1) \[ A = P_1 + t_1 (P_2 - P_1) \]

\[
\begin{align*}
X_a &= 14 + t_1 \left( 30 - 14 \right) \\
20 &= 18 + t_1 \left( 32 - 18 \right) \\
X_a &= 14 + \left( \frac{1}{7} \right) \left( 16 \right) = \frac{98 + 16}{7} = \frac{114}{7} \\
20 &= \frac{114}{7} + t_1 \left( 32 - 18 \right) \\
\Rightarrow t_1 &= \frac{5}{7} \\
8 &= 20 + t_2 \left( 30 - 20 \right) \\
X_b &= 42 + t_2 \left( 30 - 42 \right) \\
X_b &= 42 + \left( \frac{3}{5} \right) \left( -12 \right) = \frac{208 - 24}{5} = \frac{184}{5} \\
B &= P_3 + t_2 \left( P_2 - P_3 \right) \\
\end{align*}
\]

\[
\begin{align*}
X_a &= 14 + t_1 \left( 30 - 14 \right) \\
20 &= 18 + t_1 \left( 32 - 18 \right) \\
X_a &= 14 + t_1 \left( 30 - 14 \right) \\
X_b &= 42 + t_2 \left( 30 - 42 \right) \\
X_b &= 42 + \left( \frac{3}{5} \right) \left( -12 \right) = \frac{208 - 24}{5} = \frac{184}{5} \\
\end{align*}
\]

\[
P = A + t_3 \left( B - A \right) \\
\begin{align*}
\left( \begin{array}{c}
20 \\
20
\end{array} \right) &= \left( \begin{array}{c}
\frac{114}{7} \\
\frac{184}{5}
\end{array} \right) + t_3 \left( \begin{array}{c}
\frac{186}{5} - \frac{114}{7} \\
\frac{186}{5} - \frac{114}{7}
\end{array} \right) \\
20 &= \frac{114}{7} + t_3 \left( \frac{186}{5} - \frac{114}{7} \right) \\
\Rightarrow t_3 &= \frac{81}{7} - \frac{93}{35} = \frac{365}{732} = \frac{5}{12} \\
X_a &= 52 + \left( \frac{1}{7} \right) \left( 10 - 52 \right) = 52 + \left( \frac{3}{5} \right) \left( 10 - 32 \right) \\
X_b &= 32 + \left( \frac{2}{5} \right) \left( 10 - 32 \right) = 32 + \frac{116}{5} \\
46 &= 46 + \left( \frac{5}{12} \right) \left( \frac{116}{5} - 46 \right) \\
&= 46 + \left( \frac{5}{12} \right) \left( \frac{116}{5} - 46 \right) \\
&= 46 + \left( \frac{5}{12} \right) \left( \frac{116}{5} - 46 \right) \\
&= 46 + \left( \frac{17}{2} \right) = \frac{73}{2} \\
Z &= \left( P_2 - P_1 \right) \times \left( P_2 - P_1 \right) \\
&= \left( \begin{array}{cc}
30 - 14 & 142 - 14 \\
32 - 18 & 12 - 12 \\
10 - 52 & 32 - 32
\end{array} \right) \\
&= \left( \begin{array}{cc}
16 & 28 \\
14 & -20 \\
-42 & -20
\end{array} \right) \\
&= \left( \begin{array}{c}
14(20) - (16)(42) \\
(92)(28) - (14)(16) \\
(16)(-6) - (14)(18)
\end{array} \right) \\
&= \left( \begin{array}{c}
-532 \\
-856 \\
-488
\end{array} \right) \\
\|n\| = \sqrt{(-532)^2 + (-856)^2 + (-488)^2} = 1120 \\
W/\|n\| &= \left( \begin{array}{c}
-532/1120 \\
-856/1120 \\
-488/1120
\end{array} \right) = \left( \begin{array}{c}
-\frac{133}{280} \\
-\frac{103}{1120} \\
-\frac{61}{140}
\end{array} \right)
\[ n \cdot (P - P_1) = 0 \]

\[
\begin{pmatrix}
-133/280 \\
-137/140 \\
-61/140
\end{pmatrix} \cdot \begin{pmatrix}
25 - 14 \\
70 - 18 \\
2 - 52
\end{pmatrix} = \begin{pmatrix}
-133/280 \\
-137/140 \\
-61/140
\end{pmatrix} \cdot \begin{pmatrix}
11 \\
2 \\
2 - 52
\end{pmatrix}
\]

\[ = \left( \frac{-133}{280} \right) \cdot (11) + \left( \frac{-137}{140} \right) \cdot (2) + \left( \frac{-61}{140} \right) \cdot (2 - 52) = 0 \]

\[ = \frac{189}{280} - \frac{31}{2} = \frac{2}{2} \]

\[ z = \frac{-31}{2} + 52 = \frac{43}{2} \]

