1. Write an IDL function called `GEOSTROPHIC` that will accept three floating-point arguments that represent a contour interval, a distance between contours (in kilometers), and a latitude (in degrees), and that will return the geostrophic wind speed. Include two keywords, `/pressure` and `/knots`, that do the following:

   • If the keyword `/pressure` is set then the contour interval is assumed to be in millibars, and the calculation is for the surface wind (use an appropriate surface density). Otherwise, it is meters of geopotential height.
   • If the keyword `/knots` is set the output speed is in knots. Otherwise, it is in meters per second.

   The geostrophic wind speed is given by
   \[
   V_g = \frac{1}{f \rho} \left| \nabla p \right|
   \]
   \[
   V_g = \frac{g}{f} \left| \nabla Z \right|
   \]

2. a. Write an IDL main program called `geo_space.pro` that calls your `GEOSTROPHIC` function and calculates and prints to the screen the geostrophic wind (in knots) for a 4 mb contour interval at 40°N latitude, for contour spacings starting at 1000 km and decreasing in increments of 25 km down to a spacing of 25 km.

   Note: It is not a good idea to pass a loop index variable to a function or procedure, since the value of the loop index may change inside the function or procedure (recall that IDL passes variables by reference rather than value). A better idea is to assign the loop index variable to a new variable, and pass the new variable to the function or procedure.

   b. Write an IDL main program called `geo_latitude.pro` that calculates and prints to the screen the geostrophic wind (in knots) for a 4 mb contour interval and a contour spacing of 200 km for latitudes starting at 90°N and decreasing in 5° intervals down to a latitude of 0°.
3. Write an IDL procedure called GEO_GRADIENT that will accept four floating-point arguments that represent a contour interval, a distance between contours (in kilometers), a radius of curvature (in kilometers), and a latitude (in degrees), and that will return the geostrophic and gradient wind. Include the keywords /knots and /pressure as in the geostrophic function, and an additional keyword /anticyclone that acts as follows:
- If /anticyclone is set, the gradient wind for an anticyclone is calculated. Otherwise, the calculation is for a cyclone.

The gradient wind is given as

\[
\text{Cyclone: } V_{gr} = \frac{f R}{2} \left( \sqrt{1 + \frac{4V_g}{f R}} - 1 \right)
\]

\[
\text{Anticyclone: } V_{gr} = \frac{f R}{2} \left( 1 - \sqrt{1 - \frac{4V_g}{f R}} \right)
\]

where \( V_g \) is the geostrophic wind speed and \( R \) is the radius of curvature (always positive in these formulas).

Notes:
- For anticyclonic flow it is possible for the radical to give an imaginary number (which is unphysical). If this happens, set the gradient wind to NaN.
- Include error checking, so that if negative numbers are entered for any incoming arguments they are automatically converted to positives before doing any calculations.
- Your procedure must call the GEOSTROPHIC function to calculate the geostrophic wind.

Extra Credit (2 pt.): Add an additional keyword called /ANOMALOUS that if set will calculate the speed of the anomalous circulation. The gradient wind for the anomalous cases is

\[
\text{Anomalous Cyclone: } V_{gr} = \frac{f R}{2} \left( \sqrt{1 + \frac{4V_g}{f R}} + 1 \right)
\]

\[
\text{Anomalous Anticyclone: } V_{gr} = \frac{f R}{2} \left( 1 + \sqrt{1 - \frac{4V_g}{f R}} \right)
\]