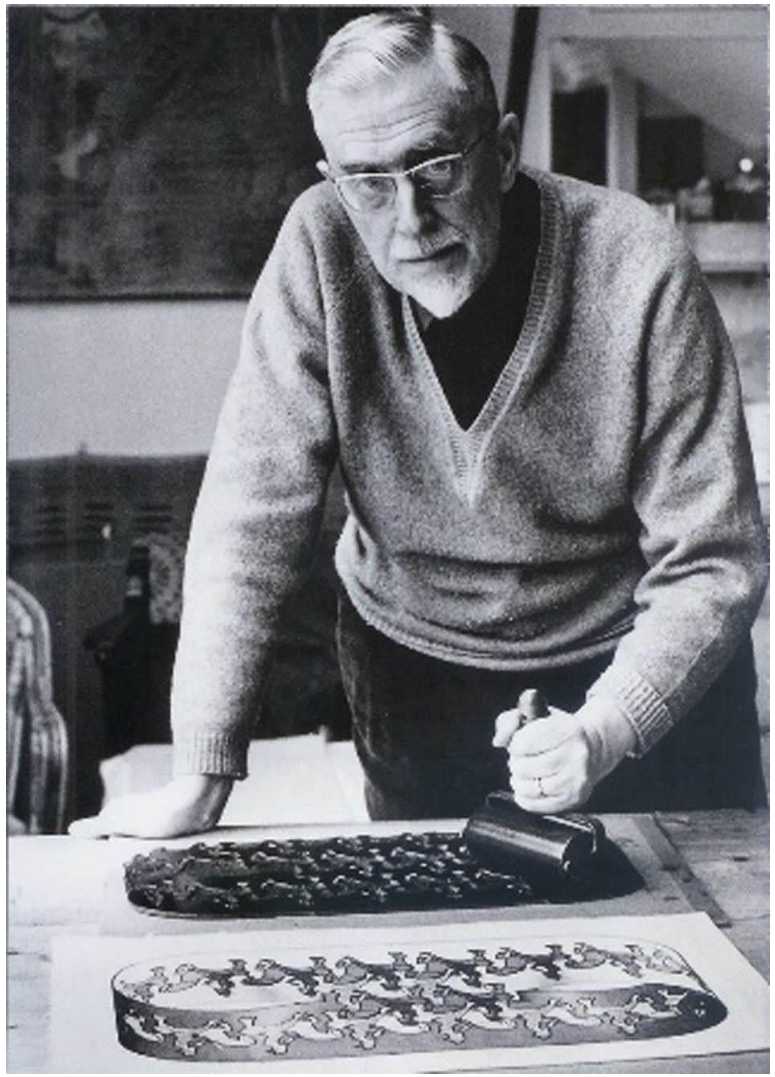
Maurits Cornelis Escher (1898-1972)

