Final (2012):

Q1: Any point (x,y) has one of three cases depending
a) on the function $f(x,y)$ provided that x is positive:

$$f(x,y) = \begin{cases} 0 & (x,y) \text{ on the line} \\ <0 & (x,y) \text{ under the line} \\ >0 & (x,y) \text{ above the line} \end{cases}$$

Initial value of d: at $(x=0, y=B)$

$$d_{initial} = d(x,y) = f(x+1, y-\tfrac{1}{2}) = (x+1)^2 + (y-\tfrac{1}{2})^2 - B^2$$

$$= (0+1)^2 + (B-\tfrac{1}{2})^2 - B^2$$

$$= 1 + \cancel{B^2} - B + \tfrac{1}{4} - \cancel{B^2}$$

$$= \tfrac{5}{4} - B \simeq 1 - B$$

Chang in d:

① if $d<0$ then: x++

$$\Delta d = d(x+2, y-\tfrac{1}{2}) - d(x,y)$$

$$= [(x+2)^2 + \cancel{(y-\tfrac{1}{2})^2} - \cancel{B^2}] - [(x+1)^2 + \cancel{(y-\tfrac{1}{2})^2} - \cancel{B^2}]$$

$$= (x^2 + 4x + 4) - (x^2 + 2x + 1) = 2x + 3$$

② if $d>0$ then: x++ y--

$$\Delta d = d(x+2, y-\tfrac{3}{2}) - d(x,y)$$

$$= [(x+2)^2 + (y-\tfrac{3}{2})^2 - \cancel{B^2}] - [(x+1)^2 + (y-\tfrac{1}{2})^2 - \cancel{B^2}]$$

$$= [(\cancel{x^2} + 4x + 4) + (\cancel{y^2} - 3y + \tfrac{9}{4})] + [(\cancel{x^2} + 2x + 1) + (\cancel{y^2} - y + 1)]$$

$$= (2x+3) + (-2y + \tfrac{5}{4})$$

$$\simeq 2(x-y) + 5$$

Scanned with CamScanner

Algorithm:

DrawCircle(x_c, y_c, R):

1. $x = 0$
2. $y = R$
3. $d = 1 - R$
4. Drawpoints(x_c, y_c, x, y)
5. if $(d < 0)$ → while$(x < y)$:

```
        d += 2x + 3;
        x++;
    else
        d += 2(x - y) + 5;
        x++;  y++;
```

6. Drawpoint(x_c, y_c, x, y)

$\boxed{b}$ slope $= \frac{15-9}{12-2} = 0.6$ $\therefore$ slop < 1

Initial value (dinitial) $= \Delta x - 2\Delta y = 10 - (2*6) = -2$

Change in decision when $d<0$: $d_1 = 2\Delta x - 2\Delta y = 20 - 12 = 8$

Change in decision when $d>0$: $d_2 = -2\Delta y = -2*6 = -12$

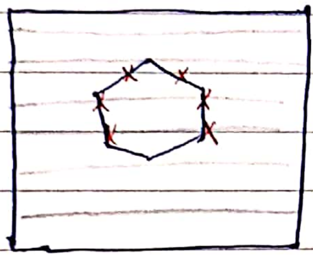
Traces

X	Y	Decision Variable
2	9	$d<0$ | $d = d + d_1 = -2 + 8 = 6$
3	10	$d>0$ | $d = d + d_2 = 6 + -12 = -6$
4	10	$d<0$ | $d = d + d_1 = -6 + 8 = 2$
5	11	$d>0$ | $d = d + d_2 = 2 - 12 = -10$
6	11	$d<0$ | $d = d + d_1 = -10 + 8 = -2$
7	12	$d<0$ | $d = d + d_1 = -2 + 8 = 6$
8	13	$d>0$ | $d = d + d_2 = 6 + 12 = -6$
9	13	

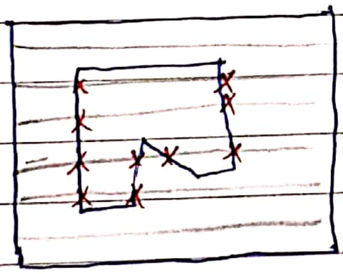
Scanned with CamScanner

Q2:

a Convex shapes: is a shape that has at most 2 intersection with Scan line.



Non-Convex shape: is a shape that has more than 2 intersection with Scan line.



Algorithm:

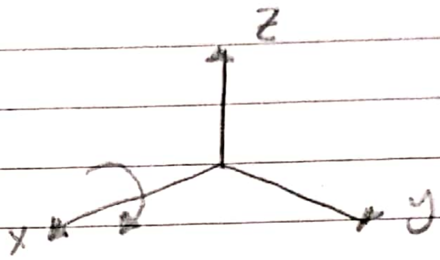
1. Initialize the edge list array 'table' with the edge information.
2. Start with y = the first index in table with non-empty edge list.
3. Let Active List = table[y]
4. while Active list is not empty
    - 4.1 Sort Active List node in an ascending order with x.
    - 4.2 Draw horizontal lines between points represented by successive pairs of node in Active List.
    - 4.3 Increment y by 1
    - 4.4 Delete from ActiveList those node with $y_{max} = y$.
    - 4.6 Append to ActiveList the new node at table[y] if not Null
    - 4.5 update x where x += minv

end while

Q4:

* Rotate an object about x axis with a rotation angle of 30° clockwise.

∴ Rotation → from z to y



$$R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(30) & \sin(30) & 0 \\ 0 & -\sin(30) & \cos(30) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

* Scale 3D object with respect to a fixed point (3, 2, 8) where the scaling factors in x, y and z directions are (2, 0.5, 3).

1. Translate object to origin:

$$T = \begin{bmatrix} 1 & 0 & 0 & -3 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & -8 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2. Scale object on origin:

$$S = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3. Undo step 1:

Scanned with CamScanner

3. Undo Step 1:

$$T^{-1} = \begin{bmatrix} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 8 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

∴ Composite Scaling Matrix:

$$T^{-1} \cdot S T = \begin{bmatrix} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 8 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -3 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & -8 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

* Transform Figure 1 into Figure 2:

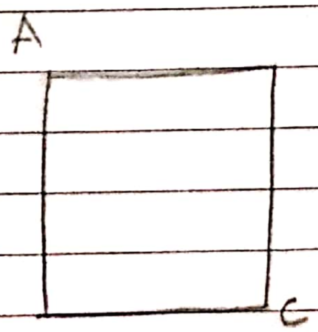


figure 1

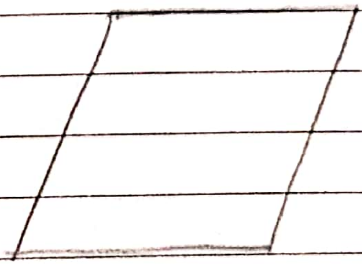


figure 2

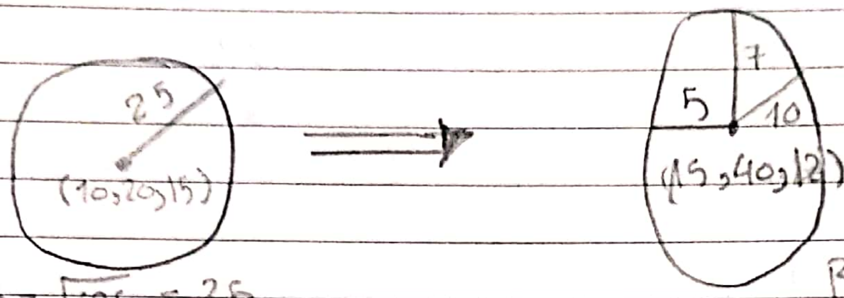
$$\therefore T = \begin{bmatrix} 1 & \alpha & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

$x' = x + \alpha y$
$y' = y + \beta x$

$x' = x + \alpha y$
$y' = y$

Map the sphere $(x-10)^2+(y-20)^2+(z-15)^2 = 625$ to the ellipsoi

$$\frac{(x-15)^2}{25}+\frac{(y-40)^2}{49}+\frac{(z-12)^2}{100} = 1$$



$R = \sqrt{625} = 25$ $\qquad$ $R = \sqrt{25}\ \ \sqrt{49}\ \ \sqrt{100}$

- if we want to map shape into another there is 2 difference:
① center of ellipsoid
② R in sphere is 25 but ellipsoid has 3 diffrent values (5, 7, 10)

① Translate sphere to origin:

$$T_1 = \begin{vmatrix} 1 & 0 & 0 & -10 \\ 0 & 1 & 0 & -20 \\ 0 & 0 & 1 & -15 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

