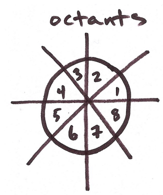
CPSC 314, Written Homework 4 Solutions

Value: 4% of final grade
Total Points: 111

Rasterization (15 pts)

1. (15 pts) Give an algorithm for scan-converting a line with the Bresenham approach that works in the seventh octant (lines with slope between negative infinity and -1), rather than the first octant as described in class (lines with slope between 0 and 1).

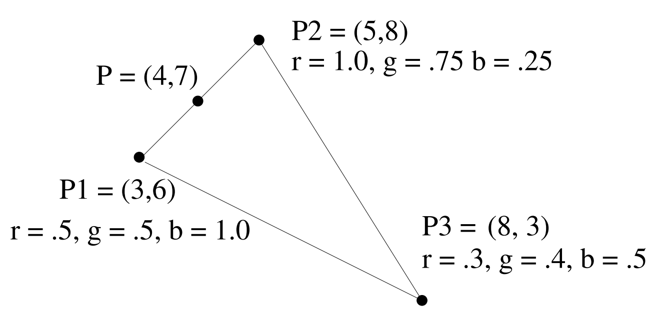


Solution

$x \leftarrow 0$
$d_x \leftarrow x_1 - x_0$
$d_y \leftarrow y_1 - y_0$
$d \leftarrow d_y + 2d_x$
$incKeepX \leftarrow 2d_x$
$incIncreaseX \leftarrow 2d_y + 2d_x$
for $y \leftarrow y_0$ to y_1 **do**
    draw(x, y)
    if $d > 0$ **then**
        $x \leftarrow x + 1$
        $d \leftarrow d + incIncreaseX$
    else
        $d \leftarrow d + incKeepX$

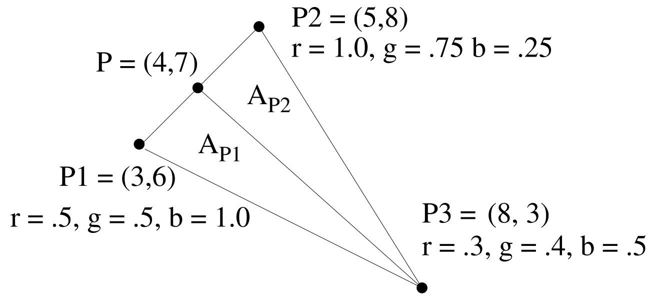
Interpolation (6 pts)

2. (6 pts) Find the barycentric coordinates α, β, and γ for P, and use them to interpolate the (r, g, b) color components at that point. Show your work.



Solution

Since P lies on the line between P_1 and P_2, $A_{P3} = 0$. By inspection, we see that P is half-way between P_1 and P_2. Therefore, $A_{P1} = A_P 2 = \frac{1}{2}A$, where A is the area of the triangle formed by P_1, P_2, P_3. We calculate the barycentric coordinates as follows:



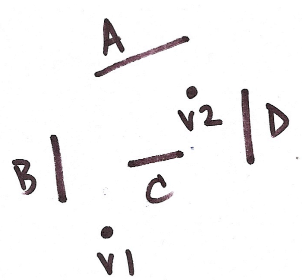
$$\alpha = A_{P1}/A = .5$$
$$\beta = A_{P2}/A = .5$$
$$\gamma = A_{P3}/A = 0$$

Using these, we calculate the colour components at P:

$$\begin{aligned} P_{rgb} &= \alpha \cdot P1_{rgb} + \beta \cdot P2_{rgb} + \gamma \cdot P3_{rgb} \\ &= .5 \cdot (.5, .5, 1.0) + .5 \cdot (1.0, .75, .25) + 0 \cdot (.3, .4, .5) \\ &= (.75, .625, .625) \end{aligned}$$

Visibility (42 pts)

3. (24 pts) Build a BSP tree for the following scene using the polygons (shown as line segments). The cutting plane induced by a polygon should just extend along the line itself. The labelled side of the polygon should be the right child in the tree, and the unlabelled side should be the left child.



a) Give the BSP tree with the single root node of polygon A, and sketch the entire scene with the addition of the new cutting plane.

b) Same as above, building on the previous answer, after adding polygon B.

c) Same as above, building on the previous answer, after adding polygon C.

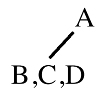
d) Same as above, building on the previous answer, after adding polygon D.