---

**Lemma 1**

\[ A = P_1 + t_1 (P_2 - P_1) \]

\[ B = P_3 + t_2 (P_3 - P_2) \]

\[ P = A + t_3 (B - A) \]

\[ A = (1-t_1) P_1 + t_1 P_2 \]

\[ B = (1-t_2) P_3 + t_2 P_2 \]

\[ P = (1-t_3) A + t_3 B \]

\[ = (1-t_3) (1-t_1) P_1 + (1-t_3) t_1 P_2 + t_3 (1-t_2) P_3 + t_3 t_2 P_2 \]

\[ = \left( \frac{1}{2}, \frac{1}{2} \right) P_1 + \left[ \left( 1 - \frac{1}{2} \right) \left( 1 - \frac{1}{2} \right) \right] P_2 + \left( \frac{1}{2} \left( 1 - \frac{1}{2} \right) \right) P_3 \]

\[ = \frac{1}{2} P_1 + \frac{1}{4} P_2 + \frac{1}{4} P_3 \]

\[ \alpha = \frac{1}{2}, \beta = \frac{1}{2}, \gamma = \frac{1}{2} \]

**\( Z_1 \)**

\[ = \begin{pmatrix}
0 & 0 & 0 & 0
\end{pmatrix} \cdot \begin{pmatrix}
1 & 2 & 3 & 4
\end{pmatrix}
\]

\[ = \frac{1}{2} (52) + \frac{1}{4} (10) + \frac{1}{4} (32) 
\]

\[ = 26 + \frac{5}{2} + 8 = \frac{72}{2} 
\]

\[ = 0.5 + 0.75 = 0.575 \]

**\( Y_P \)**

\[ = \begin{pmatrix}
0 & 0 & 0 & 0
\end{pmatrix} \cdot \begin{pmatrix}
1 & 2 & 3 & 4
\end{pmatrix}
\]

\[ = \frac{1}{2} (1) + \left( \frac{1}{2} \right) (0.3) + \frac{1}{4} (0) 
\]

\[ = 0.5 + 0.05 + 0.125 = 0.675 
\]

**\( G_P \)**

\[ = \begin{pmatrix}
0 & 0 & 0 & 0
\end{pmatrix} \cdot \begin{pmatrix}
1 & 2 & 3 & 4
\end{pmatrix}
\]

\[ = \left( \frac{1}{2} \right) (1) + \left( \frac{1}{2} \right) (0.3) + \left( \frac{1}{4} \right) (0.5) 
\]

\[ = 0.5 + 0.05 + 0.125 = 0.675 
\]

**\( b_P \)**

\[ = \begin{pmatrix}
0 & 0 & 0 & 0
\end{pmatrix} \cdot \begin{pmatrix}
1 & 2 & 3 & 4
\end{pmatrix}
\]

\[ = \frac{1}{2} (1) + \left( \frac{1}{2} \right) (0.3) + \left( \frac{1}{4} \right) (0.5) 
\]

\[ = 0.5 + 0.05 + 0.125 = 0.675 
\]
3. Another method

\[ P = \alpha P_1 + \beta P_2 + \gamma P_3 \]

\[
\begin{pmatrix}
    X_P \\
    Y_P \\
    Z_P
\end{pmatrix} =
\begin{pmatrix}
    X_1 \\
    Y_1 \\
    Z_1
\end{pmatrix} + \beta
\begin{pmatrix}
    X_2 \\
    Y_2 \\
    Z_2
\end{pmatrix} + \gamma
\begin{pmatrix}
    X_3 \\
    Y_3 \\
    Z_3
\end{pmatrix}
\]