② we need to convert R which (25) to (A, B, C) whice are (5, 7, 10) so we need to Apply two Scaling one to convert (25) to 1 and other to Rescale 1 to (5, 7, 10):

$$S_1 = \begin{vmatrix} 1/25 & 0 & 0 & 0 \\ 0 & 1/25 & 0 & 0 \\ 0 & 0 & 1/25 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Scanned with CamScanner

$$S_2 = \begin{vmatrix} 5 & 0 & 0 & 0 \\ 0 & 7 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

③ Translate from origin to ellipsoid center:

$$T_2 = \begin{vmatrix} 1 & 0 & 0 & 15 \\ 0 & 1 & 0 & 40 \\ 0 & 0 & 1 & 12 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Final matrix = $T_2 \, S_2 \, S_1 \, T_1 \times$ input

$$= \begin{vmatrix} 1 & 0 & 0 & 15 \\ 0 & 1 & 0 & 40 \\ 0 & 0 & 1 & 12 \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} 5 & 0 & 0 & 0 \\ 0 & 7 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} \frac{1}{25} & 0 & 0 & 0 \\ 0 & \frac{1}{25} & 0 & 0 \\ 0 & 0 & \frac{1}{25} & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} 1 & 0 & 0 & -10 \\ 0 & 1 & 0 & -20 \\ 0 & 0 & 1 & -15 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Final (2017):

Q:

$\boxed{b}$ (12, 10) (6, 14)

1. $\Delta y = 14 - 10 = 4$ $\Delta x = 6 - 12 = -6$

$|5| \leq |-6|$

slope < 1

↓

negative

$d = -\Delta x - 2\Delta y = -(-6) - 2(4) = -2$

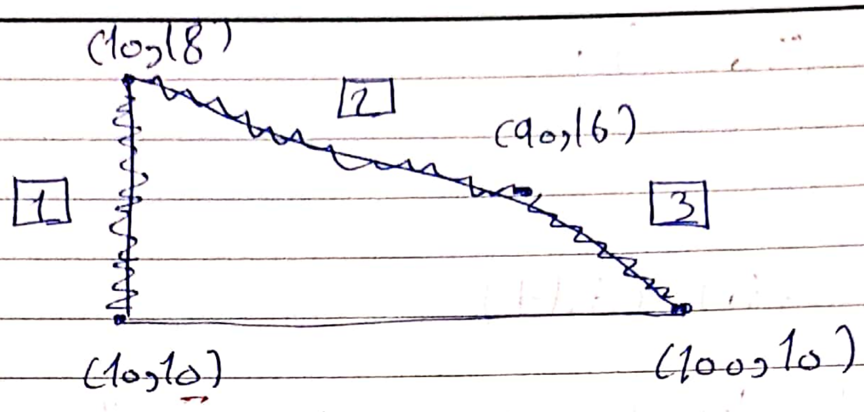
$ch_1 = -\Delta x - \Delta y = 6 - 4 = 2$

$ch_2 = -2\Delta y = -2(4) = -8$

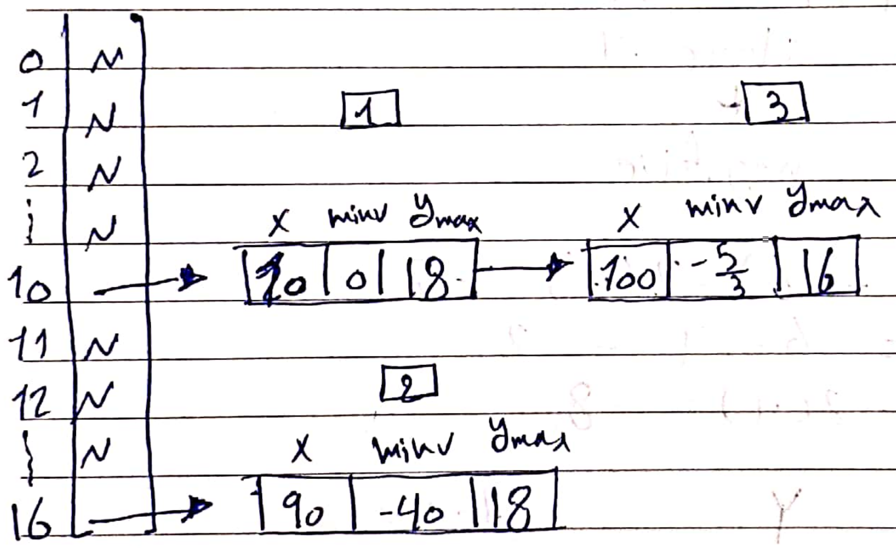
X	Y	decision		
6	14	-2	$\because d \leq 0$	$\because d += ch_1$
7	13	0	$\because d \leq 0$	$\because d += ch_1$
8	12	2	$\because d > 0$	$\because d += ch_2$
9	12	-6	$\because d \leq 0$	$\because d += ch_1$
10	11	-4	$\because d \leq 0$	$\because d += ch_1$
11	10	-2	$\because d \leq 0$	$d += ch_1$
12	9	0		

Scanned with CamScanner

C



ActiveList



start with y=0.

1 while (tbl[y] == Null) y++;

2 Active = tbl[y] ∴ Active = tbl[10]

3 while (Active != Null)

∴ y = 10

① sort x ascending ∴

x	min	ymax		x	min	ymax
10	0	18	→	100	$-\frac{5}{3}$	16

② Draw Line between (10,10) (100,10)

③ Increase y by one. ∴ y = 11

④ delet will not occure cause $y \neq y_{max}$

⑤ Update x of each nod in ActiveList

x	minv	ymax		x	minv	ymax
10	0	18	→	$\frac{295}{3}$	$\frac{-5}{3}$	16

⑥ Append if tbl[y] not Null
$\therefore$ Append will not occure cause tbl[11] Null

$\therefore y = 11$

Active → | 10 | 0 | 18 | → | $\frac{295}{3}$ | $-\frac{5}{3}$ | 16 |

① Sort x ascending
② Draw Line between (10, 11) ($\frac{295}{3}$, 11)
③ Increas y by one $\therefore y = 12$
④ delet will not occure cause $y \neq y_{max}$
⑤ update x of each node in Active List

| 10 | 0 | 18 | → | $\frac{290}{3}$ | $-\frac{5}{3}$ | 16 |

⑥ Append if tbl[y] not Null
$\therefore$ Append will not occure cause tbl[12] Null

Q2

$\boxed{a}$: The line will accepted if
$\boxed{1}$ both outcode not empty
$\boxed{2}$ both outcode not have common side

Condition of $\boxed{1}$:
if(outcode1.All != 0 && outcode2.All != 0)

Condition of $\boxed{2}$:
if(outcode1.All & outcode2.All) != 0

Q2:

[b] $P(t_1, t_2) = t_1 P_1 + t_2 P_2 + (1 - t_1 - t_2) P_3$

$0 \leq t_1, t_2 \leq 1$

$t_1 + t_2 \leq 1$

★ Every time (t_1, t_2) change we can calculate new point inside the traingle.

[1] for ① iterate on t_1 values
② iterate on t_2 values.
③ check if $t_1 + t_2 \leq 1$
④ generate p values.
⑤ set pixel.

```
for (t1=0 , t1≤1 , t+=1/max(P1,P2,P3))
    for(t2=0, t2 ≤ 1 ; t2+=1/max(P1,P2,P3))
        if (t1+t2 > 1)
            continue;
        else
        p(t1,t2) = t1 P1 + t2 P2 + (1-t1-t2) P3
        setpixel (P).
```

[2] for ① iterate on t_1 values
② iterate on t_2 values
③ check if $(t_1 + t_2) \leq 1$
④ check (x,y) values of P with (P_1, P_2, P_3) compare

```
for (t1=0, t1≤1 , t1+=1/max(P1,P2,P3))
    For(t2=0, t2≤1, t2=1/max(P1,P2,P3))
        if (t1+t2>1)
            continues;
    else
    P(t1,t2) = t1P1 + t2P2 + (1-t1-t2)·P3
    if (P.x ≤ P1.x && P.x ≤ P2.x && P.x ≤ P3.x
        && P.y ≤ P1.y && P.y ≤ P2.y && P.y ≤ P3.y)
    then Pixel is inside
```

Q3:

[a] The homogeneous coordinate systems;
- Is related to the standard 2D system through the relation H and S with H mapping from standard to homogenous and S from homogeneous to standard:

$$H: (x, y) \rightarrow (x, y, 1)$$

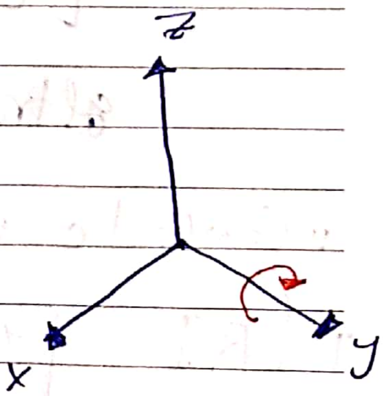
$$S: (x, y, w) \rightarrow \left(\frac{x}{w}, \frac{y}{w}\right)$$