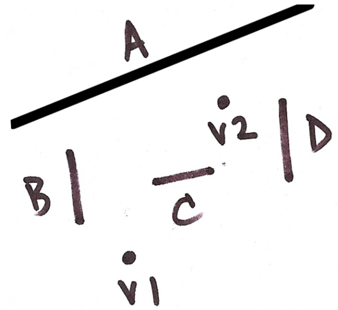
e) Traverse your BSP tree to produce a painter's algorithm ordering from eye point V1. Show your work at each step in the traversal, starting from the root of the BSP tree.

f) Same as above, instead using eye point V2.

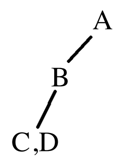
Solution

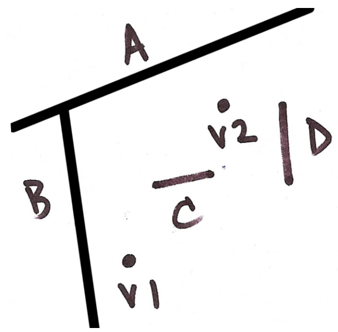
a)



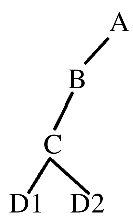


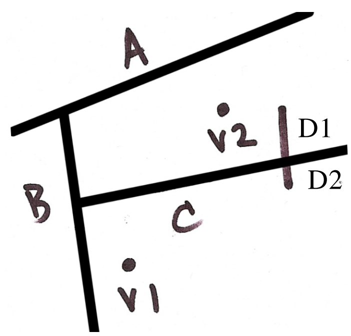
b)



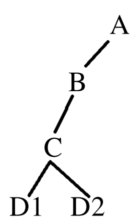


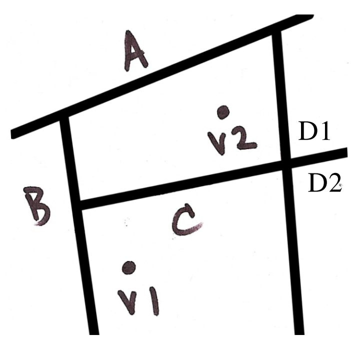
c)



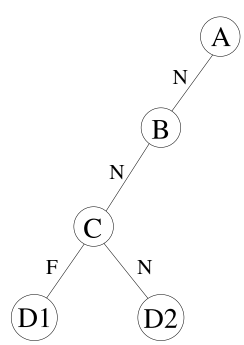


d)





e)

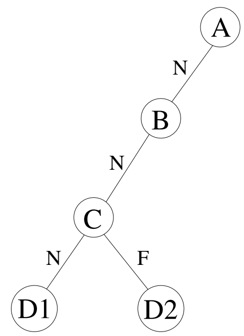


Nothing on the far side of A
Draw A
Traverse near side of A

Nothing on the far side of B
Draw B
Traverse near side of B
Traverse far side of C
Nothing on the far side of $D1$
Draw $D1$
Nothing on the near side of $D1$
Draw C
Traverse near side of C
Nothing on the far side of $D2$
Draw $D2$
Nothing on the near side of $D2$

The order of drawing is: $A, B, D1, C, D2$.

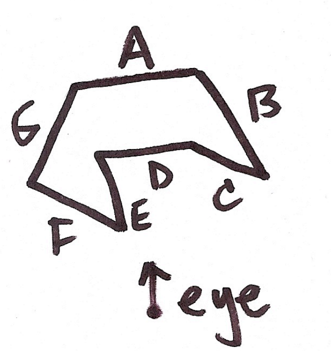
f)



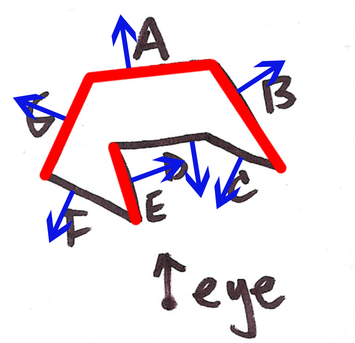
Nothing on the far side of A
Draw A
Traverse near side of A
Nothing on the far side of B
Draw B
Traverse near side of B
Traverse far side of C
Nothing on the far side of $D2$
Draw $D2$
Nothing on the near side of $D2$
Draw C
Traverse near side of C
Nothing on the far side of $D1$
Draw $D1$
Nothing on the near side of $D1$

The order of drawing is: $A, B, D2, C, D1$.