Leeuwarden, June 17



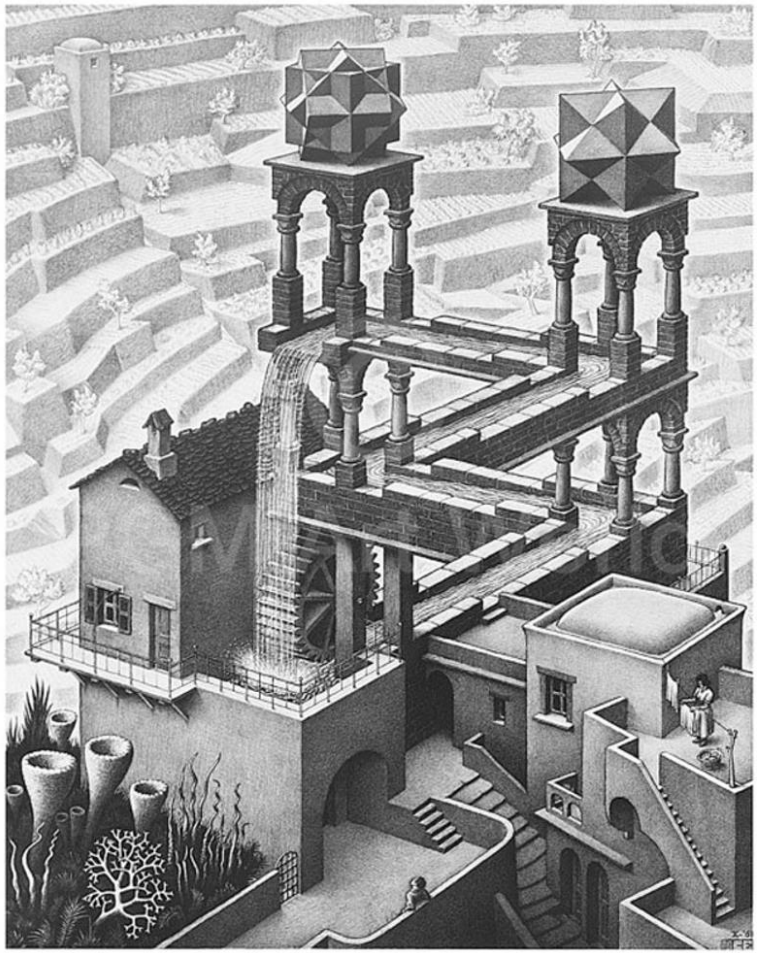
Escher

Graphical artist

Escher

Graphical artist

Optical illusions



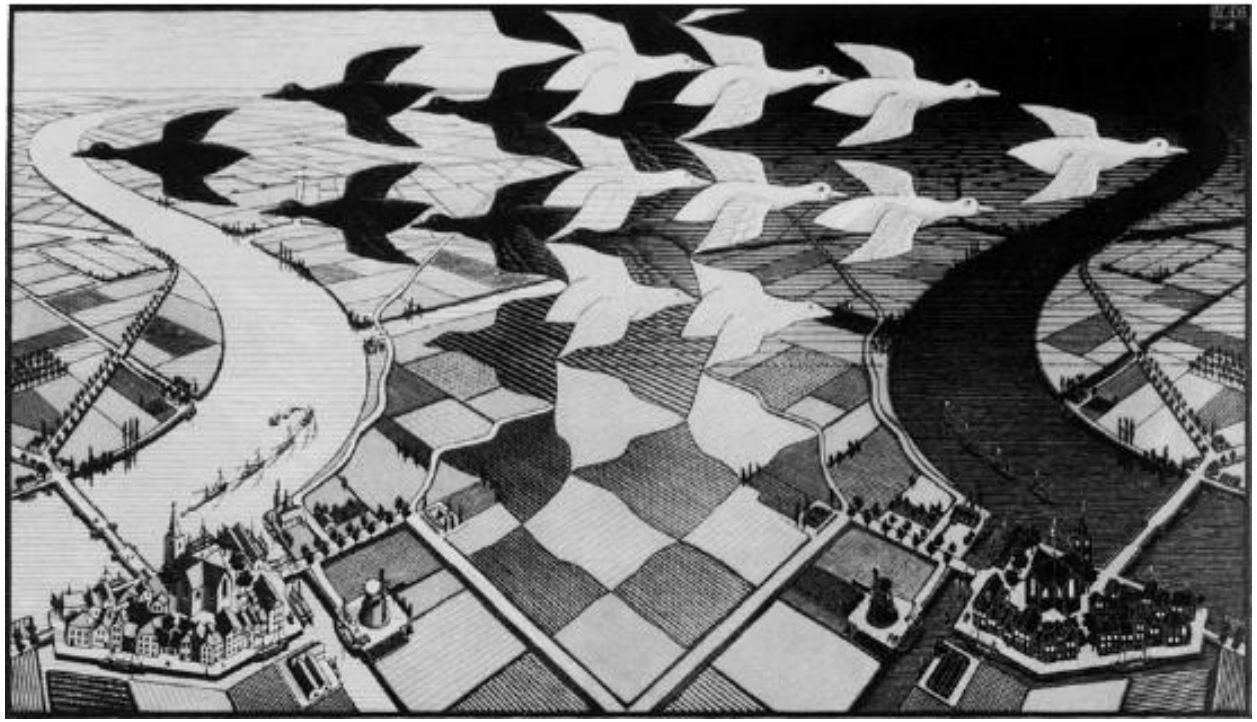
"Waterfall", 1961 Lithograph

Escher

Graphical artist

Optical illusions

Transformations



"Day and night", 1938 woodcut

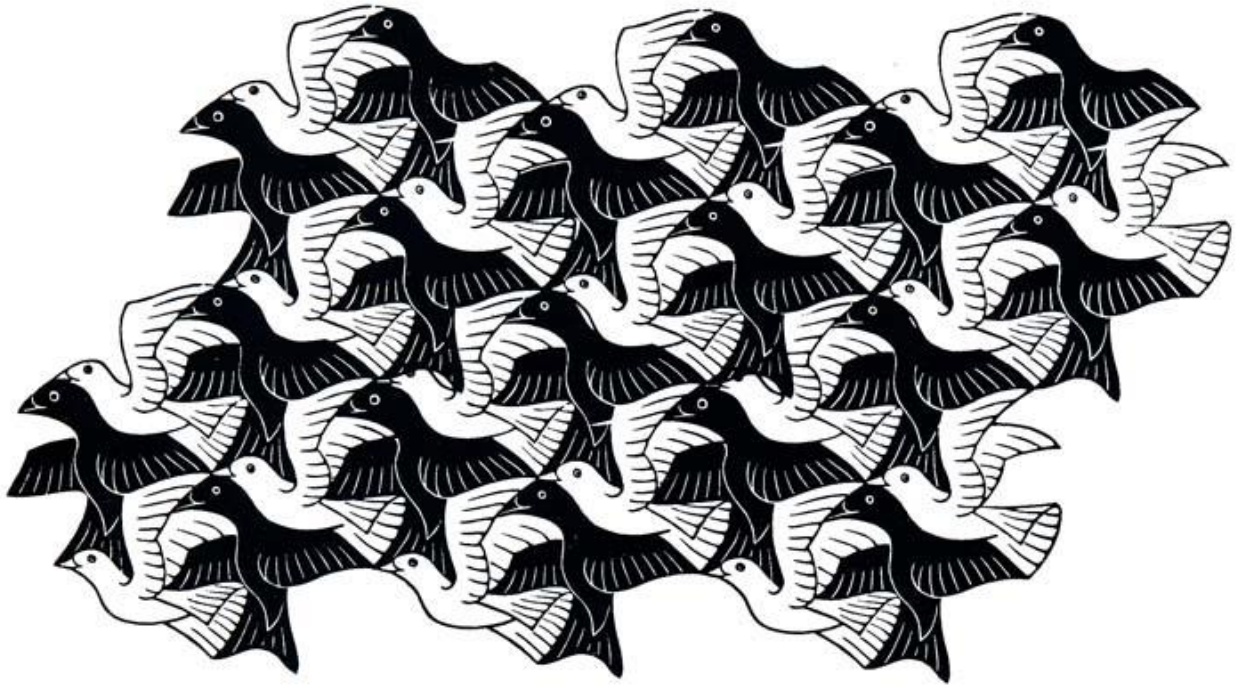
Escher

Graphical artist

Optical illusions

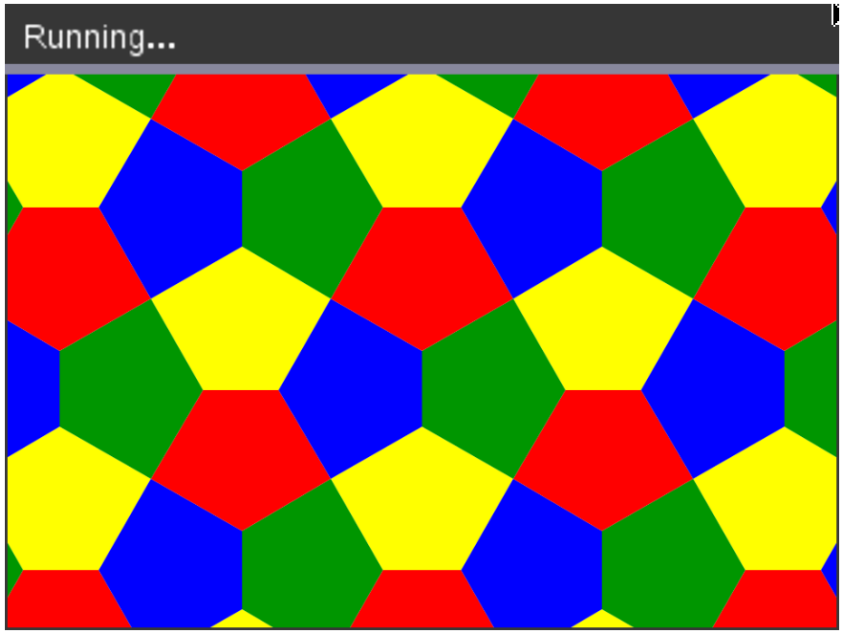
Transformations

Tessellations



"Birds", 1949 woodcut

Tessellations

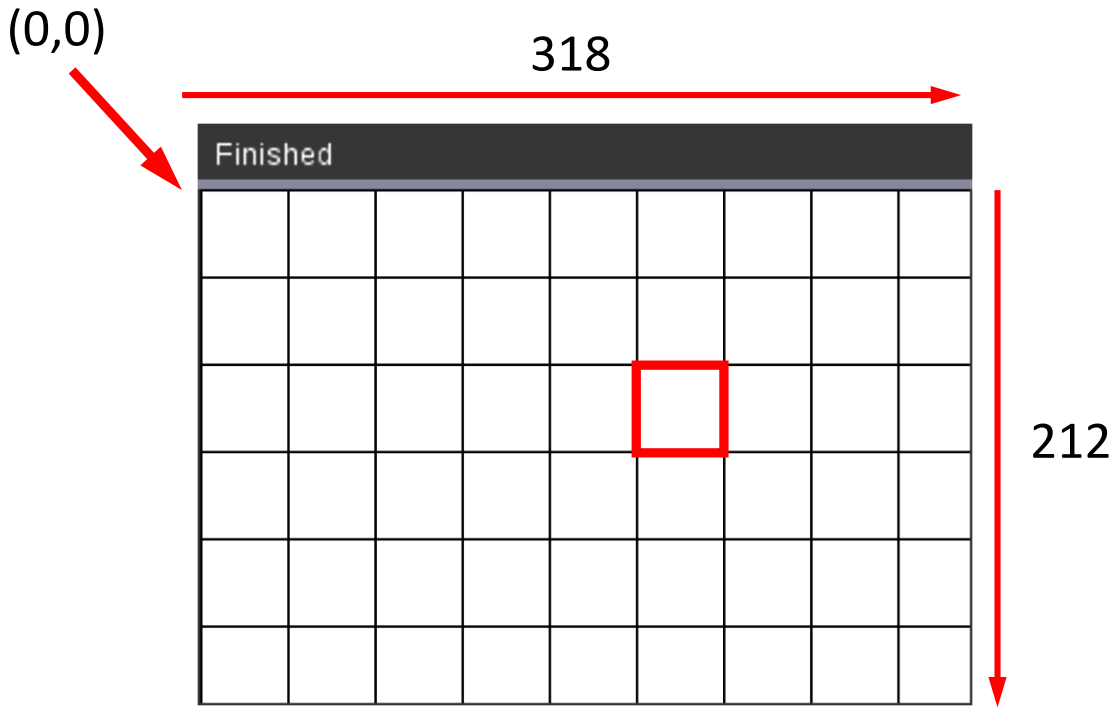


Regular tessellation

Rectangles

```
1.1 1.2 1.3 *01 Recta...les RAD
rectangles1.py 10/11
from ti_draw import *

a=36

for i in range(6):
    for j in range(9):
        x=j*a
        y=i*a
        draw_rect(x,y,a,a)
```

i : row number
j : column number



$i = 2$ $\quad x = j \cdot 36 = 180$

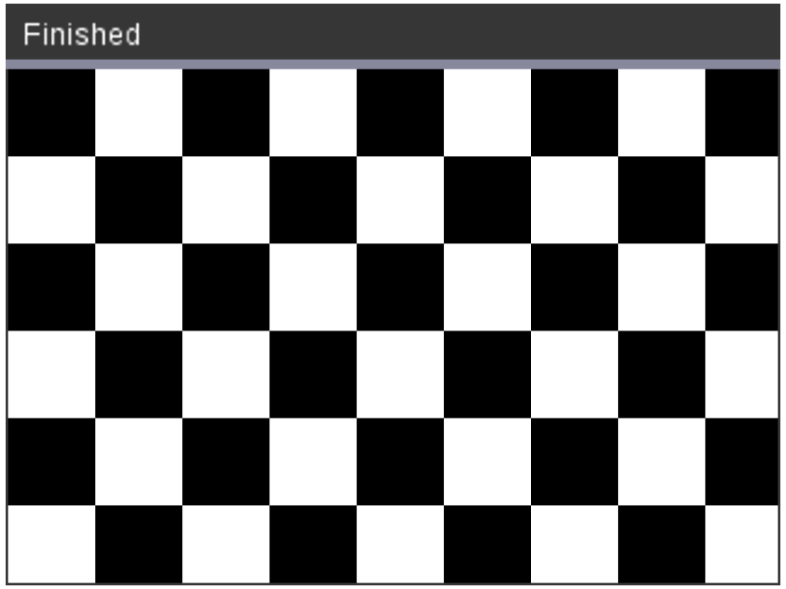
$j = 5$ $\quad y = i \cdot 36 = 72$

Rectangles

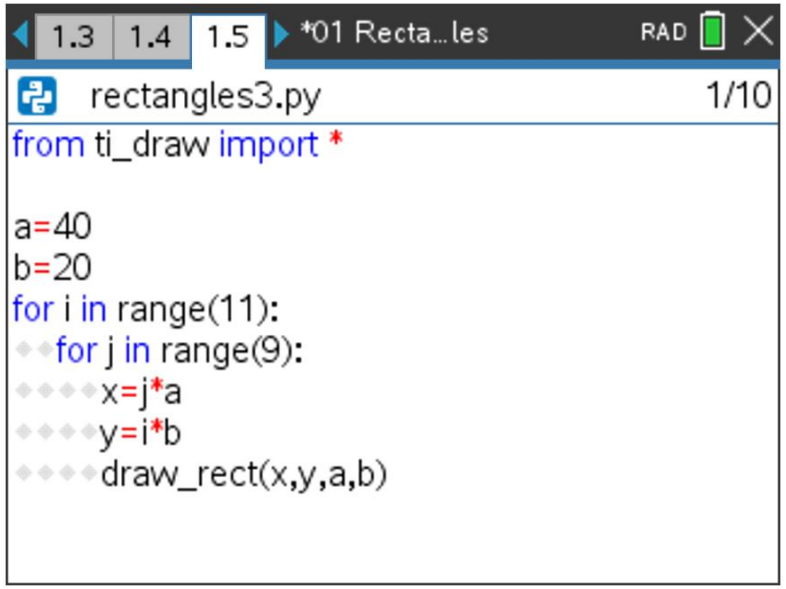
1.2 1.3 1.4 *01 Recta...les RAD

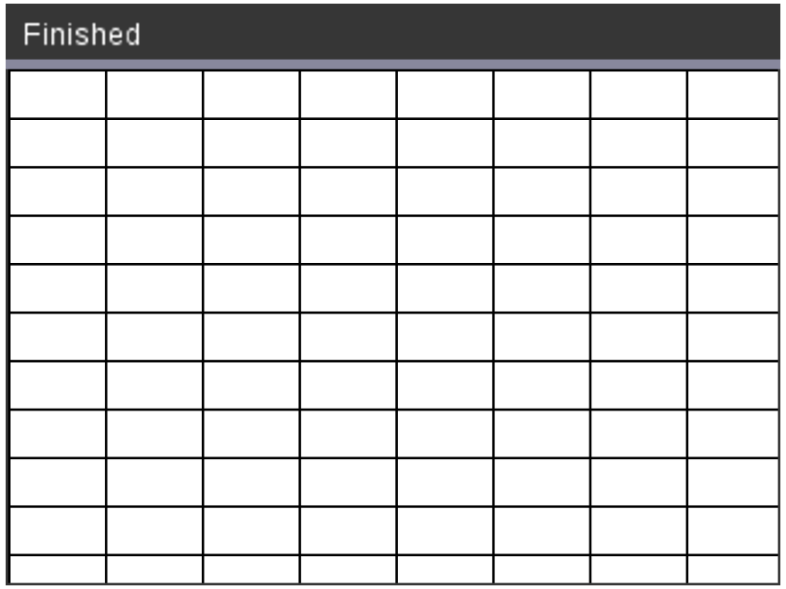
rectangles2.py 2/11

```
from ti_draw import *

a=36

for i in range(6):
    for j in range(9):
        x=j*a
        y=i*a
        if (i+j)%2 == 0:
            fill_rect(x,y,a,a)
```

