3 Methods for Polygon Storage

1. Explicit - oldest and simplest, stored as an array of polygons
   a) no  b) yes  c) no  d) yes
   Inefficiency - redundancy in points

2. Pointer to Vertex List - space efficient
   a) yes  b) yes  c) no  d) yes
   Most efficient storage

3. Pointer to Edge List (aka Explicit Edge) - less efficient
   a) yes  b) no  c) yes  d) yes
   Smaller and less vertices

→ 1. \[ P_1 = (x_1,y_1,z_1, x_2,y_2,z_2, ... x_n,y_n,z_n) \]
   \[ P_2 = (x_2,y_2,z_2, ... x_n,y_n,z_n) \]
   Sides are straight lines, has area, is flat
   Must be closed, at least 3 points

→ 2. \[ V_{\text{LIST}} = (x_1,y_1,z_1, x_2,y_2,z_2, ... x_n,y_n,z_n) \]
   \[ P_1 = (V_1, V_2, V_3, ..., V_n) \]
   Every vertex is mesh linked once
   \( V_i \) - index into a vertex list
   Traded off ease for storage efficiency

→ 3. \[ V_{\text{LIST}} = (x_1,y_1,z_1, x_2,y_2,z_2, ... x_n,y_n,z_n) \]
   \[ E_1 = (V_1, V_2, P_1, P_2) \]
   \( E_1 \) - 4 ints
   \[ P_1 = (E_1, E_2, E_3) \]
   \( P_1 \) - variable
   Vertices and polygons if outside \( P_2 = \emptyset \)
   (belongs to either 1 or 2 poly)
   \( \text{poly} \) = series of edges

no method for dealing with shared vertices!