- Importance of homogenous coordinate System:
It used for representation of affine transformation and helpfull with operation of * and + of matrices so it can simplify the operation

- Affine transforma in homogenous coordinate System:

$$T\left(\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}\right) = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

[b] : rotate from x to Z



$$R_y = \begin{bmatrix} \cos(60) & 0 & -\sin(60) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(60) & 0 & \cos(60) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

glRotatef(-θ, 0, 1, 0)

Scanned with CamScanner

c 1. Translate the object such that (10,12) become the origin using the translation matrix:

$$T = \begin{bmatrix} 1 & 0 & -10 \\ 0 & 1 & -12 \\ 0 & 0 & 1 \end{bmatrix}$$

~~glRotatef(30,0,0,1)~~
glTranslatef(-10,-12,1)

2. Rotate the object about the origin using the rotation matrix:

$$R = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$ glRotatef(30,0,0,1)

3. Undo step 1: Translate the origin to (10,12)

$$T^{-1} = \begin{bmatrix} 1 & 0 & 10 \\ 0 & 1 & 12 \\ 0 & 0 & 1 \end{bmatrix}$$

gltranslate(10,12,1)

∴ Composite Matrix:

$$T^{-1}.R.T = \begin{bmatrix} 1 & 0 & 10 \\ 0 & 1 & 12 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -10 \\ 0 & 1 & -12 \\ 0 & 0 & 1 \end{bmatrix}$$

Scanned with CamScanner

$\boxed{d}$ shearing in x-direaction:

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} 1 & \alpha & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$x = 5$
$y = 12$
$\alpha = 4$
$\beta = 4$

shearing in y-direction:

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ \beta & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

2018

b

* Recursive Algorithm:

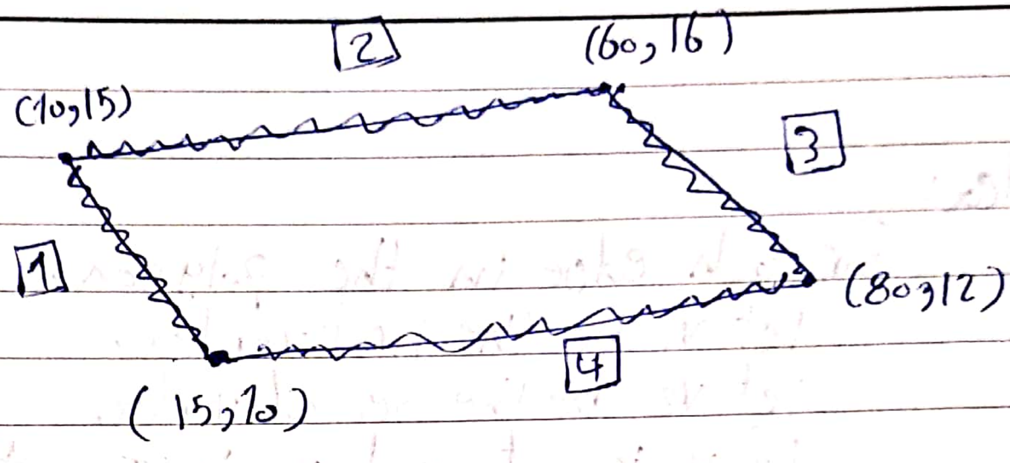
```
FloodFill (x, y, Bc, Fc):
  1. COLORREF c = Getpixel (x, y);
  2. if ( c == Bc || c == Fc) return;
  3. setpixel ( x, y, Fc);
  4. Flood fill ( x+1, y, Bc, Fc);
  5. Floodfill ( x, y+1, Bc, Fc);
  6. flood fill ( x-1, y, Bc, Fc);
  7. flood fill ( x, y-1, Bc, fc);
end
```

* Non-Recursive Algorithm:

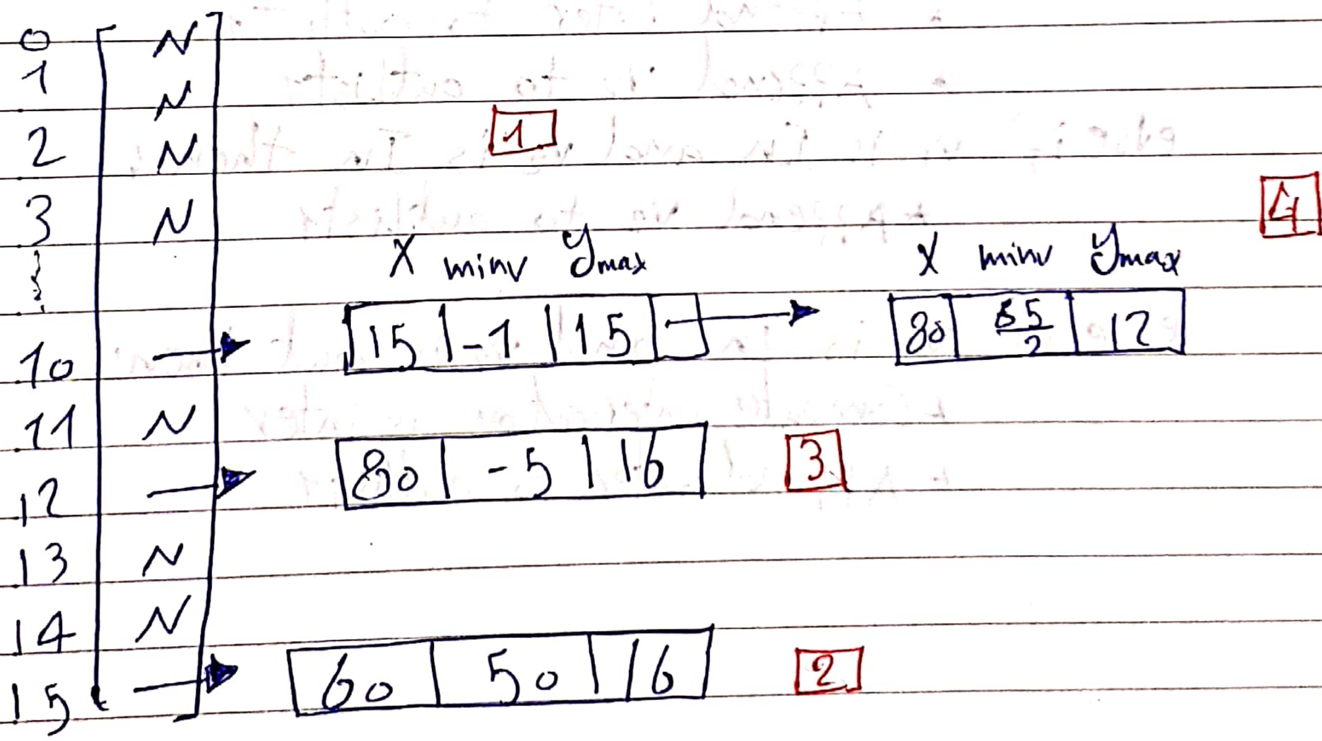
```
#include <queue>
FloodFill ( x, y, Bc, Fc):
  1. struct Point:
       1.1 x, y;
       1.2 point (x=0, y=0) : x(x) y(y) {};
  2. queue <point> qu;
  3. qu.push (point(x,y));
  4. while ( ! qu.empty()):
       4.1 point p = qu.top();
       4.2 qu.pop();
       4.3 COLORREF c = Getpixel (p.x, p.y);
       4.4 if(c == Bc || c == Fc) continue;
       4.5 setpixel (p.x, p.y, Fc);
       4.6 qu.push (point (x+1, y));
       4.7 qu.push ( point (x, y+1));
       4.8 qu.push (point (x-1, y));
       4.9 qu.push (point (x, y-1));
  5 end while
end.
```

Scanned with CamScanner

C



Active List



Q_2:

a) Idea:

for each edge in the polygon:
    let v_1 be the first vertex.
    let v_2 be the second vertex.
    if v_1 is out and v_2 is In then:
        * Comput intersection as inter
        * Append inter to outlist1
        * Append v_2 to outList1
else if v_1 is In and v_2 is In then:
        * Append v_2 to outList1

else if v_1 is In and v_2 is out then:
        * Compute intersection as inter
        * Append inter to outlist1

Scanned with CamScanner

b Draw Circle (x_c, y_c , R):