4. (2 pts) For the following 2D scene, an eye point is given with respect to an object formed by line segments. Which faces would be removed for the given eyepoint if backface culling were used? Show your work by drawing in the normals for each face.



Solution



The faces A, B, E, and G are culled.

5. (16 pts) You have bought a very cheap graphics card, which has a Z buffer of only 3 bits. You can thus only determine the visibility relationships of objects in your scene at a very coarse resolution: there are only $2^3 = 8$ bins available. These bins are represented as the base-10 integers 0 through 7. You should assume that the general OpenGL perspective matrix was used for projection, with the near plane set to .5 and the far plane set to 50.

a) Give the z-values of the planes forming the boundaries of these bins in DCS, the display coordinate system (which ranges from 0.0 at the near plane to 1.0 at the far plane). That is, what is the value of the plane between bin 0 and bin 1, between bin 1 and bin 2, and so on.

b) Give the z-values of the planes in the camera coordinate system.

Solution

a)

Plane	**z value**
near plane	0
between bin 0 and 1	.125
between bin 1 and 2	.25
between bin 2 and 3	.375
between bin 3 and 4	.5
between bin 4 and 5	.625
between bin 5 and 6	.75
between bin 6 and 7	.875
far plane	1

b) To solve this problem, we will need the following equations:

$$Z_{buf} = 1 \ll N \cdot (a + b/z)$$
$$a = \frac{z_{far}}{z_{far} - z_{near}}$$
$$b = \frac{z_{far} \cdot z_{near}}{z_{near} - z_{far}}$$

Here N, the number of bits, is 3, so $1 \ll N = 8$. We can rearrange to solve for z.

$$z = \frac{b}{(Z_{buf}/8) - a} \tag{1}$$

We need to solve equation 1 for each value of the Z-buffer given $z_{near} = .5$ and $z_{far} = 50$.

$$a = \frac{50}{50 - .5} = 1.0101$$
$$b = \frac{50 \cdot .5}{.5 - 50} = -.5051$$
$$z = \frac{-.5051}{(Z_{buf}/8) - 1.0101}$$

Plugging in values from 0 to 8 for Z_{buf} produces the following table:

Plane	z value
near plane	0.5
between bin 0 and 1	.5706
between bin 1 and 2	.6645
between bin 2 and 3	.7952
between bin 3 and 4	.9901
between bin 4 and 5	1.3115
between bin 5 and 6	1.9417
between bin 6 and 7	3.7383
far plane	50

Clipping (32 pts)

6. (16 pts) Clip the line segment with endpoints A = (2,2) and B = (-1,-2) to the box (-1,-1), (1,-1), (1,1), (-1,1). Use the Cohen-Sutherland algorithm.

   a) Show the full configuration after clipping against the top of the box with a sketch, including any new points you may need to create. If you need to create new points, give them names in alphabetical order (D, E, F, ...). Compute the exact coordinates of any new points that result from intersecting lines. (Provide the final values as decimal numbers.) Show your intermediate work, including outcodes for vertices.

   b) Same as above, building on the previous answer, after clipping against the bottom edge.

   c) Same as above, building on the previous answer, after clipping against the right edge.

   d) Same as above, building on the previous answer, after clipping against the left and final edge of the box.

Solution

Recall the following map of outcodes where bits are ordered top, bottom, left, right:

1010	1000	1001
0010	0000	0001
0110	0100	0101

a) We first compute the outcodes of A and B and check for the trivial cases.

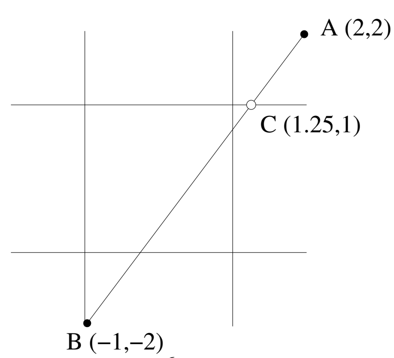
$$\begin{aligned} A_{out} &= 1001 \\ B_{out} &= 0110 \\ A_{out}|B_{out} &= 1111 \neq 0000 \\ A_{out}\&B_{out} &= 0000 = 0000 \implies \text{ we must clip} \end{aligned}$$

Since we need to clip against the top edge, we calculate the intersection point $C = (x_C, 1)$ using the slope, m, and endpoints of the line AB.

$$m = \frac{y_A - y_B}{x_A - x_B} = \frac{2-(-2)}{2-(-1)} = \frac{4}{3}$$

$$x_C = x_B + (1 - y_B)/m = -1 + 3/\tfrac{4}{3} = 1.25$$

Therefore the intersection point C is at $(1.25, 1)$.