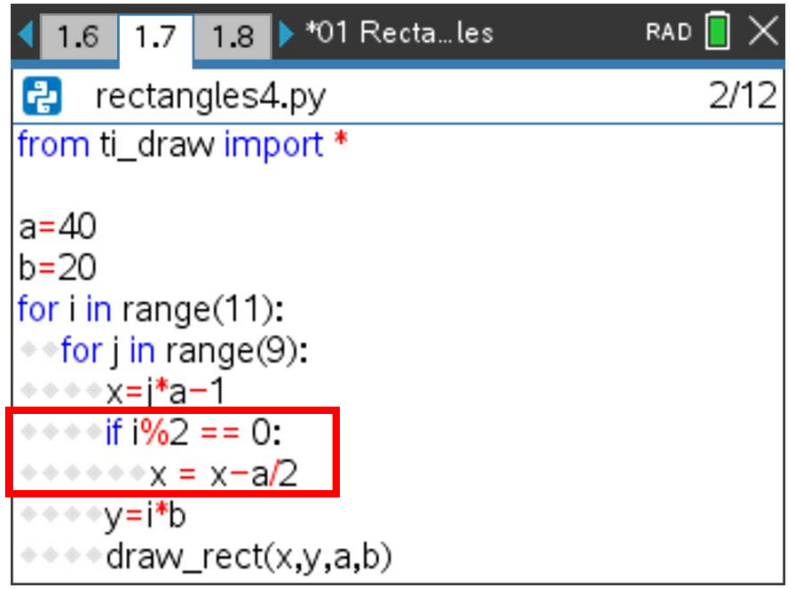


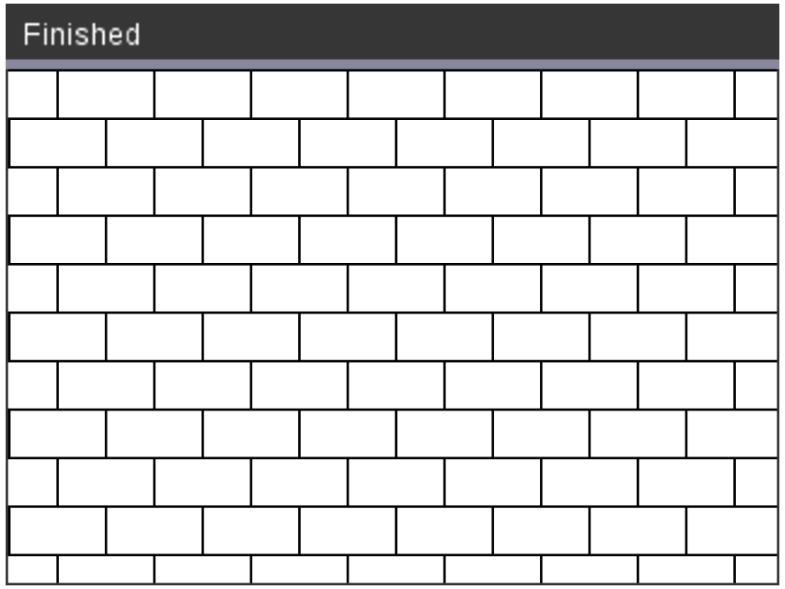
Rectangles


1.3 1.4 1.5 *01 Recta...les RAD
rectangles3.py 1/10
from ti_draw import *
a=40
b=20
for i in range(11):
for j in range(9):
x=j*a
y=i*b
draw_rect(x,y,a,b)


Finished

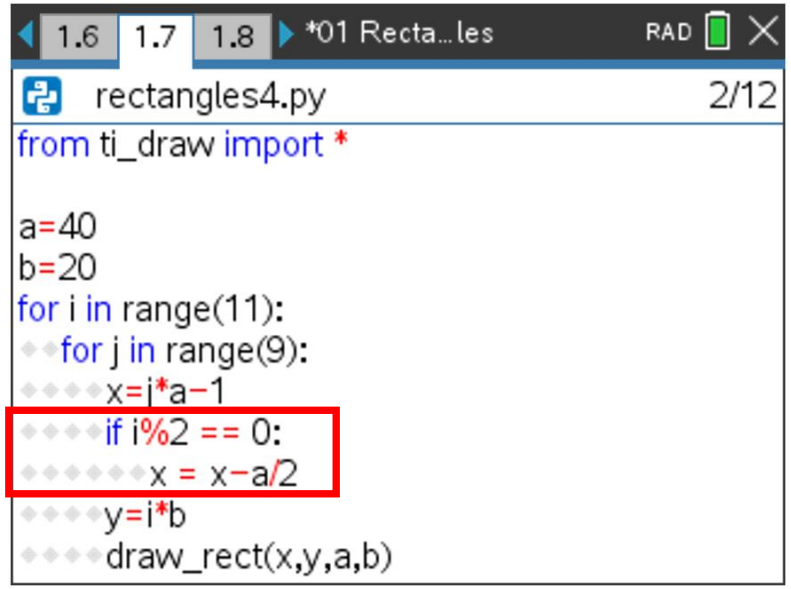
Rectangles

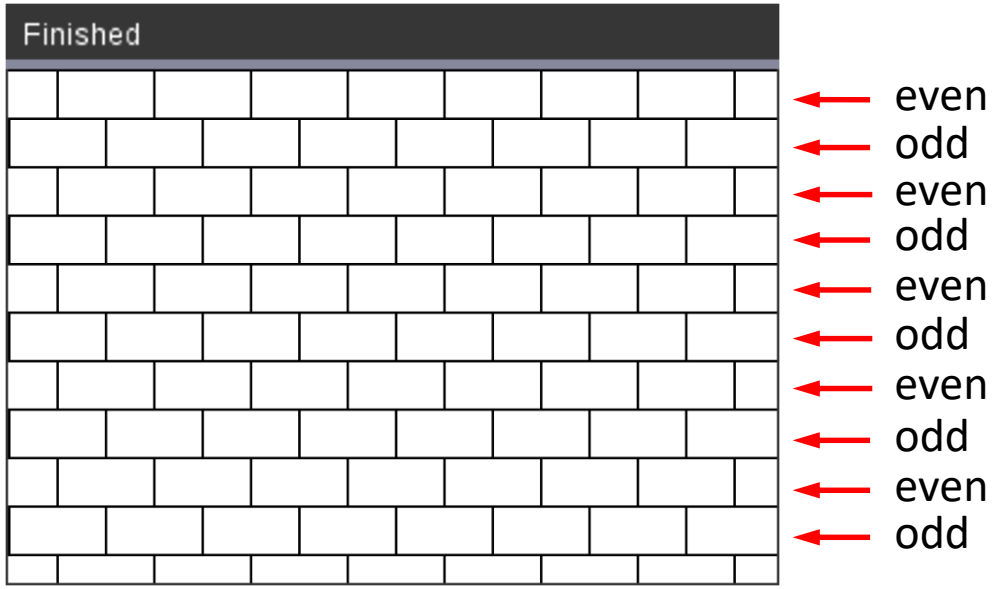




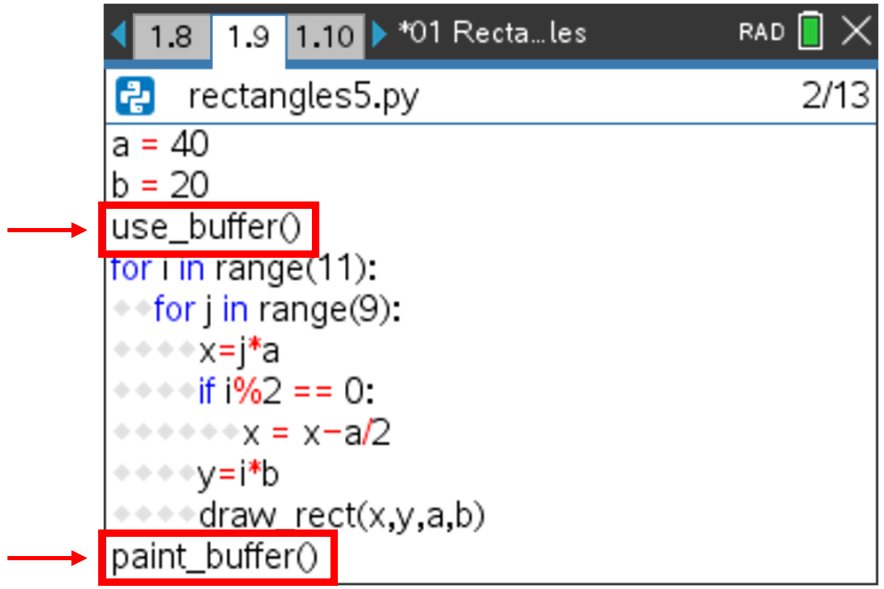
p % q returns the remainder of p divided by q

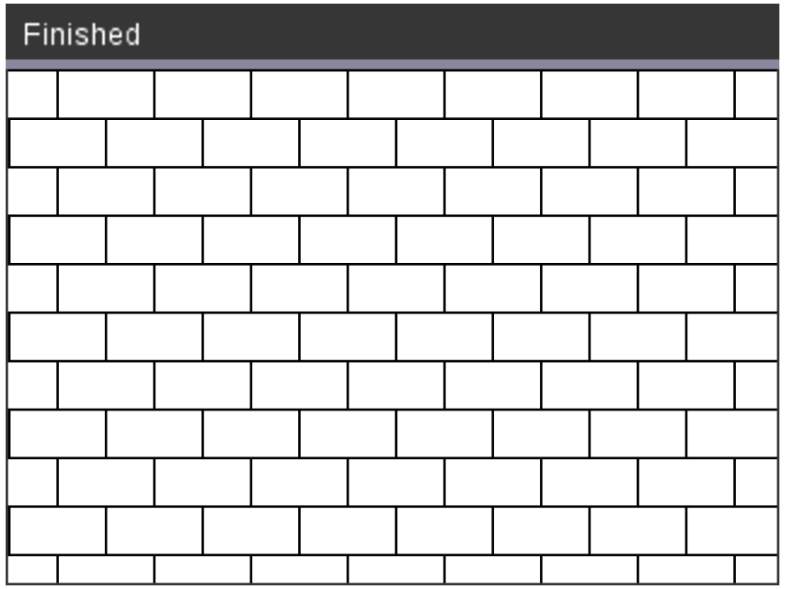
Rectangles


1.6 1.7 1.8 *01 Recta...les RAD
rectangles4.py 2/12
from ti_draw import *
a=40
b=20
for i in range(11):
for j in range(9):
x=j*a-1
if i%2 == 0:
x = x-a/2
y=i*b
draw_rect(x,y,a,b)


Finished
even
odd
even
odd
even
odd
even
odd
even
odd

Rectangles


1.8
1.9
1.10
*01 Recta...les
RAD
rectangles5.py
2/13
a = 40
b = 20
use_buffer()
for i in range(11):
for j in range(9):
x=j*a
if i%2 == 0:
x = x-a/2
y=i*b
draw_rect(x,y,a,b)
paint_buffer()


Finished

Polygons

draw_rect (x, y, width, height)

fill_rect (x, y, width, height)

draw_poly (xlist, ylist)

fill_poly (xlist, ylist)