1. x = 0 ;
2. y = R ;
3. d = 1 - R ;
4. d_1 = 3 ;
5. d_2 = 5 - 2 * R ;
6. Drawpoint (x_c, y_c, x, y, Color) ;
7. while (x < y):
    - 7.1 if (d < 0) ;
        - 7.1.1 d += d_1 ;
        - 7.1.2 d_2 += 2 ;
        - 7.1.3 d_1 += 2 ;
    - 7.2 else :
        - 7.2.1 d += d_2 ;
        - 7.2.2 d_2 += 4 ;
        - 7.2.3 d_1 += 2 ;
        - 7.2.4 y-- ;
    - 7.3 Drawpoint (x_c, y_c, x, y, Color) ;

  end while

end

Draw Point (x_c, y_c , x, y):

1. setpixel (x_c+x, y_c+y , color) ;
2. DrawLine (x_c, y_c , x_c+x, y_c+y, Color) ;
3. Set pixel (x_c+y , y_c+x , Color) ;
4. DrawLine (x_c, y_c , x_c+y , y_c+x , color) ;
5. setpixel (x_c-x , y_c+y , color) ;
6. DrawLine (x_c, y_c , x_c-x , y_c+y , Color) ;
7. Setpixel (x_c-x , y_c-y , Color) ;
8. DrawLine (x_c, y_c, x_c-x , y_c-y , Color) ;
9. Setpixel (x_c+x ; y_c - y ; Color) ;

Scanned with CamScanner

10. DrawLine (x_c, y_c, x_c+y, y_c+x, color);
11. Setpixel (x_c-y, y_c+x, color);
12. DrawLine (x_c, y_c, x_c-y, y_c+x, color);
13. Setpixel (x_c+y, y_c-x, color);
14. DrawLine (x_c, y_c, x_c+y, y_c-x, color);
15. Setpixel (x_c-y, y_c-x, color);
16. DrawLine (x_c, y_c, x_c-y, y_c-x, color);
~~17. Setpixel~~
end.

Q3:

a) General form of Affine transformation in homogenous space:

$$T\left(\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}\right) = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

b) Translate (10, 12) to origin:

$$T = \begin{bmatrix} 1 & 0 & -10 \\ 0 & 1 & -12 \\ 0 & 0 & 1 \end{bmatrix}$$

glTranslatef(-10, -12, 1);

* Rotate about origin with angle 30° anti-clock:

$$R = \begin{bmatrix} \cos 30 & -\sin 30 & 0 \\ \sin 30 & \cos 30 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3}}{2} & -\frac{1}{2} & 0 \\ \frac{1}{2} & \frac{\sqrt{3}}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

glRotatef(30, 0, 0, 1);

Scanned with CamScanner

undo step 1:

$$T^{-1} = \begin{bmatrix} 1 & 0 & 10 \\ 0 & 1 & 12 \\ 0 & 0 & 1 \end{bmatrix}$$

gltranslate(10, 12, 1);

$\therefore$ overall Matrix $=$

$$T^{-1} \cdot R T = \begin{bmatrix} 1 & 0 & 10 \\ 0 & 1 & 12 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{\sqrt{3}}{2} & -\frac{1}{2} & 0 \\ \frac{1}{2} & \frac{\sqrt{3}}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -10 \\ 0 & 1 & -12 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\sqrt{3}}{2} & -\frac{1}{2} & 10 \\ \frac{1}{2} & \frac{\sqrt{3}}{2} & 12 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -10 \\ 0 & 1 & -12 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\sqrt{3}}{2} & -\frac{1}{2} & 16-5\sqrt{3} \\ \frac{1}{2} & \frac{\sqrt{3}}{2} & 7-6\sqrt{3} \\ 0 & 0 & 1 \end{bmatrix}$$

Scanned with CamScanner

d. Shearing in x-direction:

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} 1 & \alpha & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Shearing in y-direction:

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ \beta & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

where $x = 5$ $y = 12$ $\alpha = \beta = 4$.

Scanned with CamScanner

Final (2019):

Q1

a) Bresenham_Algorithm(x_1, y_1, x_2, y_2):

1. $x = x_1$
2. $y = y_2$
3. $dx = x_2 - x_1$
4. $dy = y_2 - y_1$
5. setPixel (x, y)
6. $d = dx - 2dy$
7. $d_1 = -2 * dy$
8. $d_2 = 2 * dx - 2 * dy$
9. while($x < x_2$):

    9.1 if($d > 0$):

        x++ y++

        d+=d2

    9.2 else:

        x++

        ~~y++~~

        d+=d1

    9.3 setpixel (x, y);

  endwhile

end.

Scanned with CamScanner

DDA Algorithm	Bresenham Algorithm
* less efficient	* More efficient
* calculation speed less.	* Calculation speed faster.
* Costlier	* cheaper
* less accuracy	* more accuracy
* more complex	* simple complex
* use multiplication and division	* use only substraction and addition.

$\boxed{b}$ $(6,10) \rightarrow (12,18)$ slope $= 1.33$ >1

$x_1 = 6$ $x_2 = 12$ $y_1 = 10$ $y_2 = 18$

$\Delta x = 6$

$\Delta y = 8$

$d_{initial} = 2\Delta x - \Delta y = 12 - 8 = 4$

$d_1 = 2\Delta x - 2\Delta y = 12 - 16 = -4$

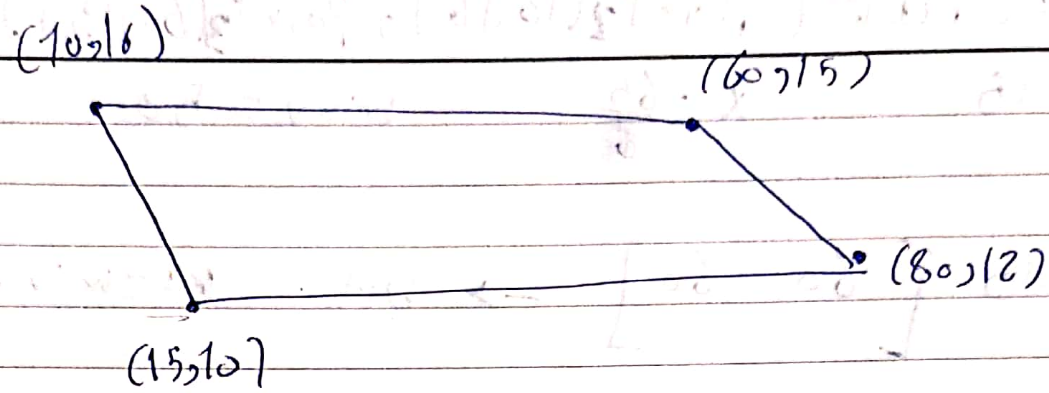
$d_2 = 2\Delta x = 12$

Traces

X	Y	deision		
6	10	$d \geq 0$	$d += d_1$	$d = 0$
7	11	$d \geq 0$	$d += d_1$	$d = 4$
8	12	$d < 0$	$d += d_2$	$d = 8$
8	13	$d \geq 0$	$d += d_1$	$d = 4$
9	14	$d \geq 0$	$d += d_1$	$d = 0$
10	15	$d \geq 0$	$d += d_1$	$d = 4$
11	16	$d \leq 0$	$d += d_2$	$d = 8$
11	17	$d \geq 0$	$d += d_1$	$d = 4$
12	18			

Scanned with CamScanner

d



Edg2Table (point$_1$ (15,10) , point$_2$ (10,16)):

Initial Table :

$$\begin{array}{cc} X_{left} & X_{right} \\ \left[\begin{array}{c} \infty \\ \infty \\ \infty \end{array}\right. & \left.\begin{array}{c} -\infty \\ -\infty \\ -\infty \end{array}\right] \end{array}$$

$\because y_1 = 10 \qquad x = 15 \qquad min_v = -\frac{5}{6}$

$$\begin{array}{ccc} & X_{left} & X_{right} \\ y = 10 & \left[15 \right. & \left. 15 \right] \end{array} \longrightarrow y{+}{+} \qquad x {+}{=} minv$$

$\because y = 11 \qquad x = \frac{85}{6}$

$$y = 11 \begin{bmatrix} \frac{85}{6} & \frac{85}{6} \end{bmatrix} \longrightarrow y{+}{+} \qquad x {+}{=} minv$$

$\because y = 12 \qquad x = \frac{40}{3}$

$$y = 12 \begin{bmatrix} \frac{40}{3} & \frac{40}{3} \end{bmatrix} \longrightarrow y{+}{+} \qquad x {+}{=} minv$$

$\because y = 13 \qquad x = \frac{25}{2}$

$$y = 13 \begin{bmatrix} \frac{25}{2} & \frac{25}{2} \end{bmatrix} \longrightarrow y{+}{+} \qquad x {+}{=} minv$$

$\because y = 14 \qquad x = \frac{35}{3}$

$$y = 14 \begin{bmatrix} \frac{35}{3} & \frac{35}{3} \end{bmatrix} \longrightarrow y{+}{+} \qquad x {+}{=} minv$$