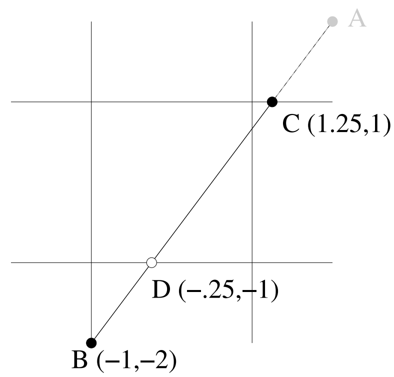
b) We compute the outcodes of B and C and check for the trivial cases.

$$\begin{aligned} B_{out} &= 0110 \\ C_{out} &= 0001 \\ B_{out}|C_{out} &= 0111 \neq 0000 \\ B_{out}\&C_{out} &= 0000 = 0000 \implies \text{we must clip} \end{aligned}$$

We need to clip against the bottom edge, so we calculate the intersection point $D = (x_D, -1)$ using the line BC. The slope m is the same as in part a.

$$x_D = x_B + (-1 - y_B)/m = -1 + 1/\tfrac{4}{3} = -.25$$

Therefore the intersection point D is at $(-.25, -1)$.



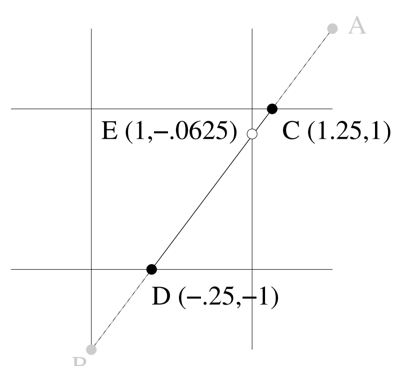
c) We compute the outcodes of C and D and check for the trivial cases.

$$\begin{aligned} C_{out} &= 0001 \\ D_{out} &= 0000 \\ C_{out}|D_{out} &= 0001 \neq 0000 \\ C_{out}\&D_{out} &= 0000 = 0000 \implies \text{we must clip} \end{aligned}$$

We need to clip against the right edge, so we calculate the intersection point $E = (1, y_E)$ using the line CD. The slope m is the same as in part a.

$$y_E = y_D + m(1 - x_D) = -1 + \tfrac{3}{4}(1.25) = -.0625$$

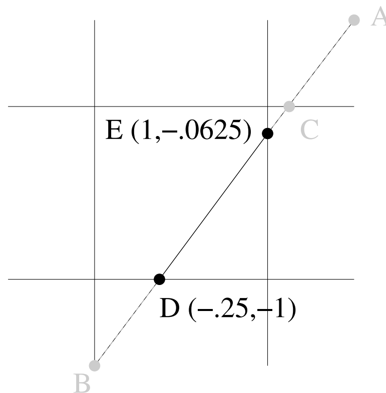
Therefore the intersection point E is at $(1, -.0625)$.



d) We compute the outcodes of D and E and check for the trivial cases.

$$\begin{aligned} D_{out} &= 0000 \\ E_{out} &= 0000 \\ D_{out}|E_{out} &= 0000 = 0000 \implies \text{no clipping necessary} \end{aligned}$$

Both points D and E are within the box, so we do not need to clip against the left edge.



7. (16 pts) Clip the polygon with points A = (-1, 2), B = (-1.5, -.5), C = (1.5, -1.5) against the box (-1,-1), (1,-1), (1,1), (-1,1). Use the Sutherland-Hodgeman algorithm to clip the polygon.

   a) Give the vertex list after clipping against the top edge, and draw a sketch showing the configuration including the location of any new points you may need to create. If you need to create new points, give them names in alphabetical order (C, D, E, ...). Use the same labels for the new points in your sketch and in the vertex list. Show your intermediate work, in terms of inside/outside checks for each vertex pair. You do **not** need to compute exact values of the intersection points.

   b) Same as above, building on the previous answer, after clipping against the bottom edge.