Polygons

draw_rect (x, y, width, height)

fill_rect (x, y, width, height)

draw_poly (xlist, ylist)

fill_poly (xlist, ylist)

Polygons

Module tessellations

multiply (polygon, factor)

rotate (polygon, angle)

mirror_hor (polygon)

mirror_vert (polygon)

draw (x, y, polygon)

fill (x, y, polygon)

x_correction (polygon)

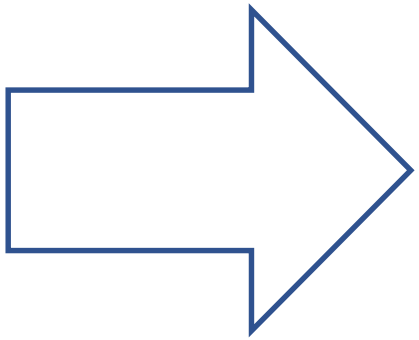
y_correction (polygon)

} *isometric grid*

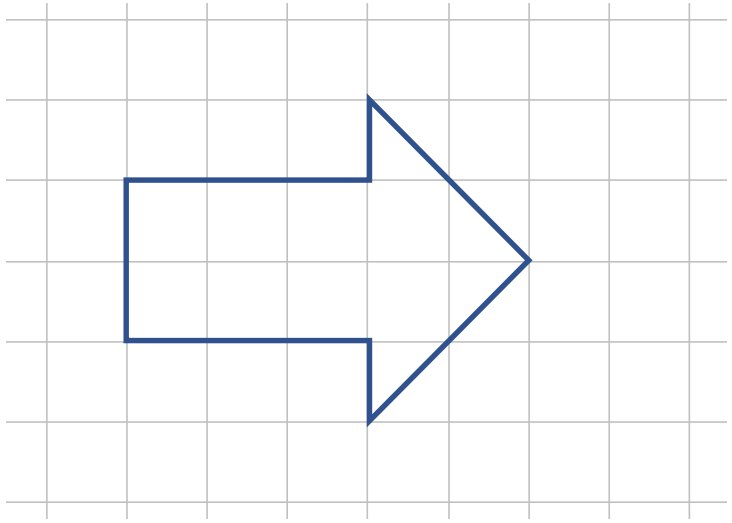
Polygon is a list with vertices

Window(-159,159,-106,106)

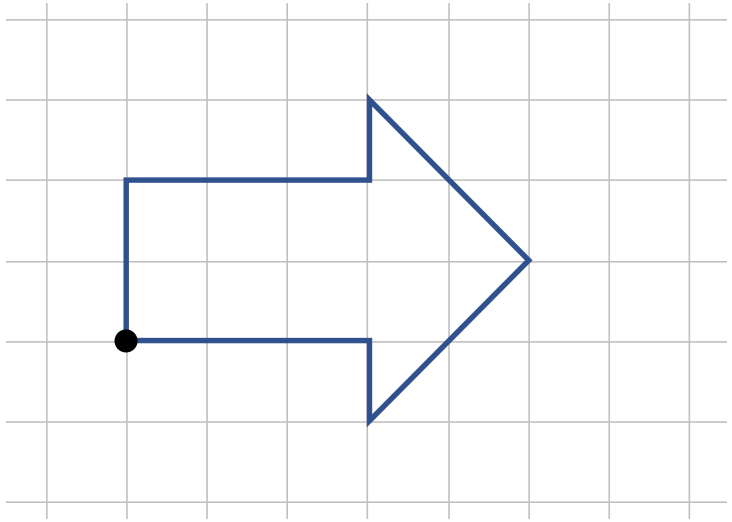
Polygons



Polygons

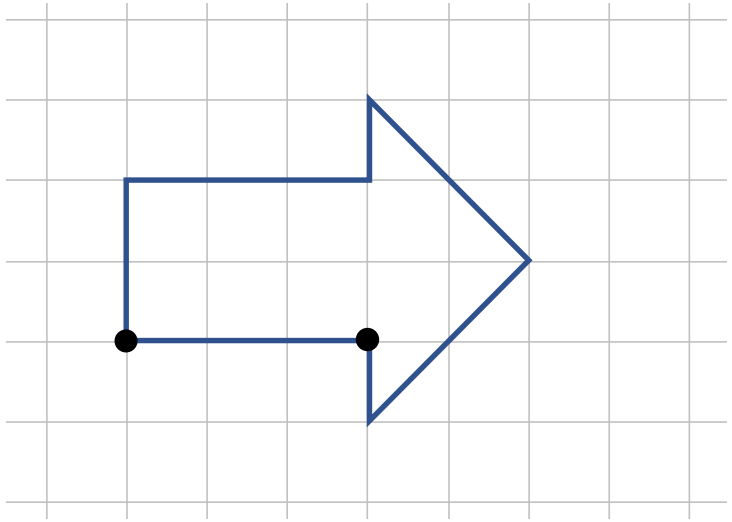


Polygons



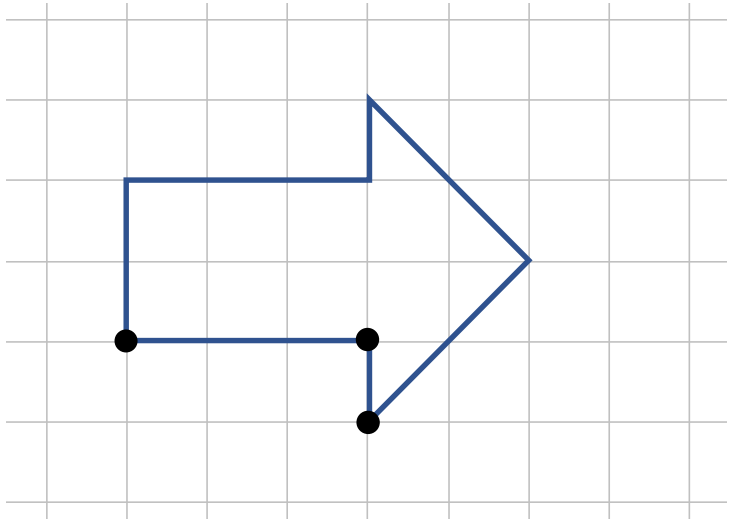
arrow = [$(0,0)$

Polygons



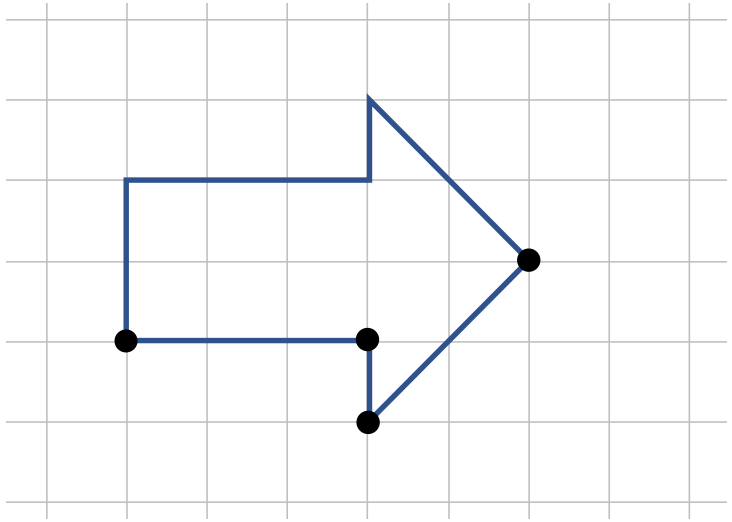
arrow = [$(0,0), (3,0)$

Polygons



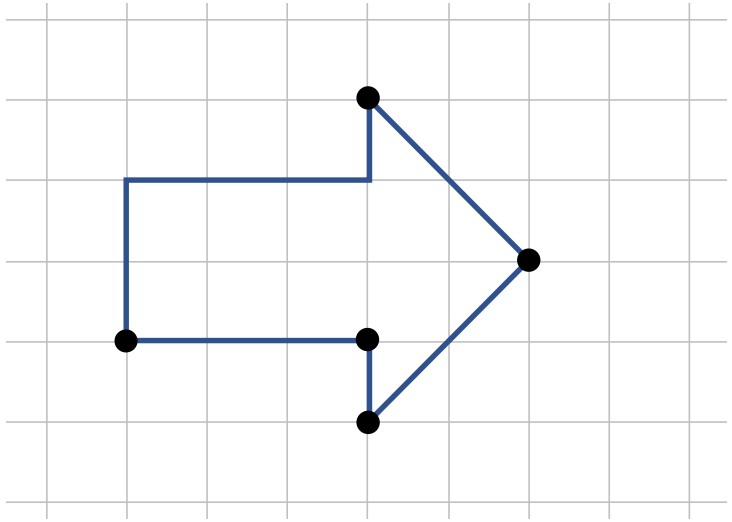
arrow = [$(0,0), (3,0), (3,-1)$

Polygons



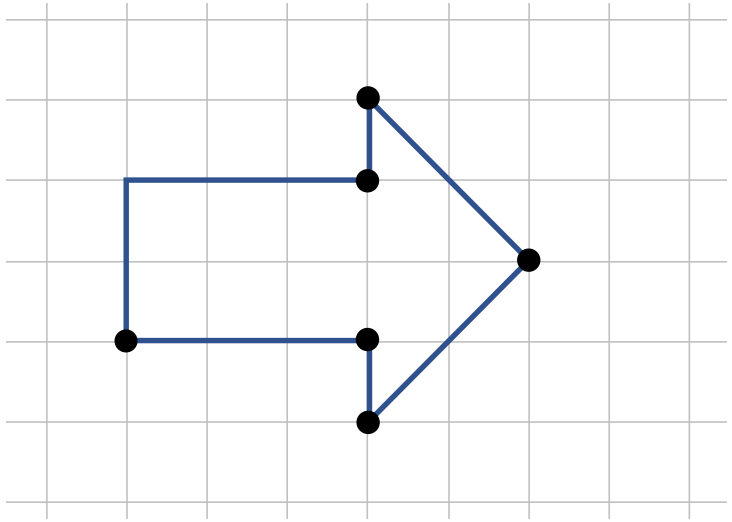
arrow = [$(0,0), (3,0), (3,-1), (5,1)$

Polygons



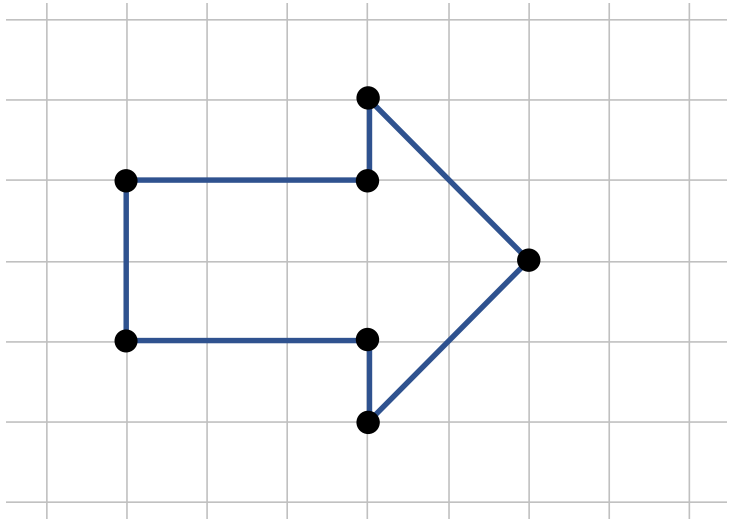
arrow = [$(0,0), (3,0), (3,-1), (5,1), (3,3)$

