Q1:A)

General Bresenham's algorithm for all quadrants

\[ X = X_1 \]
\[ Y = Y_1 \]
\[ dX = \text{Abs} (X_2 - X_1) \]
\[ dY = \text{Abs} (Y_2 - Y_1) \]
\[ S1 = \text{Sign} (X_2 - X_1) \]
\[ S2 = \text{Sign} (Y_2 - Y_1) \]
If \( dY > dX \) Then

Begin

\[ T = dX \colon dX = dY \colon dY = T \colon \text{Interchange} = 1 \]

End

Else

\[ \text{Interchange} = 0 \]

End If

\[ E = 2 \ dy - dx \]

For \( I = 1 \) to \( dX \)
Plot \((X, Y)\)
While \( E \geq 0 \)

Begin

If \( \text{Interchange} = 1 \) Then \( X = X + S1 \)
Else \( Y = Y + S2 \)

End If

\[ E = E - 2 \ dx \]

End While

If \( \text{Interchange} = 1 \) Then \( Y = Y + S2 \)
Else \( X = X + S1 \)

End If
\[ E = E + 2 dy \]

Next I

Finish

Sol 4: X=0; Y=0; dX=8; dY=4; S1=-1; S2=-1

Because \( dX > dY \) then Interchange=0; E=0

<table>
<thead>
<tr>
<th></th>
<th>Plot</th>
<th>E</th>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(0,0)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-16</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-8</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>2</td>
<td>(-1,-1)</td>
<td>0</td>
<td>-2</td>
<td>-1</td>
</tr>
<tr>
<td>3</td>
<td>(-2,-1)</td>
<td>-16</td>
<td>-2</td>
<td>-2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-8</td>
<td>-3</td>
<td>-2</td>
</tr>
<tr>
<td>4</td>
<td>(-3,-2)</td>
<td>0</td>
<td>-4</td>
<td>-2</td>
</tr>
<tr>
<td>5</td>
<td>(-4,-2)</td>
<td>-16</td>
<td>-4</td>
<td>-3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-8</td>
<td>-5</td>
<td>-3</td>
</tr>
<tr>
<td>6</td>
<td>(-5,-3)</td>
<td>0</td>
<td>-6</td>
<td>-3</td>
</tr>
<tr>
<td>7</td>
<td>(-6,-3)</td>
<td>-16</td>
<td>-6</td>
<td>-4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8</td>
<td>-7</td>
<td>-4</td>
</tr>
<tr>
<td>8</td>
<td>(-7,-4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----</td>
<td>---------</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>-8</td>
<td>-4</td>
<td></td>
</tr>
</tbody>
</table>

**Q1:** B) Draw in any quadric and any direction

**Q2:** A)

The Calligraphic Refresh graphics display

A Calligraphic (line drawing or vector) refresh CRT display uses a very short persistence phosphor. These displays are frequently called random scan display. Because of the short persistence of the phosphor, the picture painted on the CRT must be repainted or refreshed many times each second. The minimum refresh rate is at least 30 times each second. Refresh rates much lower than 30 times each second result in a flickering image.

The basic calligraphic refresh display requires two elements in addition to the CRT. These are the display buffer and the display controller. The display buffer is contiguous memory containing all the information required to draw the picture on the CRT. The display controller’s function is to repeatedly cycle through this information at the refresh rate. Two factors which limit the complexity (number of vectors displayed) of the picture are the size of the display buffer and the speed of the display controller. A further limitation is the speed at which picture information can be processed.

**Q2:** B) Features of calligraphic refresh displays

1- It is a vector graphics display
2- Resolution is the same as storage tube display
3- Employee the concept of picture segmentation that support the interactive graphics programs

**Q3:** 1) The result is (0,0),(2,0),(1,2)

2: Rotation by 180

\[
\begin{align*}
0 & \quad 0 & 1 \\
2 & \quad 0 & 1 \\
1 & \quad 2 & 1
\end{align*}
\]

\[
\begin{align*}
-1 & \quad 0 & 0 \\
0 & \quad -1 & 0 \\
0 & \quad 0 & 1
\end{align*}
\]

\[
\begin{align*}
0 & \quad 0 & 1 \\
-2 & \quad 0 & 1 \\
-1 & \quad -2 & 1
\end{align*}
\]
3: The result is (2,1),(0,1),(1,-1)

Q3:2)
\[
\begin{array}{ccc}
2 & 1 & 1 \\
0 & 1 & 1 \\
1 & -1 & 1 \\
\end{array}
\times
\begin{array}{c}
1 \\
0 \\
0 \\
\end{array} =
\begin{array}{c}
2 \\
0 \\
1 \\
\end{array}
\]

Q3:3)
\[
\begin{array}{ccc}
2 & -1 & 1 \\
0 & -1 & 1 \\
1 & 1 & 1 \\
\end{array}
\times
\begin{array}{c}
2 \\
0 \\
0 \\
\end{array} =
\begin{array}{c}
4 \\
0 \\
2 \\
\end{array}
\]

Q4:1)
\[
\begin{array}{ccc}
2 & 2 & 2 & 1 \\
4 & 2 & 2 & 1 \\
3 & 2 & 4 & 1 \\
3 & 4 & 3 & 1 \\
\end{array}
\times
\begin{array}{c}
-1 \\
0 \\
0 \\
0 \\
\end{array} =
\begin{array}{c}
-2 \\
-2 \\
-3 \\
-3 \\
\end{array}
\]

Q4:2)
\[
\begin{array}{ccc}
-2 & -2 & 2 & 1 \\
-4 & -2 & 2 & 1 \\
-3 & -2 & 4 & 1 \\
-3 & -4 & 3 & 1 \\
\end{array}
\times
\begin{array}{c}
1 \\
0 \\
0 \\
0 \\
\end{array} =
\begin{array}{c}
-2 \\
-4 \\
-3 \\
-3 \\
\end{array}
\]

Q5)

The Cohen Algorithm

1. Determine the endpoint \((X1, Y1)\) code1

If \(X1 < X_{min}\) Then

If \(Y1 > Y_{max}\) Then code1 = 1001
If \((Y1 < Y_{max}) \text{ And } (Y1 > Y_{min})\) Then code1 = 0001
If \(Y1 < Y_{min}\) Then code1 = 0101

End IF

If \(X1 > X_{max}\) Then

If \(Y1 > Y_{max}\) Then code1 = 1010
If \((Y1 < Y_{max}) \text{ And } (Y1 > Y_{min})\) Then code1 = 0010
If \(Y1 < Y_{min}\) Then code1 = 0110

End IF
If \((X_1 \leq X_{\text{max}}) \And (X_1 \geq X_{\text{min}})\) Then

- If \(Y_1 > Y_{\text{max}}\) Then \(\text{code}_1 = 1000\)
- If \((Y_1 \leq Y_{\text{max}}) \And (Y_1 \geq Y_{\text{min}})\) Then \(\text{code}_1 = 0000\)
- If \(Y_1 < Y_{\text{min}}\) Then \(\text{code}_1 = 0100\)

End IF

2- Determine the endpoint \((X_2, Y_2)\) \(\text{code}_2\) [as \(\text{code}_1\)]

3- Determine the visibility of the line

- If \(\text{code}_1 = \text{code}_2 = 0000\) then the line is visible; draw the line
- If \(\text{code}_1 \And \text{code}_2 \neq 0000\) then the line is not visible
- If \(\text{code}_1 \And \text{code}_2 = 0000\) then the line is candidate for clipping

The line \(KL\) is visible and the line \(AB\) is not visible.