Scanned with CamScanner

Edg2Table ($Point_2(10,16)$, $Point_3(60,15)$):

$y = 15$ $\quad x = \frac{65}{\text{[illegible]}}$ $\quad minv = 50.$

$y = 15 \begin{bmatrix} 60 & 60 \end{bmatrix} \rightarrow y{+}{+} \quad x {+}{=} minv$

Edg2 Table ($point_3(60,15)$, $Point_4(80,12)$):

$y = 12 \quad x = 80 \quad \cancel{m}inv = -\frac{20}{3}$

$y = 12 \begin{bmatrix} \frac{40}{3} & 80 \end{bmatrix}$

$y{+}{+} \quad x {+}{=} minv \rightarrow y = 13. \quad x = \frac{220}{3}$

$y = 13 \begin{bmatrix} \frac{25}{2} & \frac{220}{3} \end{bmatrix}$

$y{+}{+} \quad x {+}{=} minv \rightarrow y = 14 \quad x = \frac{200}{3}$

$y = 14 \begin{bmatrix} \frac{35}{3} & \frac{200}{3} \end{bmatrix}$

$y{+}{+} \quad x {+}{=} minV \rightarrow y = 15 \quad x = 60$

EdgeTable (point$_4$(80, 12), point$_1$(15, 10)):

$y = 10$ $\qquad$ $x = 15$ $\qquad$ $minv = \frac{65}{2}$

$y = 10 \quad \begin{bmatrix} 15, & 32.5 \end{bmatrix} \rightarrow y{+}{+} \quad x = \frac{95}{2}$

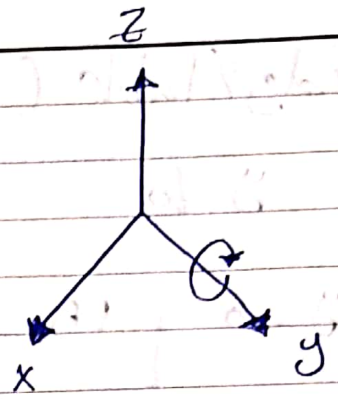
$y = 11 \quad \begin{bmatrix} \frac{85}{6} & \frac{95}{2} \end{bmatrix} \rightarrow y{+}{+} \quad x = 80$

$\therefore$ EdgTable =

$$\begin{array}{c} y \\ 10 \\ 11 \\ 12 \\ 13 \\ 14 \\ 15 \end{array} \begin{array}{c} \\ \begin{bmatrix} 15 & 32.5 \\ \frac{85}{6} & \frac{95}{2} \\ \frac{40}{3} & 80 \\ \frac{25}{2} & \frac{220}{3} \\ \frac{35}{3} & \frac{200}{3} \\ 60 & 60 \end{bmatrix} \end{array}$$

Q2:

c) from x to z



$$R_y = \begin{bmatrix} \cos(60) & 0 & -\sin(60) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(60) & 0 & \cos(60) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{2} & 0 & -\frac{\sqrt{3}}{2} & 0 \\ 0 & 1 & 0 & 0 \\ \frac{\sqrt{3}}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

d) 1. Translate object to origin:

$$T = \begin{bmatrix} 1 & 0 & 0 & -3 \\ 0 & 1 & 0 & -4 \\ 0 & 0 & 1 & -7 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2. Scale object on the origin:

$$S = \begin{bmatrix} 0.5 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Scanned with CamScanner

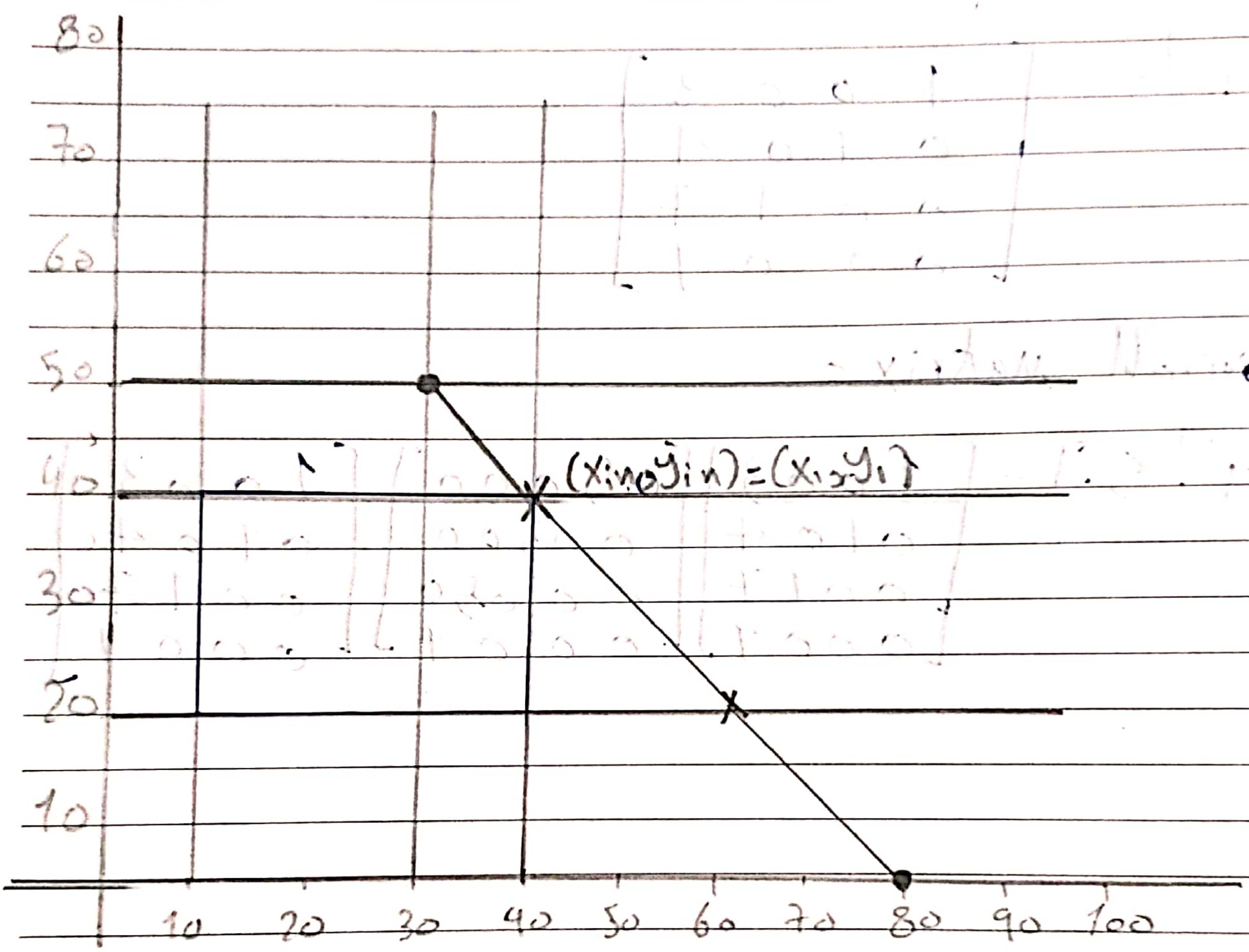
undo step 1:

$$T^{-1} = \begin{bmatrix} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 1 & 7 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

overall matrix =

$$T^{-1} \cdot S \cdot T = \begin{bmatrix} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 1 & 7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.5 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 1 & 7 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Scanned with CamScanner



$out_1 = \{0010\}$ $out_2 = \{0101\}$

$\boxed{1}$ $out_1.lef = 0$ $out_1.bottom = 0$ $out_1.top = 1 \rightarrow$ intersect with top side

$\because y_{top} = 40$ $\because y_{in} = y_{top} = 40$ where $(x_1, y_1) = (30, 50)$

$(x_2, y_2) = (80, 0)$

$\because x_{in} = x_1 + (x_2 - x_1) * (y_{in} - y_1) / (y_2 - y_1)$

$= 30 + (80 - 30) * (40 - 50) / (0 - 50) = 40$

$\therefore (x_{in}, y_{in}) = (x_1, y_1) = (40, 40)$

$\therefore out_1 = \{0000\}$ # trivial empty

$\boxed{2}$ $out_2 = \{0101\}$

$out_2.left = 0$ $out_2.bottom = 1 \Rightarrow$ intersect with y_{bottom}

$\because y_{bottom} = y_{in} = 20$ where $(x_1, y_1) = (40, 40)$

$(x_2, y_2) = (80, 0)$

$\because x_{in} = x_1 + (x_2 - x_1) * (y_{in} - y_1) / (y_2 - y_1)$

$= 40 + (80 - 40) * (20 - 40) / (0 - 40) = 60$

$\because (x_{in}, y_{in}) = (x_2, y_2) = (60, 20)$

$out_2 = \{0001\}$

$out_2.lef = 0$ $out_2.bottom = 0$ $out_2.top = 0$ $out_2.right = 1$

$\therefore$ intersect with x_{right}

$\because x_{right} = x_{in} = 40$ where $(x_1, y_1) = (40, 40)$

$(x_2, y_2) = (60, 20)$

$y_{in} = y_1 + (x_{in} - x_1) * (y_2 - y_1) / (x_2 - x_1)$

$= 40 + (40 - 40) * (20 - 40) / (60 - 40) = 40$

$\therefore (x_{in}, y_{in}) = (x_2, y_2) = (40, 40)$

$out_2 = \{0000\}$

$(out_1.All == 0 \; \&\& \; out_2.All == 0)$

Draw point $(40, 40)$.

