```
// longitude/latitude of observer location
lon=11.42 * %CL_deg2rad;   lat=48.15 * %CL_deg2rad; alt=531;

// scal=reference date (string, calendar)
scal=[];

// Get input data from the user
desc_param=list(..
   CL_defParam("Longitude of location",lon,units=["rad","deg"],..
               valid='$x>=0 & $x <=360'), ..
   CL_defParam("Latitude of location",lat,units=["rad","deg"],..
               valid='$x>=-90 & $x <=90'), ..
   CL_defParam("Altitude of location",alt,units=["m"],..
               valid='$x>=-100 & $x <=3000'), ..
   CL_defParam("Reference time (calendar format, TREF)",scal,typ="cal"));

[lon,lat,alt,scal]=CL_inputParam(desc_param);
loc=[lon;lat;alt];

// Convert calendar date to modified (1950.0) Julian day
t0=CL_dat_cal2cjd(CL_dat_str2cal(scal));

// Get position of the celestial body using JPL's DE405 ephemerides
// The results are given in the ICRS frame
pos_ICRS=CL_eph_de405("Venus",t0,"Earth");

// Transform position of venus from the ICRS frame to the ECI frame
// (Earth Centered Innertial Frame)
pos_ECI=CL_fr_convert("ICRS","ECI",t0,pos_ICRS);

// Transform position of celestial body in ECF frame
// (Earth Centered Fixed Frame)
pos_ECF=CL_fr_convert("ECI","ECF",t0,pos_ECI);

// Transform elliptical coordinates of the position of the observer
// location to cartesian coordinates
loc_car=CL_co_ell2car(loc);

// Get the transformation matrix to the local Topocentric North frame
M_topoN=CL_fr_topoNMat(loc);

// Transform the position of celestial body to the local Topocentric
// North frame in cartesian coordinates as seen by the observer (origin)
pos_topoN=M_topoN*(pos_ECF-loc_car);

// Transform cartesian coordinates of celestial body to spherical
// coordinates
pos_topoN_sph=CL_co_car2sph(pos_topoN);

// Get azimuth and elevation of celestial body
azim=360-pos_topoN_sph(1)*%CL_rad2deg;
elev=pos_topoN_sph(2)*%CL_rad2deg;

disp(azim); disp(elev);
```