d.%7.7d", i[0],i[1],i[2],i[3] );
    iauA2af ( 6, dc, &pm, i );
    printf ( " %c%2.2d %2.2d %2.2d.%6.6d\n", pm, i[0],i[1],i[2],i[3] );
}

```