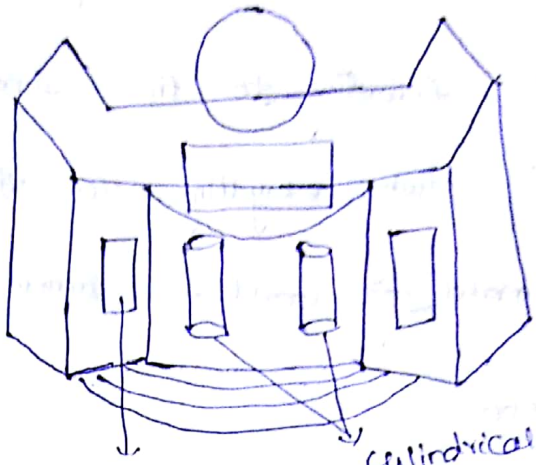
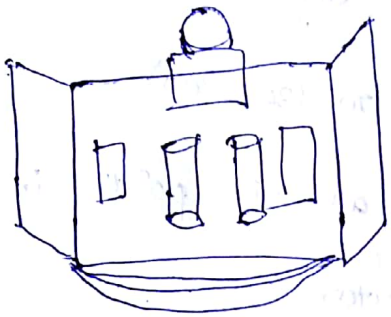
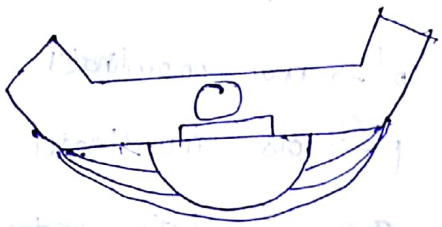
b) a) cylinder, rectangles, sphere



front view :-



top view



c)

a) Create models:- Design asphere two cylinders and then 3 rectangular surfaces. sphere of globe, cylinder for beams, 2 rectangular surfaces for LED surfaces and rectangular surface for banner.

b) world:- import each model into the world i.e; portico here and adjust its positions in accordance with

require merits and adjust the size.

c) Camera:- Adjust the camera to get a proper view and set it to a suitable position.

d) Viewport:- According to the camera set a view port

e) Screen:- Flush everything to the screen

create model → coord → camera → viewport → screen

d) 600 × 600

initially beams are at origin.

1. beam 1 is already at origin, we need not transform it.

To get beam 2 to (300, 300)

translate all the points in cylinder by adding $\begin{bmatrix} 300 \\ 300 \end{bmatrix}$ translation vector

So for all p in cylinder $p' = p + T$

where p' → new coordinates

p → old coordinates

T → translation vector.

2. Translation

3. In normal coordinates it is as simple as $p' = p + T$

In homogeneous coordinates, it is

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

x', y' → new coordinates

x, y → old coordinates

$(t_x, t_y) = (300, 300)$

4) the canvas size is in ratio of 1:1 its better to choose ratio of 1:1

f) → Aspect ratio

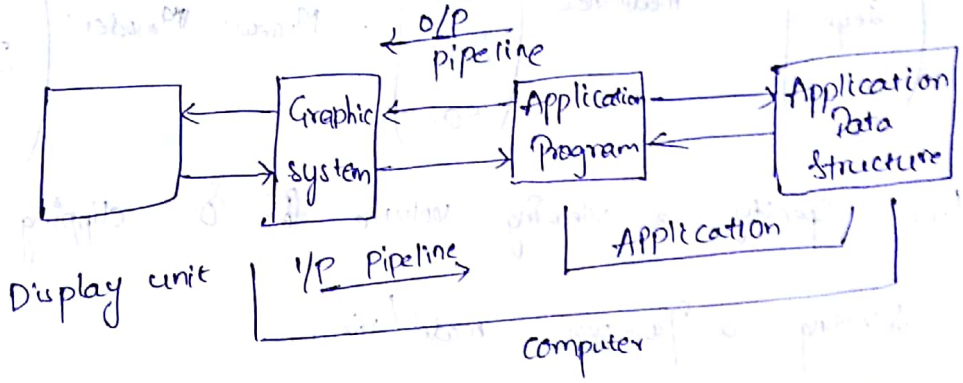
→ camera position.