Polygons



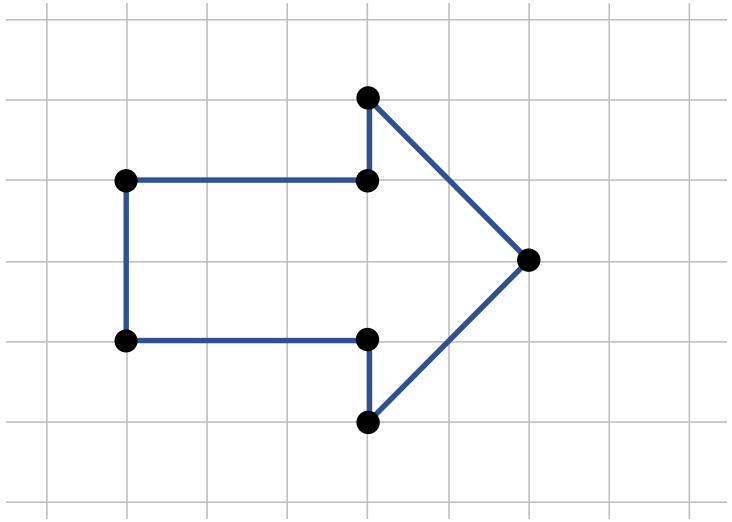
arrow = [$(0,0), (3,0), (3,-1), (5,1), (3,3), (3,2)$

Polygons



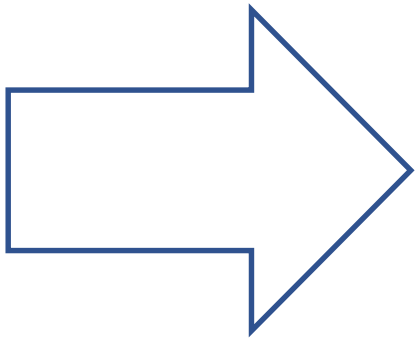
arrow = [$(0,0), (3,0), (3,-1), (5,1), (3,3), (3,2), (0,2)$

Polygons



arrow = [$(0,0), (3,0), (3,-1), (5,1), (3,3), (3,2), (0,2), (0,0)$]

Polygons



arrow = [$(0,0), (3,0), (3,-1), (5,1), (3,3), (3,2), (0,2), (0,0)$]

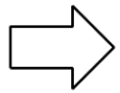
Polygons

1.1 1.2 1.3 *02 Arrows RAD

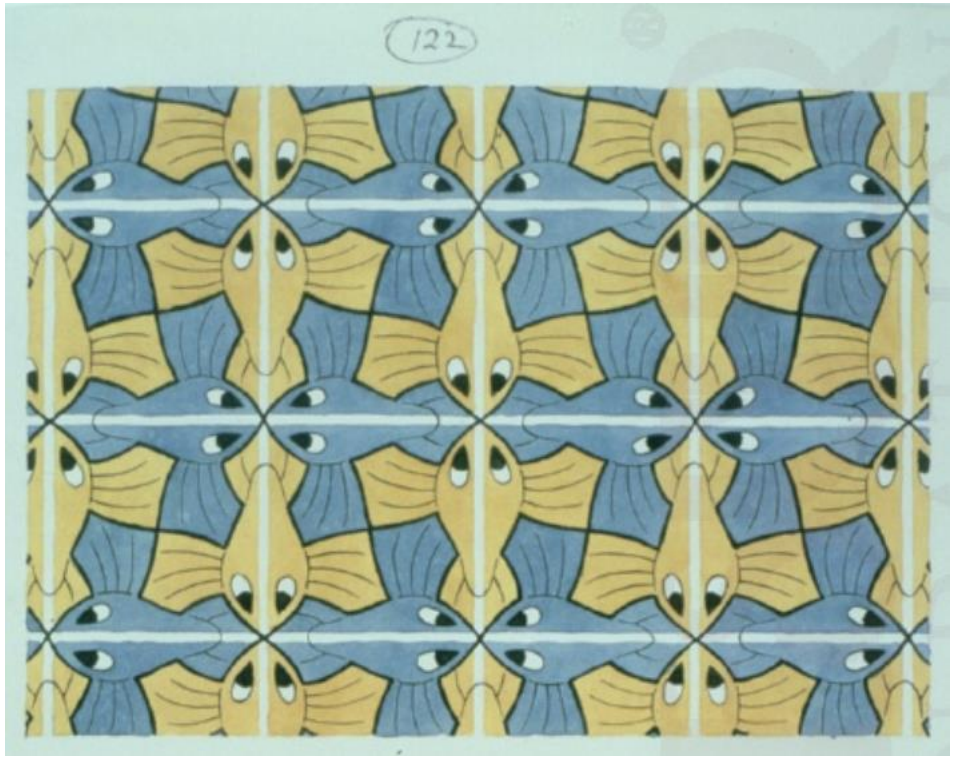
arrows1.py 8/10

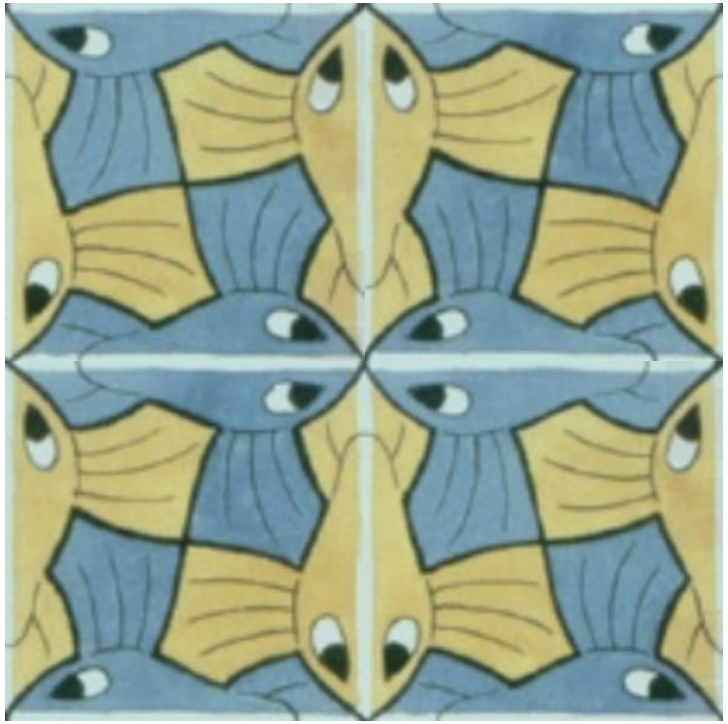
```
from tessellation import *

arrow = [
  (0,0), (3,0), (3,−1), (5,1),
  (3,3), (3,2), (0,2), (0,0)]

arrow = multiply(arrow, 8)

draw(0, 0, arrow)
```