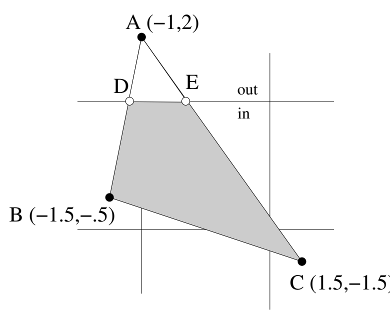
   c) Same as above, building on the previous answer, after clipping against the right edge.

   d) Same as above, building on the previous answer, after clipping against the left and final edge of the box.

Solution

a)

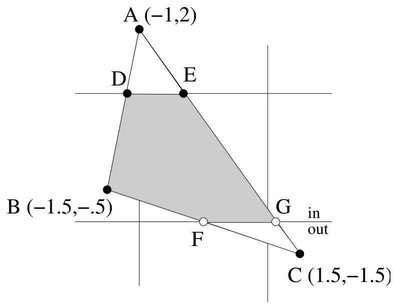
Vertex Pair	Inside/Outside	Output Vertices
A	out	
$A \to B$	out $\to$ in	D, B
$B \to C$	in $\to$ in	C
$C \to A$	in $\to$ out	E



We get vertices D, B, C, E after clipping against the top edge.

b)

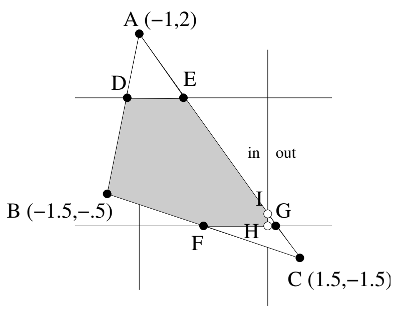
Vertex Pair	Inside/Outside	Output Vertices
D	in	D
$D \to B$	in $\to$ in	B
$B \to C$	in $\to$ out	F
$C \to E$	out $\to$ in	G, E
$E \to D$	in $\to$ in	



We get vertices D, B, F, G, E after clipping against the bottom edge.

c)

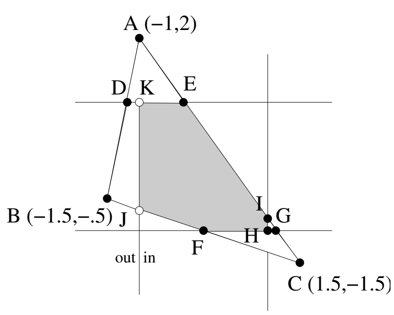
Vertex Pair	**Inside/Outside**	**Output Vertices**
D	in	D
$D \to B$	in $\to$ in	B
$B \to F$	in $\to$ in	F
$F \to G$	in $\to$ out	H
$G \to E$	out $\to$ in	I, E
$E \to D$	in $\to$ in	



We get vertices D, B, F, H, I, E after clipping against the bottom edge.

d)

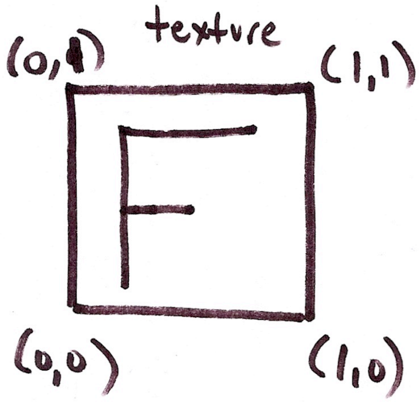
Vertex Pair	**Inside/Outside**	**Output Vertices**
D	out	
$D \to B$	out $\to$ out	
$B \to F$	out $\to$ in	J, F
$F \to H$	in $\to$ in	H
$H \to I$	in $\to$ in	I
$I \to E$	in $\to$ in	E
$E \to D$	in $\to$ out	K



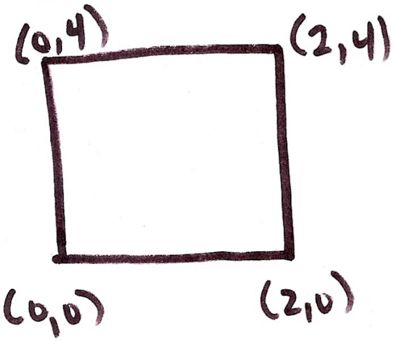
We get vertices J, F, H, I, E, K after clipping against the left edge.