(2017):

write an efficient Algorithm to fill an ellipse with axes parallel to the x and y axes.

```
DrawEllipse( Xc, Yc, A, B, color):
    x=0    y=B
    Draw4point(Xc, Yc, X, Y, Color)
    while(xB^2 < yA^2):
        x++
        y=Round(Sqrt(B^2*A^2 - x^2/A^2))
        Draw4point(Xc, Yc, X, y, Color)
    endwhile
    y=0    x=A
    Draw4point(Xc, Yc, X, Y, Color)
    while(xB^2 > yA^2):
        y++
        x=Round(sqrt(B^2*A^2 - y^2/B^2))
        Draw4point(Xc, Yc, X, Y, Color)
    end while
end
Draw4Point(Xc, Yc, X, Y, Color):
    setpixel(X+Xc, Y+Yc, Color)
    setpixel(X-Xc, Y-Yc, Color)
    drawLine(X+Xc, y+Yc, X-Xc, y-Yc, Color)
    setpixel( X+Xc, y-Yc, Color)
    setpixel(X-Xc, y+Yc, Color)
    drawLine(X+Xc, y-Yc, X-Xc, y+Yc, Color)
```

Scanned with CamScanner

write an efficent algorithm to fill an ellipse using the following parametric equations:

$$X = A\cos\theta$$
$$y = B\sin\theta$$

```
DrawEllipse( xc, yc, B, r², color):
    dtheta = 1.0/B
    For ( theta = 0.0 , theta < 6.28 , theta += dtheta)
        x = xc + r2 * cos(theta)
        y = yc + B * sin(theta)
        setpixel (Bound(x), Bound(y), color)
        DrawLine (xc, yc, Bound(x), Round(y), color)
    endfor
end
```

Scanned with CamScanner

2017:

The Barycentric equation of the triangle with vertices P_1, P_2, P_3 is:

$$P(t_1, t_2) = t_1 P_1 + t_2 P_2 + (1 - t_1 - t_2) P_3$$
$$0 \leq t_1, t_2 \leq 1$$
$$t_1 + t_2 \leq 1$$

1 write an algorithm to fill-in a triangle based on this equation

$t_2 + t_1 \leq 1 \Rightarrow t_2 \leq 1 - t_1$

```
for (t1 = 0.0 , t1 <= 1 , t1 += 1/max(P1,P2,P3) )
    for(t2 = 0.0, t2 <= 1-t1, t2 += 1/max(P1,P2,P3) )
        Point p = t1 * P1 + t2 * P2 + (1-t1-t2) P3
        Drawpixel (p.x, p.y, color)
    end for
end for
```

2 write an algorithm to determine whether a given point is inside the triangle.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = t_1 \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + t_2 \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} + (1 - t_1 - t_2) \begin{bmatrix} x_3 \\ y_3 \end{bmatrix}$$

$$x' = t_1 x_1 + t_2 x_2 + (1 - t_1 - t_2) x_3$$
$$y' = t_1 y_1 + t_2 y_2 + (1 - t_1 - t_2) y_3$$

$$x' - x_3 = (x_1 - x_3) t_1 + (x_2 - x_3) t_2$$
$$y' - y_3 = (y_1 - y_3) t_1 + (y_2 - y_3) t_2$$

matrix form $=$ $\begin{bmatrix} x' - x_3 \\ y' - y_3 \end{bmatrix} = \begin{bmatrix} x_1 - x_3 & x_2 - x_3 \\ y_1 - y_3 & y_2 - y_3 \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}$

Scanned with CamScanner

$$\because \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} = \frac{1}{(x_1-x_3)*(y_2-y_3)-(x_2-x_3)*(y_1-y_3)}$$

$$* \begin{bmatrix} y_2-y_3 & -(x_2-x_3) \\ -(y_1-y_3) & x_1-x_3 \end{bmatrix}$$

if: ① $t_1, t_2 \leq 1 \Rightarrow$ inside

$> 1 \Rightarrow$ outside

Algorithm:

```
for(t1=0.0 , t1≤1 , t1 += 1/max(P1,P2,P3) )
    for(t2=0.0, t2 ≤1-t1 , t2 += 1/max(P1,P2,P3) )
        if(t1+t2 > 1)
            continue
        else
            Point p = t1 * P1 + t2 P2 + (1-t1-t2) P3
            if(P.x ≤ P1.x && P.x ≤ P2.x && P.x ≤ P3.x
               && P.y ≤ P1.y && P.y ≤ P2.y &&
               P.y ≤ P3.y)
            the pixel is inside
```

2012:

Find the orthogonal projection of an object the plane passing through the points (1,-3,0), (2,4,1), (0,1,0)

$$\hat{u} = \frac{(1,-3,0)}{\sqrt{1^2+-3^2+0^2}} = \left(\frac{1}{\sqrt{10}}, \frac{-3}{\sqrt{10}}, 0\right)$$

$$\hat{v} = \frac{(2,4,1)}{\sqrt{2^2+4^2+1^2}} = \left(\frac{2}{\sqrt{21}}, \frac{4}{\sqrt{21}}, \frac{1}{\sqrt{21}}\right)$$

$$\hat{w} = \frac{(0,1,0)}{\sqrt{0^2+1^2+0^2}} = \left(0, \frac{1}{\sqrt{1}}, 0\right)$$

$\boxed{1}$ convert to $O(x,y,z)$:

$$C = \begin{bmatrix} \hat{u} \\ \hat{v} \\ \hat{w} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{10}} & \frac{-3}{\sqrt{10}} & 0 \\ \frac{2}{\sqrt{21}} & \frac{4}{\sqrt{21}} & \frac{1}{\sqrt{21}} \\ 0 & 1 & 0 \end{bmatrix}$$

$$P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

C^{-1} To convert from $O(x,y,z)$ to $O(u,v,w)$

$$C^{-1} = \begin{bmatrix} \frac{1}{\sqrt{10}} & \frac{2}{\sqrt{21}} & 0 \\ \frac{-3}{\sqrt{10}} & \frac{4}{\sqrt{21}} & \frac{1}{\sqrt{21}} \\ 0 & 1 & 0 \end{bmatrix}$$

$$\text{Composite} = C^{-1} \cdot P \cdot C$$

Scanned with CamScanner

- Find orthogonal projection of point (6,2,3) on plane passing through points (1,3,5) (2,4,1) and (0,1,0)

1- Construct 3 vectors

$u \rightarrow (1,3,5)$

$v \rightarrow (2,4,1)$

$w \rightarrow (0,1,0)$

2- Normalize vectors:

$$\hat{u} = \left(\frac{1}{\sqrt{1^2+3^2+5^2}}, \frac{3}{\sqrt{1^2+3^2+5^2}}, \frac{5}{\sqrt{1^2+3^2+5^2}}\right)$$

$$\hat{v} = \left(\frac{2}{\sqrt{2^2+4^2+1^2}}, \frac{4}{\sqrt{2^2+4^2+1^2}}, \frac{1}{\sqrt{2^2+4^2+1^2}}\right)$$

$$\hat{w} = (0, 1, 0)$$

$\therefore \hat{u} = (0.16, -0.507, 0.84)$

$\hat{v} = (0.436, 0.87, 0.218)$

$\hat{w} = (0, 1, 0)$

3- Convert to $o(x,y,z)$

$$C = \begin{bmatrix} \hat{u} \\ \hat{v} \\ \hat{w} \end{bmatrix} = \begin{bmatrix} 0.16 & -0.507 & 0.84 \\ 0.436 & 0.87 & 0.218 \\ 0 & 1 & 0 \end{bmatrix}$$

4- return to x, y, z origin

$$P = \begin{bmatrix} 6 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Scanned with CamScanner

5- Convert to $O(u, v, w)$:

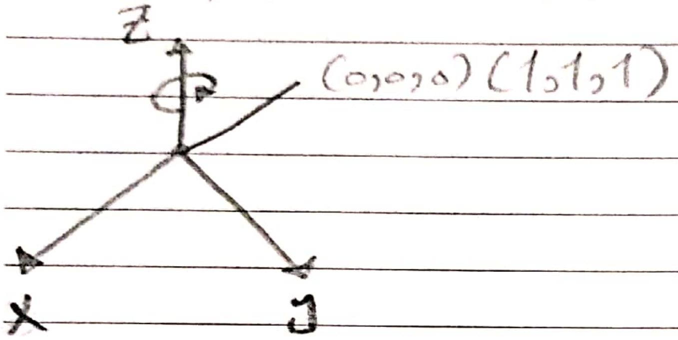
$$C^{-1} = \left[\hat{u} | \hat{v} | \hat{w}\right] = \begin{bmatrix} 0.16 & 0.436 & 0 \\ -0.507 & 0.87 & 1 \\ 0.84 & 1 & 0 \end{bmatrix}$$

Composite $= C^{-1} \cdot P \cdot C$

Scanned with CamScanner

2018:

write the transformation matrix and opengl code needed to rotate some point 60 degree clock wise about the line connecting the points (0,0,0) and (1,1,1).