Finished



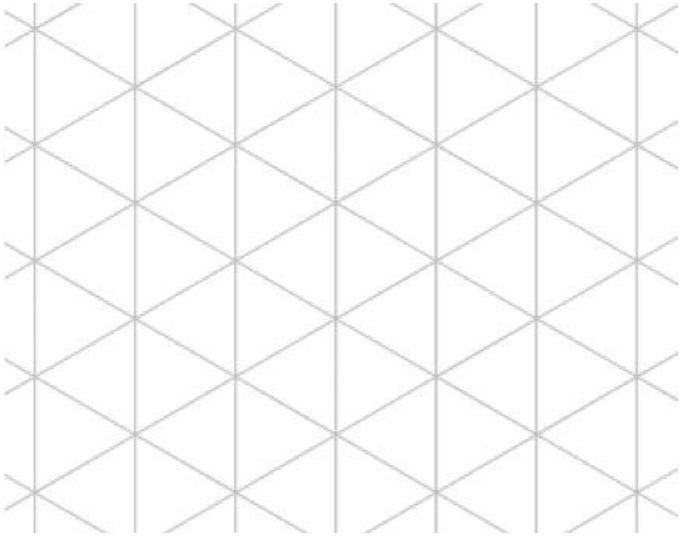
Polygons



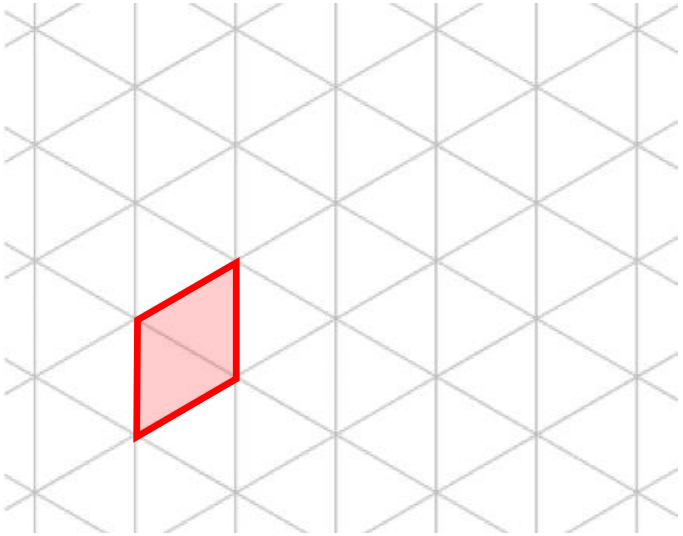


Example 1

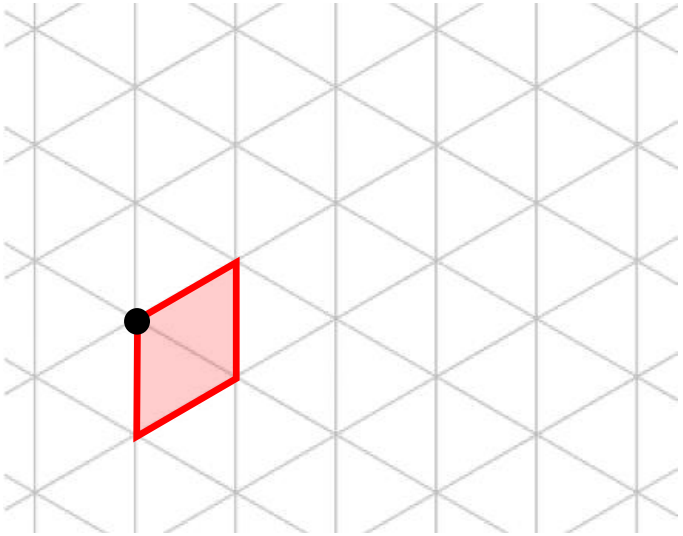
Isometric grid



Isometric grid

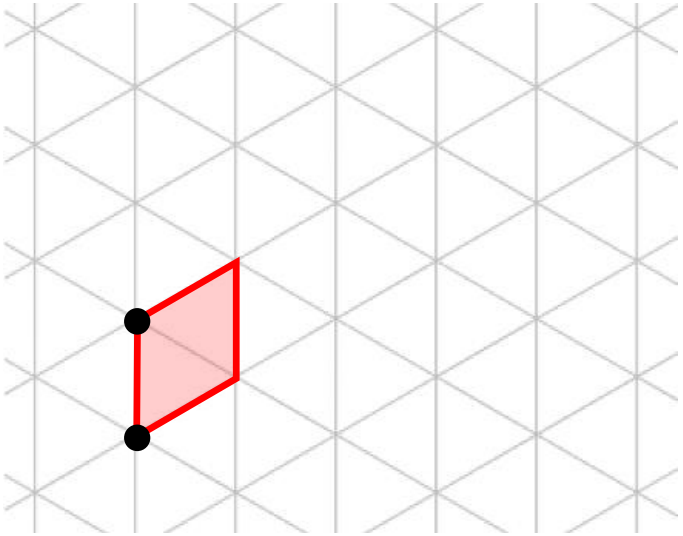


Isometric grid



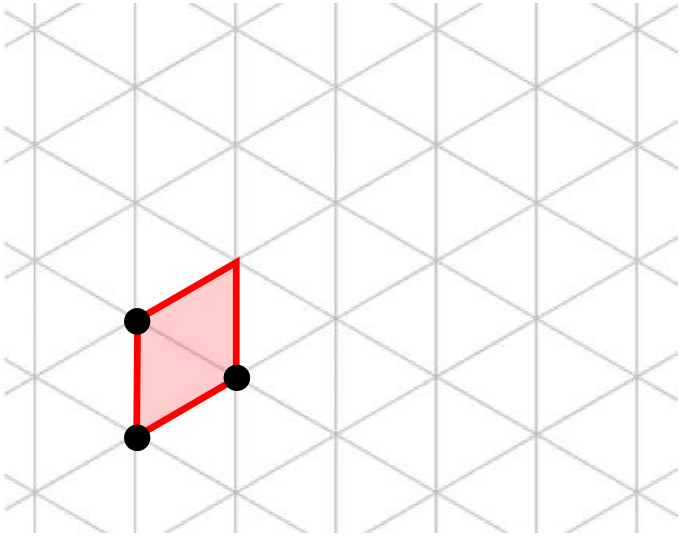
kite = [$(0, 0)$

Isometric grid



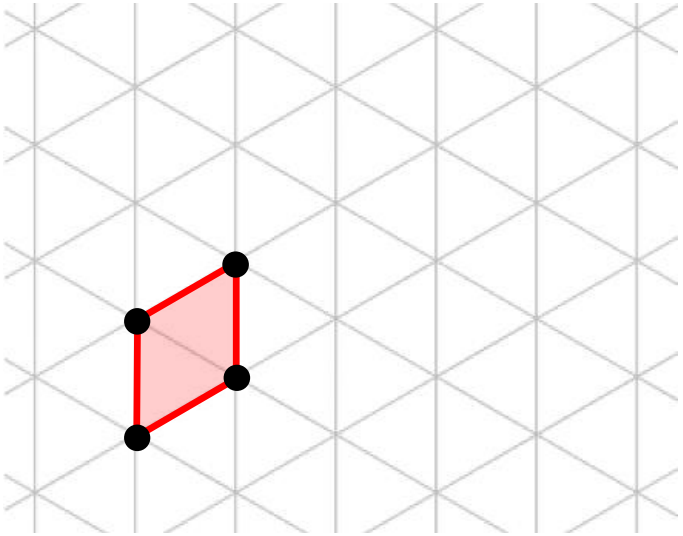
kite = [$(0, 0)$, $(0, -1)$

Isometric grid



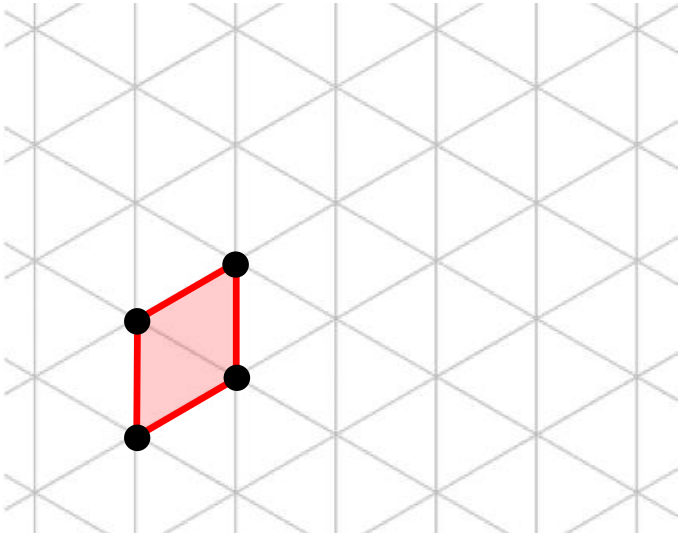
kite = [$(0, 0)$, $(0, -1)$, $\left(\frac{1}{2}\sqrt{3}, -\frac{1}{2}\right)$

Isometric grid



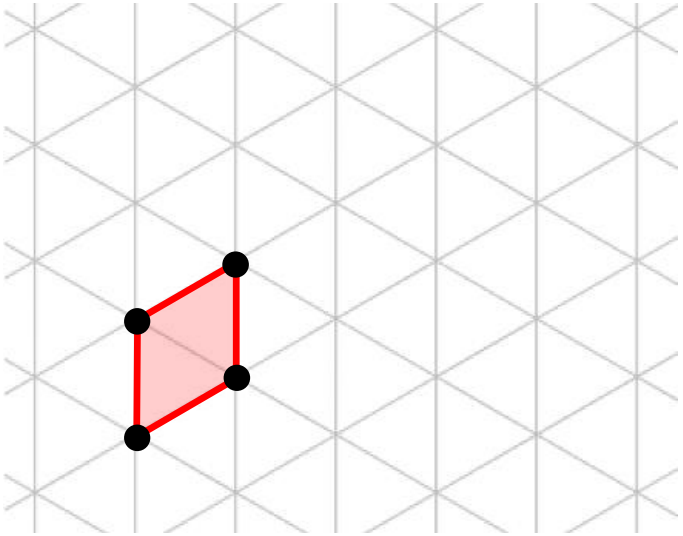
kite = $[\ (0,0),\ (0,-1),\ \left(\frac{1}{2}\sqrt{3},-\frac{1}{2}\right),\ \left(\frac{1}{2}\sqrt{3},\frac{1}{2}\right)$

Isometric grid



kite = [$(0,0)$, $(0,-1)$, $\left(\frac{1}{2}\sqrt{3},-\frac{1}{2}\right)$, $\left(\frac{1}{2}\sqrt{3},\frac{1}{2}\right)$, $(0,0)$]

Isometric grid

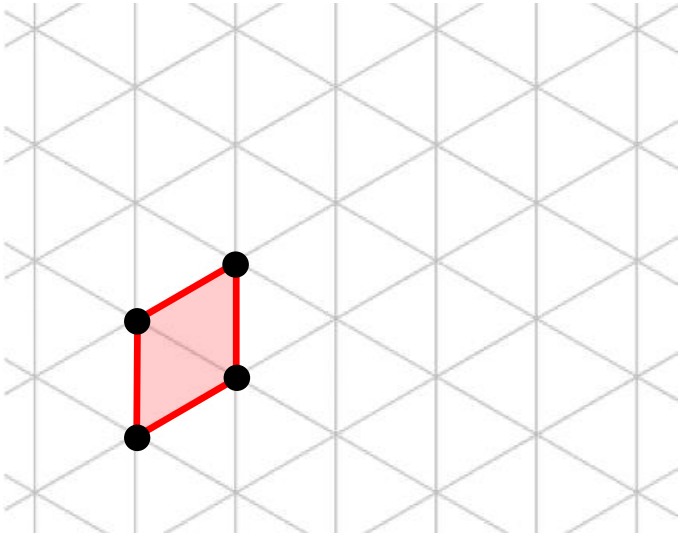


kite = [$(0, 0)$, $(0, -1)$, $\left(1, -\frac{1}{2}\right)$, $\left(1, \frac{1}{2}\right)$, $(0, 0)$]

kite = x_correction(kite)

kite = [$(0, 0)$, $(0, -1)$, $\left(\frac{1}{2}\sqrt{3}, -\frac{1}{2}\right)$, $\left(\frac{1}{2}\sqrt{3}, \frac{1}{2}\right)$, $(0, 0)$]

Isometric grid

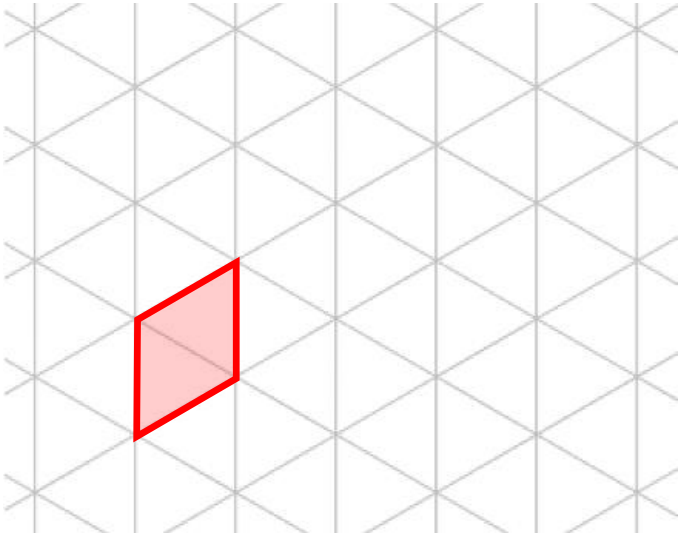


$$\text{kite} = [\,(0,0),\ (0,-1),\ \left(1,\ -\tfrac{1}{2}\right),\ \left(1,\ \tfrac{1}{2}\right),\ (0,0)\,]$$