→ Depth of field

→ lighting and exposure

→ field of view

e)



After the rendering process, a vector image is produced which is composed of points and paths rather than pixels this image contains the image of building either in top view or front view defined with all the requirements ie, two cylindrical beams, globe and flat surfaces in appropriate places

g) this can be done by ray tracing which is a rendering technique for generating an image by tracing the path of the light as pixels in an image plane simulating the effects of its encounters with virtual objects.

a)
$$\begin{pmatrix} x_{world} \\ y_{world} \\ z_{world} \\ 1 \end{pmatrix} = M_{model} \begin{pmatrix} x_{object} \\ y_{object} \\ z_{object} \\ 1 \end{pmatrix}$$

where M_{model} performs the appropriate coordinate change (rotation, translation, scaling) to each vector. then transform the vertices from world space to the eye or camera space we need other matrix (M_{view}) to apply this transformation.

$$\begin{pmatrix} x_{eye} \\ y_{eye} \\ z_{eye} \\ w_{eye} \end{pmatrix} = M_{modelview} \cdot \begin{pmatrix} x_{obj} \\ y_{obj} \\ z_{obj} \\ w_{obj} \end{pmatrix} = M_{view} \cdot M_{model} \cdot \begin{pmatrix} x_{obj} \\ y_{obj} \\ z_{obj} \\ w_{obj} \end{pmatrix}$$

then specify a viewing volume & clipping volume and selecting a Projection. model/view

$$\begin{pmatrix} x_{clip} \\ y_{clip} \\ z_{clip} \\ w_{clip} \end{pmatrix} = M_{projection} \begin{pmatrix} x_{eye} \\ y_{eye} \\ z_{eye} \\ w_{eye} \end{pmatrix}$$

b) By using glpushmatrix() and glpopmatrix(), we can apply transformation on beam 2 without effecting beam1 and other objects. push matrix saves the current coordinate system in stack where as popmatrix restores it.

c) No, it can't be done unless you rasterise the picture. Because any operation such as color correction, adding textures etc, can be done only through pixels which is a primitive of raster picture. this can be easily found in Adobe photoshop

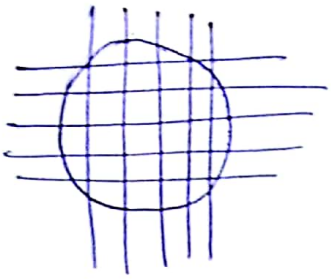
where you rasterise picture for apply some correction and all.

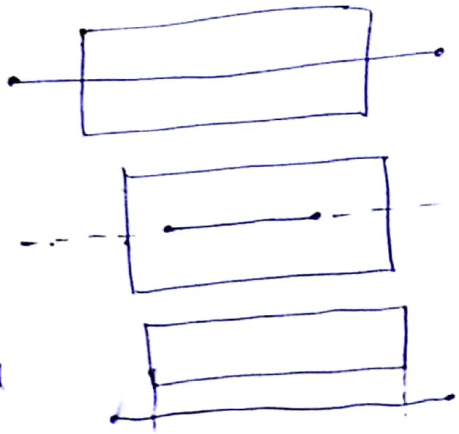
j) when we apply culling to beam 1, it has no effect on beam 2, but beam 1 will be completely excluded out of Pipeline

m) RGB stands for Red, green and blue and ranges from 0 to 255 for R, G & B

(1) (0,0,0) → Beams will be filled with black color

(11) (255,255,255) → Beams will be filled with color.





Partially visible :- Both intersect and both outside the window or one inside the window

Fully visible :- both inside the window

Not visible :- intersects outside window

k) since clipping is cutting out a position of an object. after clipping beams, you can only see beams.