[1] Translate line to origin so the line will be parell with z axes.

$$T = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

gltranslate(-1,-1,0)

[2] clockwise → from x to y and $-\theta$

Rotate the line about z clock wise.

$$R = \begin{bmatrix} \cos(60) & -\sin(60) & 0 & 0 \\ \sin(60) & \cos(60) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Scanned with CamScanner

Undo step 1:

$$T^{-1} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$\therefore$ Composite Matrix =

$T^{-1} \cdot R \cdot T$

Scanned with CamScanner

2018:

Find the affine transformation needed to compute the position of some 2D world position in the coordinate system having basis vectors $(0.6, 0.8)$ and origin $(20, 25)$
$(-0.8, 0.6)$

1. Construct the vectors:

$u \rightarrow (0.6, 0.8)$ $\quad$ $v \rightarrow (-0.8, 0.6)$

2. Normalize the vectors:

$$\hat{u} = \left(\frac{0.6}{\sqrt{(0.6)^2 + (0.8)^2}}, \frac{0.8}{\sqrt{(0.6)^2 + (0.8)^2}} \right) = (0.6, 0.8)$$

$$\hat{v} = \left(\frac{-0.8}{\sqrt{(-0.8)^2 + (0.6)^2}}, \frac{0.6}{\sqrt{(-0.8)^2 + (0.6)^2}} = (-0.8, 0.6) \right.$$

3. Translate to origin:

$$T_1 = \begin{bmatrix} 1 & 0 & -20 \\ 0 & 1 & -25 \\ 0 & 0 & 1 \end{bmatrix}$$

4. Rotat $(\hat{u}, \hat{v}) \rightarrow (x, y)$

$$B = \begin{bmatrix} 0.6 & 0.8 & 0 \\ -0.8 & 0.6 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Scanned with CamScanner

5. Undo step 3

$$T^{-1} = \begin{bmatrix} 1 & 0 & 20 \\ 0 & 1 & 25 \\ 0 & 0 & 1 \end{bmatrix}$$

composit Matrix

$= T^{-1} \cdot B \cdot T$

Scanned with CamScanner

2015:

* Trace the general polygon filling algorithm with:

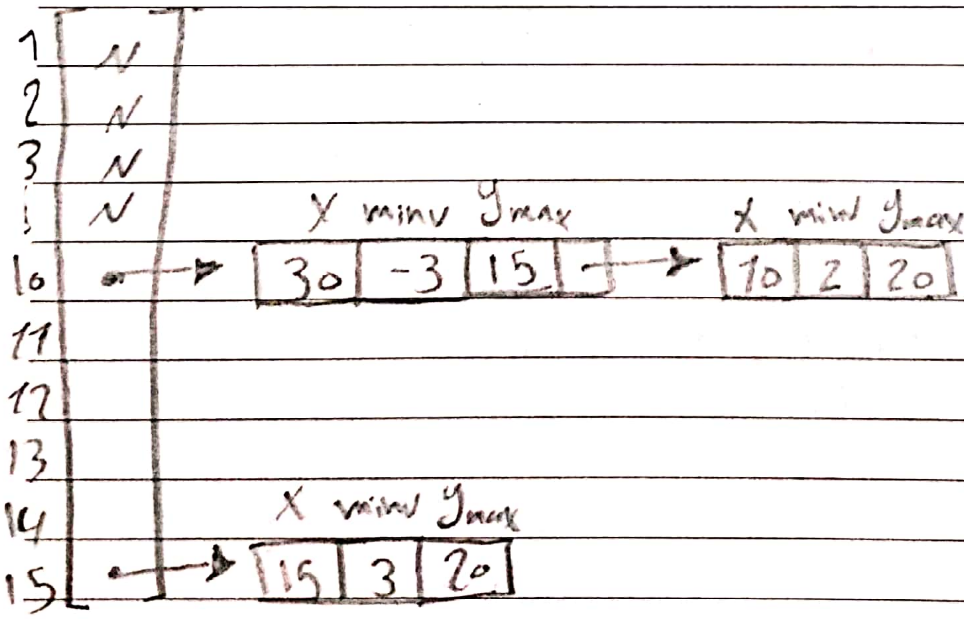
(30,10) - (15,15) - (30,20) - (10,10)

(30,10) (15,15) → $\frac{15-30}{15-10} = -3$

(15,15) (30,20) → $\frac{30-15}{20-15} = 3$

(30,20) (10,10) → $\frac{10-30}{10-20} = 2$

(10,10) (30,10) × have same y



$y = 10$

Sort ⇒ | 10 | 2 | 20 | ⟶ | 30 | -3 | 15 |

DrawLine (10, 10) → (30, 10)

$y = 11$

update ⇒ | 12 | 2 | 20 | ⟶ | 27 | -3 | 15 |

DrawLine (12, 11) → (27, 15)

$y = 12$

update ⇒ | 14 | 2 | 20 | ⟶ | 24 | -3 | 15 |

DrawLine (14, 12) → (24, 12)

$y = 13$

update ⇒ | 16 | 2 | 20 | ⟶ | 21 | -3 | 15 |

DrawLine (16, 13) → (21, 13)

$y = 14$

update ⇒ | 18 | 2 | 20 | ⟶ | 18 | -3 | 15 |

DrawLine (18, 14) → (18, 14)

$y = 15$

delete ⇒ | 18 | -3 | 15 |

update ⇒ | 20 | 2 | 20 |

Active | 15 | 3 | 20 | ⟶ | 20 | 2 | 20 |

DrawLine (15, 15) → (20, 15)

end #

Scanned with CamScanner

2015:

Rotate figure 1 to figure 2

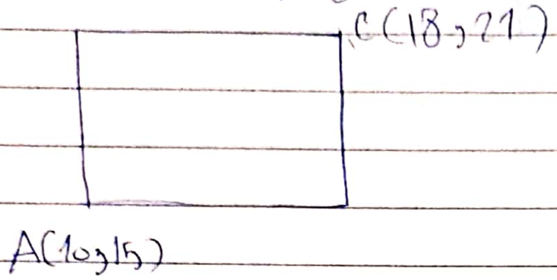


Figure (1)

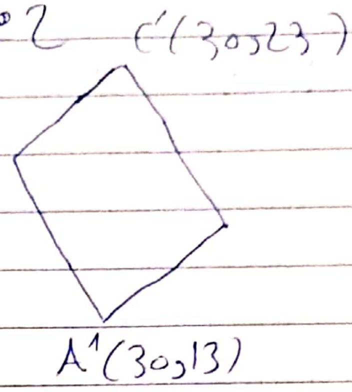


Figure (2)

$\boxed{1}$ Compute center of figure (1):

center $= \frac{10+18}{2} = 14$ $\quad \frac{21+15}{2} = 18$

$\boxed{2}$ Translate shape to origin:

$$T = \begin{bmatrix} 1 & 0 & -14 \\ 0 & 1 & -18 \\ 0 & 0 & 1 \end{bmatrix}$$

$\boxed{3}$ Rotate figure (2) at origin anti-clockwise:

$$R = \begin{bmatrix} \cos(45) & -\sin(45) & 0 \\ \sin(45) & \cos(45) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$\boxed{4}$ Compute center of figure (2)

center $= \frac{30+30}{2} = 30$ $\quad \frac{23+13}{2} = 18$

Scanned with CamScanner

$\boxed{5}$ Translate figure (2) to (30, 18):