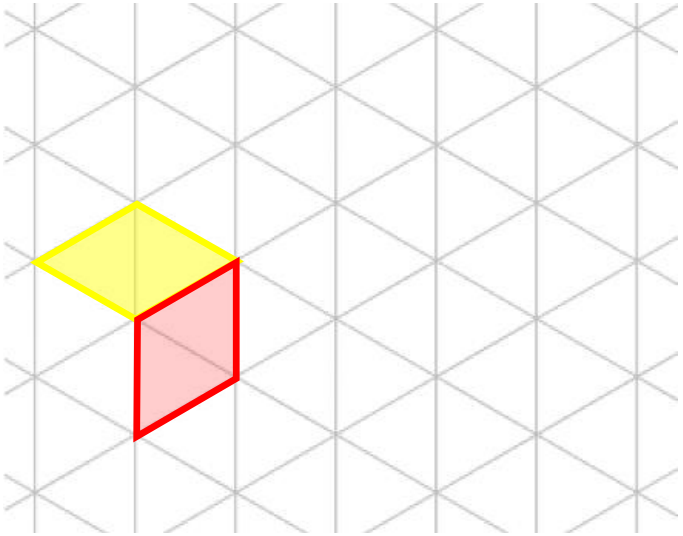
kite = x_correction(kite)

$$\text{kite} = [\,(0,0),\ (0,-1),\ \left(\tfrac{1}{2}\sqrt{3},-\tfrac{1}{2}\right),\ \left(\tfrac{1}{2}\sqrt{3},\tfrac{1}{2}\right),\ (0,0)\,]$$

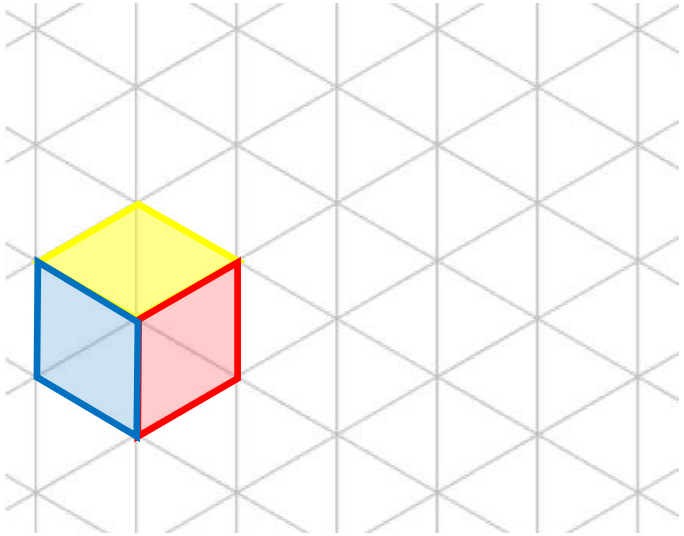
Isometric grid



Isometric grid

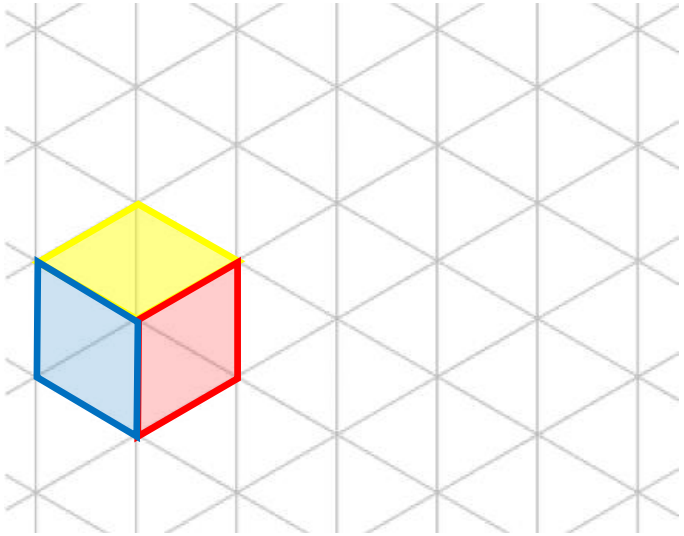


Isometric grid

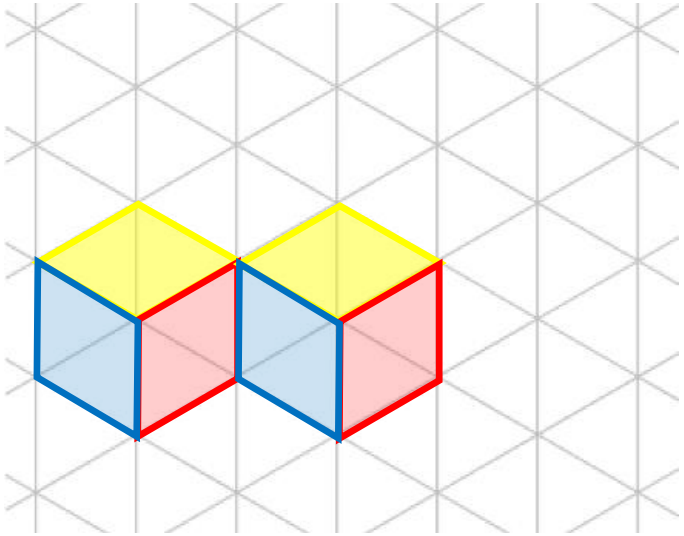


```
kite = [(0,0), (0,−1), (1,−0.5), (1,0.5), (0,0)]
kite = x_correction(kite)

colors = [(255,0,0), (255,255,0), (0,0,255)]

def hexagon(x,y):
    for i in range(3):
        kyte = rotate(kite, 120*i)
        set_color(colors[i])
        fill(x,y,kyte)
```

