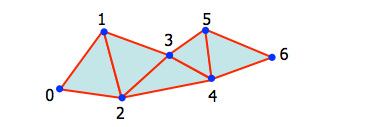
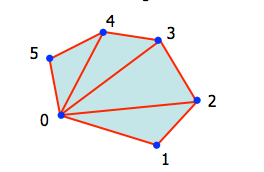
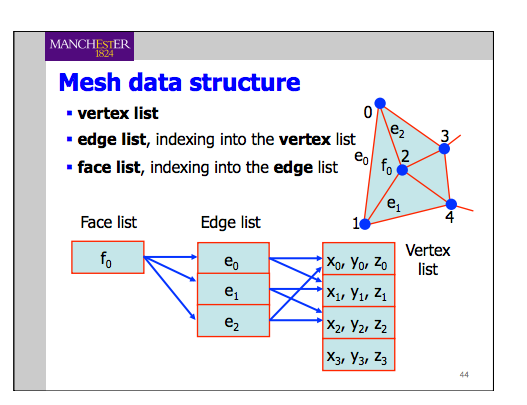
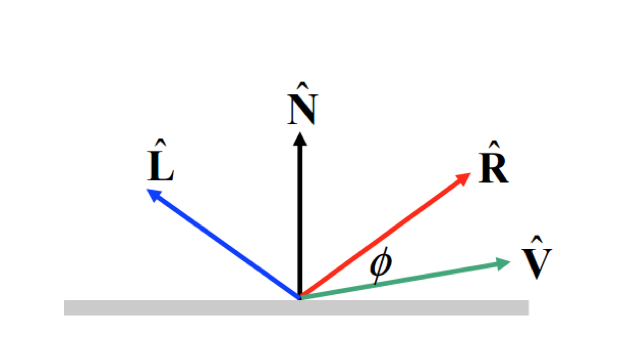
Question 1:

1. Because triangles are always planer, planer is when all vertices lie on the plane that many graphical algorithm relies upon this property.
2. Triangle strips: N linked triangle can be represented with N + 2 vertices as opposite of 3N if each triangle was drawn separately  
      
   Triangle fan: N linked triangle can be represented with N + 2 vertices as opposite of 3N if each triangle was drawn separately  
   
3. Orientation, face and back.  
   Used in illumination where we can multiply the surface normal matrix with direction of light to get the angle of light..   
   It has a lot of other usage this is one of them.
4. Find the vector of AB edge, inverse it calling it AB-1   
   Find vector of BC edge.  
   Normalise the vectors  
   And their cross product is the surface normal.  
   N = AB-1 X BC.
5. 1. Tracking the face, edge and vertices of each polygon, to allow easy editing.
   2. 
   3. I would have struct mesh {  
      faceList\* faceList;  
      edgeList\* edgeList;  
      vertexList\* vertexList;  
      }  
      Where I have struct faceList {  
      edgeList\* adjEdges;  
      }  
      Struct edgeList {  
      vertexList\* vertexList  
      }  
      Struct vertexList {  
      Int x;  
      Int y;  
      Int z;  
      }

Question 2:

1. Modelling: Is the composition of the 3 dimensional space, skeleton of the scene.  
   Rendering: Is the process of taking a scene and converting it to an object.   
   Taking care of the interaction between objects in the scene such as lighting, reflection and so on.  
   It is useful to draw a distinguishment between them as rendering work on what and how we see the scene, modelling is how the computer going to see it.
2. 1. Old approach was trying to workout the visibility of a surface in the world space, by working out automatically in 3D space and drawing the result.  
       This extremely hard.  
      The new approach work out on the display space, where during scan-conversion step, we determine if the generated pixel P is hiding behind any other pixel, if it is don’t draw it, otherwise draw it.
   2. During the scan-conversion step, whenever we generate pixel P we check whether there is another object nearer to the eye also maps to P.
   3. Not always, due to lack of precision in the z-buffer which leads to incorrect rendering of pixels with similar z-values. There are ways to correct it for example in opengl: glPolyOffset() which set the precision offset to certain number.
3. Illumination model :   
   i = iaka + ip/d’ [kd(N.L) + ks(R.V)n]  
     
   Ia - constant ambient lighting intensity   
   Ka - ambient reflection coefficient   
   Ip - intensity of source   
   D’ - takes the effect of distance, d’ = kc + kld + kqd2   
   Kd - diffuse reflection coefficient   
   N - the surface normal  
   L - the direction of light source from surface  
   Ks - specular coefficient, an approximated inbound angle and wavelength   
   R - reflected outbound   
   V - Viewer position relative to surface



Question 3  
NOT SURE AT ALL FROM THIS QUESTION

1. This noise is introduced by the scanning algorithm, and can be amended using smoothing methods, such as the median filter.
2. Smoothing operation such as Median filter.
3. Connected component analysis, then calculating the perimeter, orientation and area and using central moment of area and chain codes storing that information in our training data.  
     
   Any new tests would be compared against that the training data and given a label accordingly.  
     
   This method sensitive to noise and expensive computationally, however it should yield a good classification results.
4. The noise in the image
5. We will need to enhance the edges using sharpening method, then apply edge detection kernel such as sobel.   
   It would be faster and more reliable, disadvantage it wouldn’t look nice to human eyes.

Question 4

Have I said I hate this module? Because I REALLY DO, god I hope tomorrow goes well...