I don’t know what to say here.   
Usage of code is the same as sample solution, no more information needed.  
  
  
Tech:  
With in the *Draw\_Cell()* function suggested in the spec, we need to have the following:  
First, we need to know how to clip an edge to frustum, steps as follows.

1. **Define the Frustum**:

The frustum is represented by two rays (leftRay and rightRay) originating from a common starting point (the viewer's position).

1. **Determine Segment’s Points’ Position**:  
   A sample point in front of the viewer, positioned within the frustum should be in the same *quadrant* of the two axes. Check this by applying cross product.
2. **Compute Intersections**:  
   compute intersects by the method that check whether the two lines intersect within the range of parameters of parametric lines.
3. **Handle Different cases**:  
   for different cases, the clip result may be the endpoints or intersects. Return the corresponding result.

Second, we need to draw the clipped edge on screen. Assume that the edge is (P1, P2). The wall we want to draw is : [(P1.x, P1.y, -1), (P1.x, P1.y, 1), (P2.x, P2.y, 1), (P2.x, P2.y, -1)].   
Originally, we can call:  
*glMatrixMode(GL\_PROJECTION);*  
*glLoadIdentity(); // Reset projection matrix*  
*gluPerspective(verticalFOV, viewAspect, 0.001f, 1000.0f);*  
*glMatrixMode(GL\_MODELVIEW);  
glLoadIdentity();*  
*gluLookAt(viewerPos.x, viewerPos.y, 0.0f, viewerPos.x + cos(rotationRad), viewerPos.y + sin(rotationRad), 0.0f, 0.0f, 0.0f, 1.0f);*

But in this task, we should implement this manually. This is not that hard. For details, look into my *TransformAPI* class.