We have

\[ X_P = \alpha X_1 + \beta X_2 + \gamma X_3 \]
\[ Y_P = \alpha Y_1 + \beta Y_2 + \gamma Y_3 \]
\[ \delta = 1 - \alpha - \beta \]

\[ X_P = \alpha X_1 + \beta X_2 + (1 - \alpha - \beta) X_3 \]
\[ = \alpha X_1 + \beta (X_2 - X_3) + X_3 \]
\[ Y_P = \alpha Y_1 + \beta (Y_2 - Y_3) + Y_3 \]

\[ 25 = \alpha (14 - 42) + \beta (30 - 42) + 42 \Rightarrow 25 = -28\alpha - 12\beta + 42 \]
\[ 20 = \alpha (18 - 12) + \beta (32 - 72) + 12 \Rightarrow 20 = 6\alpha + 20\beta + 12 \]

\[ 28\alpha + 12\beta = 17 \Rightarrow 14\alpha + 6\beta = 83 \]
\[ 6\alpha + 30\beta = 8 \Rightarrow 18\alpha + 90\beta = 24 \]

\[ 132\alpha = 61 \Rightarrow \alpha = \frac{1}{2} \]

\[ 6 \left( \frac{1}{2} \right) + 30\beta = 8 \Rightarrow 3 + 30\beta = 8 \Rightarrow 30\beta = 5 \Rightarrow \beta = \frac{1}{5} \]
\[ \delta = 1 - \alpha - \beta = 1 - \frac{1}{2} - \frac{1}{5} = \frac{1}{10} \]

\[ \alpha = \frac{1}{2}, \beta = \frac{1}{5}, \delta = \frac{1}{10} \]
5. We draw the objects first.

\[
\begin{align*}
&h, \\
&\quad \quad a \\
&\quad \quad \quad b \\
&\quad \quad \quad \quad c \\
&\quad \quad \quad \quad \quad d \\
&\quad \quad \quad \quad \quad f \\
&\quad \quad \quad \quad \quad \quad g \\
&\quad \quad \quad \quad \quad \quad \quad i
\end{align*}
\]

where

\[
\begin{align*}
&h_1, \\
&\quad \quad h_2
\end{align*}
\]

Final order: \( f g e h_2 b h_1 i d c a \)
The gamut of CMY is shown in left figure. Since monitor is additive color mixing, the gamut is just the triangle of points C, M and Y. This would not make a good monitor because

1. The gamut of CMY is much smaller than that of RGB
2. CMY can't make pure Red, Green or Blue colors, which are very important for display.

BSP tree will choose a polygon as root, then split the other two polygons and add parts to the front/behind list.

For example, we choose polygon A as the root for the left figure. Suppose the plane of A intersects with polygon B and C. Then these two polygons are splitted into two parts, which are added to the front/behind list of A for recursive processing.
Here are some reasons:

1. Different type of monitor: RGB<sub>2019</sub> ≠ RGB<sub>2020</sub>
2. The system Gamma and Gamut are different
3. The brightness and contrast settings are different
4. The environment, such as the ambient light, is different.

\[ P_1 (-4, -5, 7), P_2 (0, -6, -10), P_3 (0, -4, -12) \]
\[ b = -1, t = 1, c = -1, r = 1, n = 3 \]

**Plane equation:**

- **Left:** \( x + t * z / n = x - 2 / 3 = 0 \)
- **Right:** \( -(x + z / 3) = 0 \)
- **Top:** \( -(y + t * z / n) = -(y + z / 3) = 0 \)
- **Bottom:** \( y - z / 3 = 0 \)
- **Near:** \( -(x + 3) = 0 \)
- **Far:** \( z + 9 = 0 \)

**Clipping:**

**Top**

- \( P_1: (-5 + 7/3) > 0 \) **Inside**
- \( P_2: (-6 -10/3) > 0 \) **Inside**
- \( P_3: (-4 -12/3) > 0 \) **Inside**

**Output:** \( P_1, P_2, P_3 \)

**Bottom**

- \( P_1: -5 - 7/3 < 0 \) **Outside**
- \( P_2: -6 + 10/3 < 0 \) **Outside**
- \( P_3: -4 + 12/3 = 0 \) **Outside**

**Output:** None

After clipping of bottom plane, nothing left in the frustum.
Assume the circle is $x^2 + y^2 = R^2$

For the function $d = F(x, y) = x^2 + y^2 - R^2$

we have:

- $d = 0$ : $(x, y)$ is on the circle
- $d > 0$ : $(x, y)$ is outside
- $d < 0$ : $(x, y)$ is inside

Since the symmetry of a circle, we only need to compute 1/8 of the whole circle (Figure 2), starting from $(0, R)$ to the point where $x > y$

Similar to Bresenham algorithm, we only need to consider the E and SE directions for next step, depending on which pixel is closer to the circle. That is, whether the midpoint of E and SE is inside or outside the circle (Figure 3)

The computation of $d_{old}, d_{new}$ and $d_{initial}$ is:

\[
\begin{align*}
    d_{old} &= F(x_p + 1, y_p - \frac{1}{2}) = (x_p + 1)^2 + (y_p - 1/2)^2 - R^2 \\
    d_{new} &= \begin{cases} 
    \text{next = E: } F(x_p + 2, y_p - \frac{1}{2}) & d_{old} + 2x_p + 3 \\
    \text{next = SE: } F(x_p + 2, y_p - \frac{3}{2}) & d_{old} + 2x_p - 2y_p + 5 
    \end{cases} \\
    d_{initial} &= F(x_0 + 1, y_0 - \frac{1}{2}) = \frac{1}{4}y_0^2 - R
\end{align*}
\]

What we need is integer only algorithm. But \( d_{\text{initial}} \) is not an integer! There are two methods to solve this. One is multiply \( 4 \) for each expression of \( d \), the other is just using \( 1-R \) for \( d_{\text{initial}} \) since the sign of
\[
\begin{align*}
\text{if } d &\leq 0 \Rightarrow d-\frac{1}{4} \leq 0 \\
\text{else } d &> 0 \Rightarrow d-\frac{1}{4} > 0
\end{align*}
\]
The pseudo code for drawing a circle is
\[
\begin{align*}
x &= 0; \quad y = R; \quad d = d_{\text{initial}} \\
\text{draw } &\text{way point } (x, y) \\
\text{while } &\text{ } (x < y) \\
\text{if } &\text{ } (d < 0) \\
\quad &E \text{ direction} \\
\text{else } &\text{ } \\
\quad &SE \text{ direction} \\
\text{draw } &\text{way point } (x, y)
\end{align*}
\]
Q11. Faces that would be removed by backface culling are:

D, F, G & I

Q12 Graphics Pipeline

Q13 Gimbal Lock:

It is a problem that can occur while specifying rotations using Euler angles. If due to a sequence of rotations about the three axes, two of the axes get aligned then we arrive at a configuration called gimbal lock. One degree of freedom is lost as rotation about one of these axes results in rotation about the other axis.

Two methods that avoid gimbal locks are:

Axis-angle rotation
Quaternions rotation
Situation A: 10,000 Polygons & 640 x 480 Pixel Frame Buffer
Situation B: 1,000 Polygons & 4800 x 2400 Pixel Frame Buffer

Hidden surface removal can be done both at the object level (polygons) and image level (pixels). In class, we saw a few object level algorithms for resolving visibility such as the binary space partition (BSP) trees and the Warnock's algorithm as well as the Z-buffer (image level) algorithm.

In Situation A, there are many more polygons and comparatively few pixels. Most polygons could be occluded as due to a limited number of pixels, many polygons could project to the same pixels. In such a situation, it would make sense to resolve visibility at the pixel level by using the Z-buffer algorithm. Most polygons would fail the depth test eliminating the need to modify the frame buffer. Also, resolving visibility at the object level is typically scene dependent and requires managing data structures such as BSP trees. If the scene changes, recomputation of the tree is expensive for a high polygon count.

In Situation B, the trade-off is reversed. A few polygons should be easy to manage at the object level. Also, with many pixels to project to, there will potentially be many updates to the depth & frame buffer. Also, there is this huge memory requirement for such large depth & frame buffers.
Q15.

The figure depicts the response of the checkerboard signal with white being high and black being low. The period of this signal is 10 pixels. According to the Nyquist rate, we need to sample this signal at twice the period of its highest frequency (here there is only one frequency). Thus we need to sample every 5 pixels to be at the Nyquist rate. Note that by sampling at every 5 pixels you do not reconstruct the entire signal, just that white and black pixels alternate in your signal. Therefore sampling at Nyquist rate tells you the essential characteristics of the signal.