Textures (16 pts)

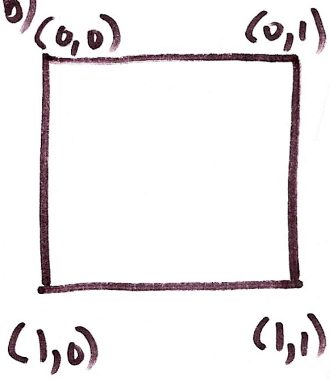
8. (16 pts) In the following figure, sketch the texture (top) as it would appear in each of the rectangles with the specified texture coordinates. Assume the texture mode is GL_REPEAT.



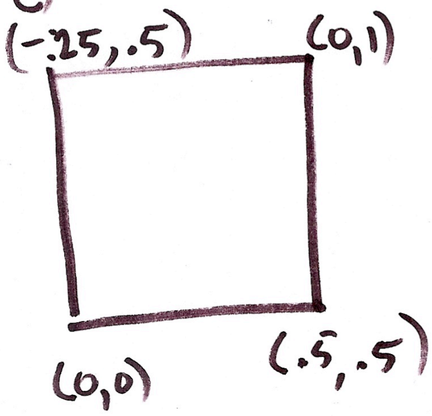
a)



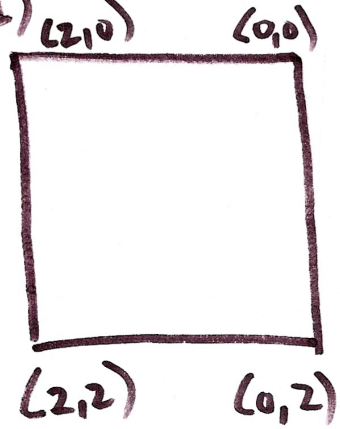
b)



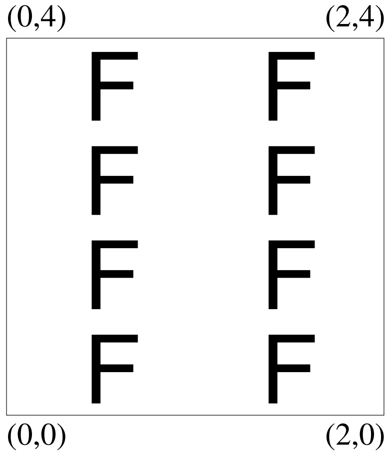
c)



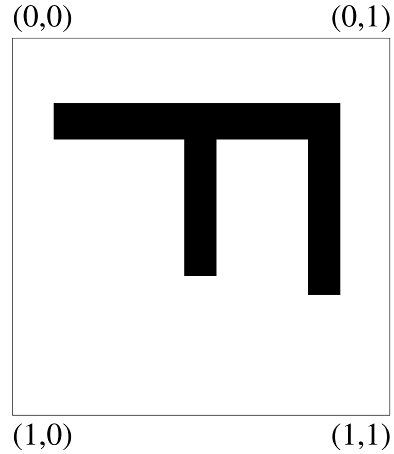
d)



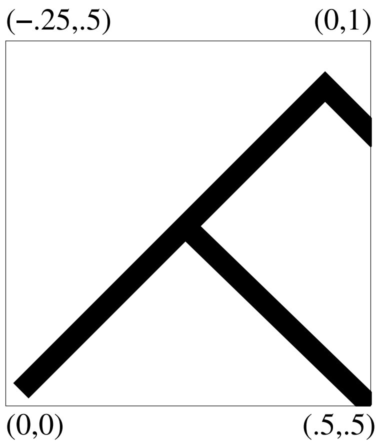
Solution



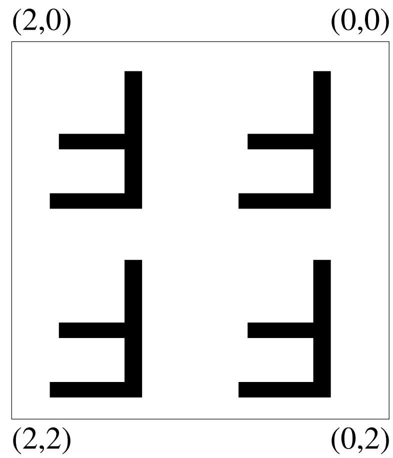
a)



b)



c)



d)