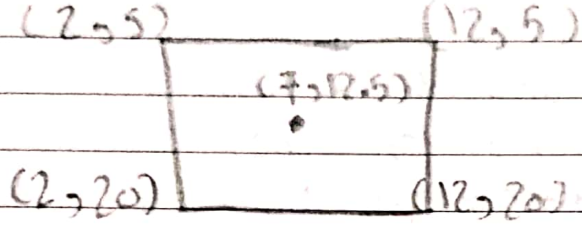
$$T^{-1} = \begin{bmatrix} 1 & 0 & 30 \\ 0 & 1 & 18 \\ 0 & 0 & 1 \end{bmatrix}$$

Composite Matrix =

$T^{-1} \cdot B \cdot T$

Scanned with CamScanner

Map A rectangle left = 2 top = 5 width = 10 height = 15
to squre centered at (8, 2) with lenght = 1



$\boxed{1}$ Calculate center of rectangle =

center = $\frac{12+2}{2} = 7$ $\quad \frac{20+5}{2} = 12.5$

$\boxed{2}$ Translate Rectangle to origin:

$$T = \begin{bmatrix} 1 & 0 & -7 \\ 0 & 1 & -12.5 \\ 0 & 0 & 1 \end{bmatrix}$$

$\boxed{3}$ Scale width to 1 lenght to 1:

$$S_1 = \begin{bmatrix} 1/10 & 0 & 0 \\ 0 & 1/15 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$\boxed{4}$ Translate from origin to squre center

$$T_2 = \begin{bmatrix} 1 & 0 & 8 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix}$$

∴ Composite matrix = $T_2 \cdot S_1 \cdot T$

2012:

write algorithm to draw spiral with parametric equations:

$$x = \theta \cos \theta \qquad \theta \in [0, 20\pi]$$

$$y = \theta \sin \theta$$

```
for (int theta=0, theta≤62.8, theta+= 1/(20*22/7)
    int x = theta * cos(cos(theta));
    int y = theta * sin(sin(theta));
    setpixel( Round(x), Round(y), C)
end for
```

Scanned with CamScanner

Derive an algorithm to compute the intersection between line segment and the Circle:

$$(x-x_c)^2+(y-y_c)^2=B^2$$

Use parametric equation of line:

$$x(t) = x_1 + (x_2-x_1)t$$
$$y(t) = y_1 + (y_2-y_1)t$$

blug parametric equation into circle:

$$(x_1+(x_2-x_1)t)^2+(y_1+(y_2-y_1)t)^2 = B^2$$

$$\left[x_1^2 + 2x_1(x_2-x_1)t + (x_2-x_1)^2t^2\right] + \left[y_1^2 + 2y_1(y_2-y_1)t + (y_2-y_1)^2t^2\right] = B^2$$

Using form of qudratic equation:

$$t^2\left[(x_2-x_1)^2+(y_2-y_1)^2\right] + t\left[2x(x_2-x_1)+2y(y_2-y_1)\right] + x_1^2+y_1^2-B^2=0$$

$$\Rightarrow At^2+Bt-c=0$$

$$t = \frac{-B \pm \sqrt{B^2-4AC}}{2A}$$

$\therefore d = B^2-4AC \Rightarrow$ determant

if $d < r$ then line cuts circle in two point

if $d = r$ then line and circle are tangent

if $d > r$ then line not cut circle in any point

Derive an algorithm to compute the intersection between line segment and the circle:

$$\frac{(x-x_c)^2}{a^2}+\frac{(y-y_c)^2}{b^2}=1$$

Use parametric equation of line:

$$x(t)=x_1+(x_2-x_1)t$$
$$y(t)=y_1+(y_2-y_1)t$$

blug parametric equation into ellipse:

$$\frac{(x_1+(x_2-x_1)t)^2}{a^2}+\frac{(y_1+(y_2-y_1)t)^2}{b^2}=1$$

$$\left[\frac{1}{a^2}(x_1^2+2x_1(x_2-x_1)t+(x_2-x_1)^2t^2)\right]+\left[\frac{1}{b^2}(y_1^2+2y_1(y_2-y_1)t+(y_2-y_1)^2t^2)\right]=1$$

using form of quadratic equation:

$$t^2\left[\frac{(x_2-x_1)^2}{a^2}+\frac{(y_2-y_1)^2}{b^2}\right]+t\left[\frac{2x(x_2-x_1)}{a^2}+\frac{2y(y_2-y_1)}{b^2}\right]+\left[\frac{x_1^2}{a^2}+\frac{y_1^2}{b^2}-1\right]=0$$

$$\Rightarrow At^2+Bt-c=0$$

$$t=\frac{-B\pm\sqrt{B^2-4Ac}}{2A}$$

$\therefore d=B^2-4AC \rightarrow$ determent

if $d<0$ the line and ellipse does not intersect

if $d=0$ the line and ellipse tangent

if $d>0$ the line and ellipse intersect

Scanned with CamScanner

Trace the convex polygon filling algorithm.
Compute the edge table values for a polygon of vertices (15,10), (80,12), (60,15), (10,16)

Initial table:

$$\begin{array}{cc} X_{left} & X_{right} \\ \left[\begin{matrix} \infty & -\infty \\ \infty & -\infty \\ \infty & -\infty \end{matrix}\right] \end{array}$$

$\boxed{1}$ (10,16) (15,10) swap $\Rightarrow$ (15,10) (10,16)

* $\therefore y=10 \quad x=15 \quad minv=-\frac{5}{6}$

$y=10 \quad [\,15 \quad 15\,] \rightarrow y{+}{+} \quad x{+}{=}minv$

* $\therefore y=11 \quad x=\frac{85}{6} \quad minv=-\frac{5}{6}$

$y=11 \quad [\,\frac{85}{6} \quad \frac{85}{6}\,] \rightarrow y{+}{+} \quad x{+}{=}minv$

* $\therefore y=12 \quad x=\frac{40}{3} \quad minv=-\frac{5}{6}$

$y=12 \quad [\,\frac{40}{3} \quad \frac{40}{3}\,] \rightarrow y{+}{+} \quad x{+}{=}minv$

* $\therefore y=13 \quad x=\frac{25}{2} \quad minv=-\frac{5}{6}$

$y=13 \quad [\,\frac{25}{2} \quad \frac{25}{2}\,] \rightarrow y{+}{+} \quad x{+}{=}minv$

* $\therefore y=14 \quad x=\frac{35}{3} \quad minv=-\frac{5}{6}$

$y=14 \quad [\,\frac{35}{3} \quad \frac{35}{3}\,] \rightarrow y{+}{+} \quad x{+}{=}minv$

* $\therefore y=15 \quad x=\frac{65}{6} \quad minv=-\frac{5}{6}$

$y=15 \quad [\,\frac{65}{6} \quad \frac{65}{6}\,] \rightarrow y{+}{+} \quad x{+}{=}minv$

* $y=16$ end of $\boxed{1}$

Scanned with CamScanner

$\boxed{2}$ $(15,10)$ $(80,12)$ $\rightarrow$ $\frac{80-15}{12-10} = \frac{65}{2}$

A $y=10$ $x=15$ $minv = \frac{65}{2}$

$y_{10} = [15 \quad 15] \rightarrow y{+}{+} \quad x{+}{=}minv$

A: $y=11$ $x = \frac{95}{2}$ $minv = \frac{65}{2}$

$y_{11} = [\frac{85}{2} \quad \frac{95}{2}] \rightarrow y{+}{+} \quad x{+}{=}minv$

A: $y=12$ end of $\boxed{2}$

$\boxed{3}$ $(80,12)$ $(60,15)$ $\rightarrow$ $\frac{60-80}{15-12} = \frac{-20}{3}$

A $y=12$ $x=80$ $minv = \frac{-20}{3}$

$y=12$ $[\frac{40}{3} \quad 80] \rightarrow y{+}{+} \quad x{+}{=}minv$

A $y=13$ $x = \frac{220}{3}$ $minv = \frac{-20}{3}$

$y=13$ $[\frac{25}{2} \quad \frac{220}{3}] \rightarrow y{+}{+} \quad x{+}{=}minv$

A $y=14$ $x = \frac{200}{3}$ $minv = \frac{-20}{3}$

$y=14$ $[\frac{35}{3} \quad \frac{200}{3}] \rightarrow y{+}{+} \quad x{+}{=}minv$

A $y=15$ end of $\boxed{3}$

Scanned with CamScanner

EdgeTable =

$$\begin{array}{c} 10 \\ 11 \\ 12 \\ 13 \\ 14 \\ 15 \end{array}
\begin{bmatrix} 15 & 15 \\ \frac{85}{6} & \frac{95}{2} \\ \frac{40}{3} & 80 \\ \frac{25}{2} & \frac{220}{3} \\ \frac{35}{3} & \frac{200}{3} \\ \frac{65}{6} & \frac{65}{6} \end{bmatrix}$$

Scanned with CamScanner