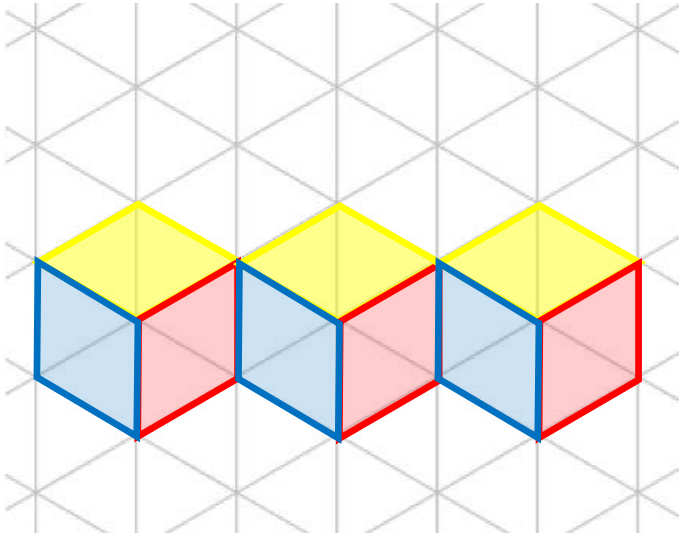
Isometric grid



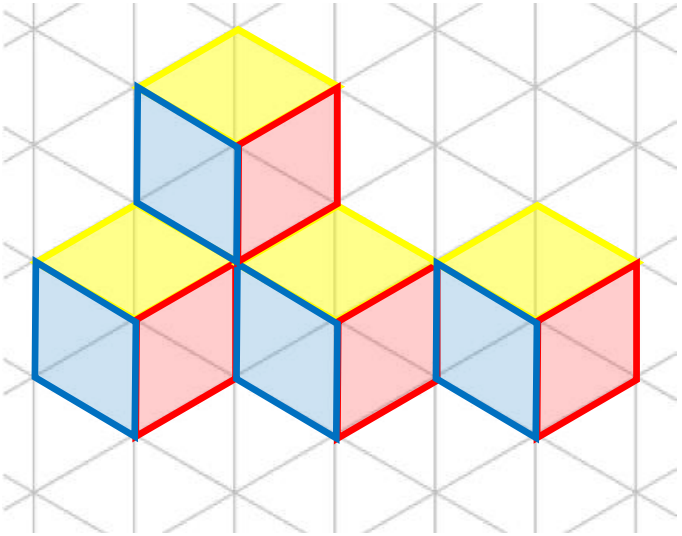
Isometric grid



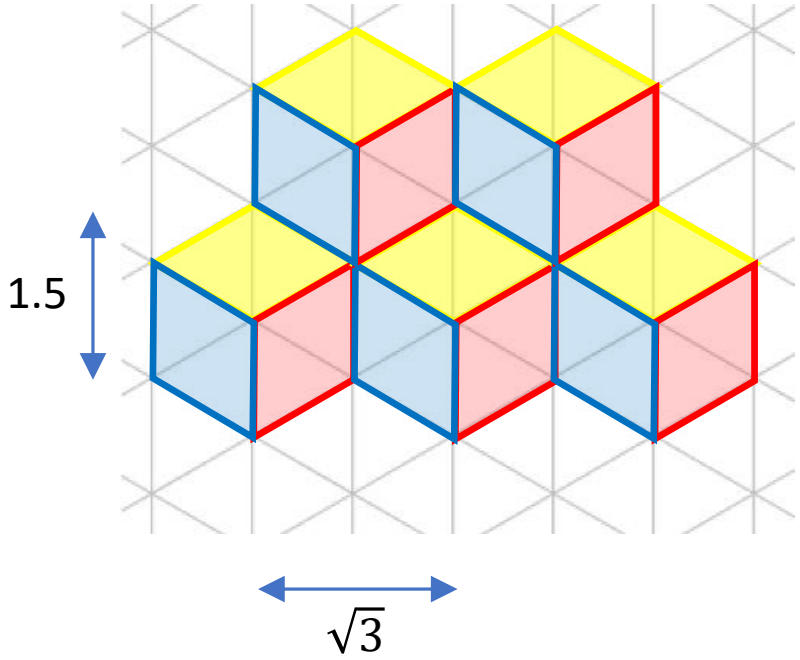
Isometric grid



Isometric grid



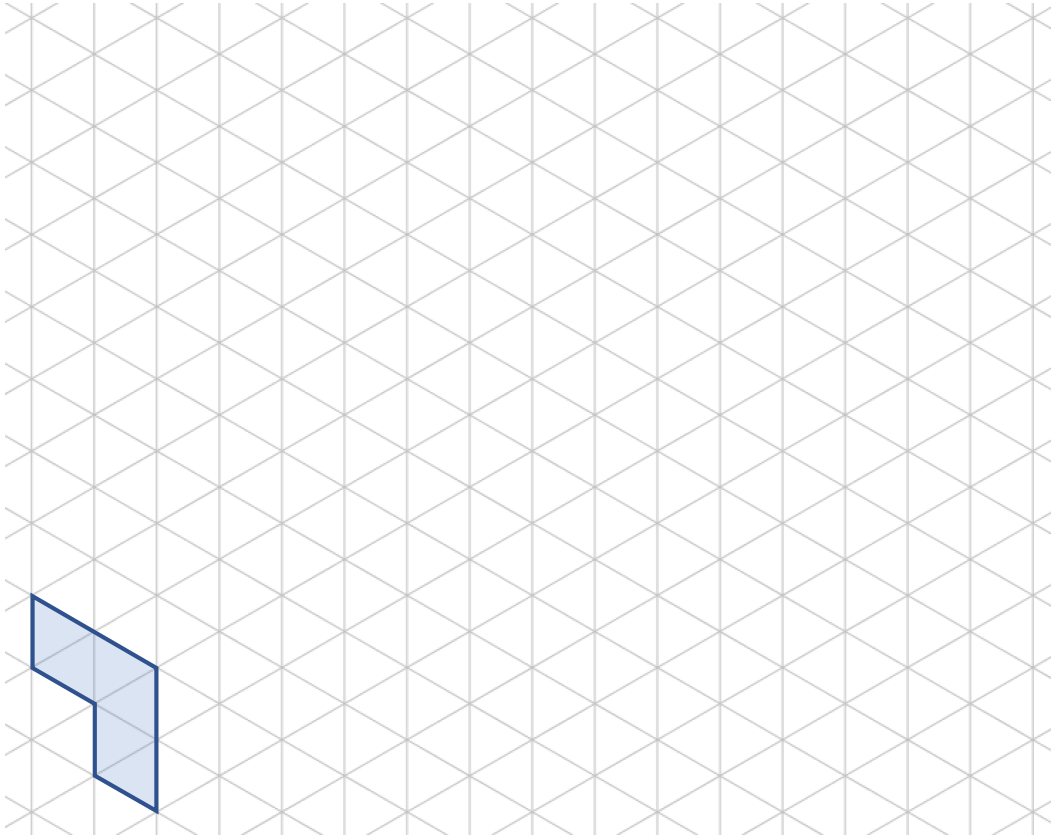
Isometric grid



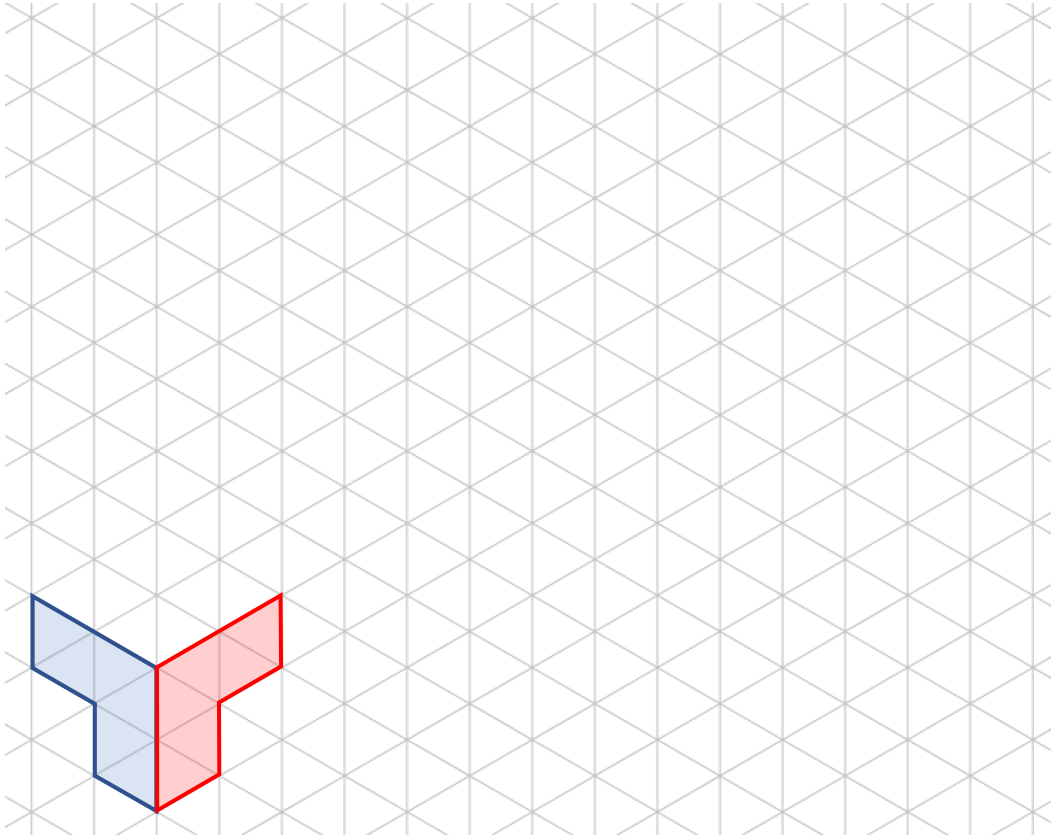

```
for i  in range(−2,3):
    for j in range(−3,4):
        x = j*sqrt(3)*a
        if i%2 == 0:
            x = x+0.5*sqrt(3)*a
        y = i*1.5*a
        hexagon(x,y)
```

T-shapes

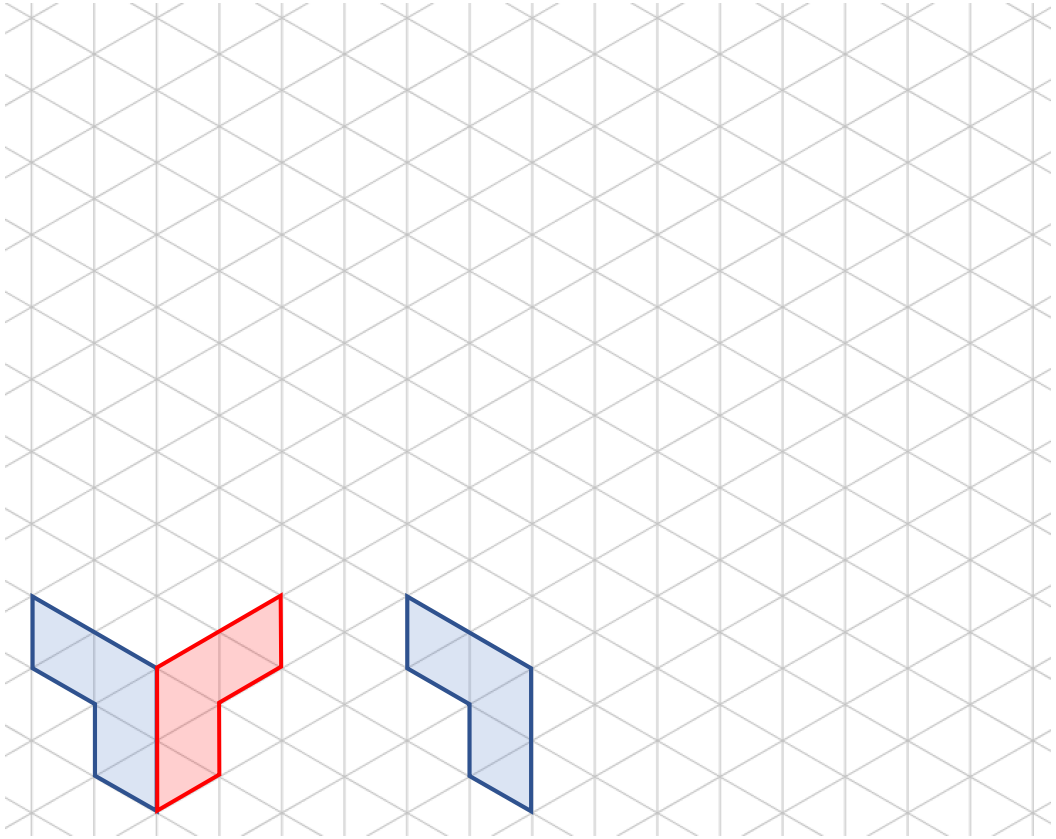
T-shapes



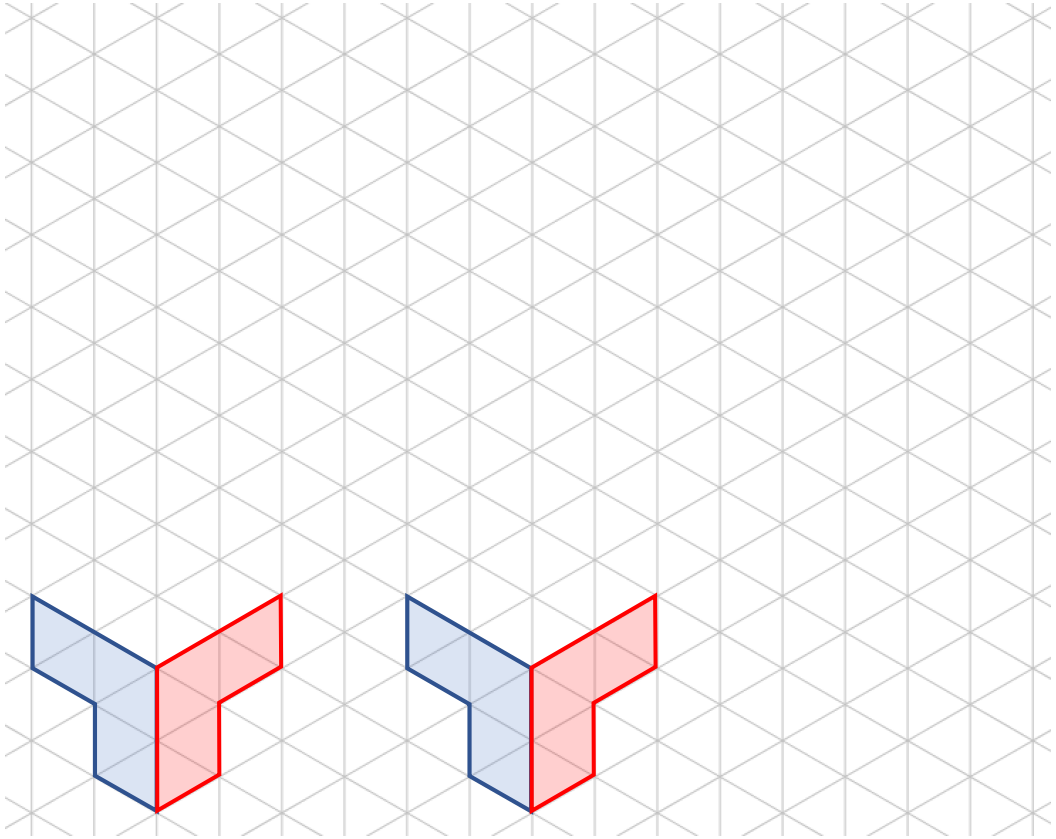
T-shapes



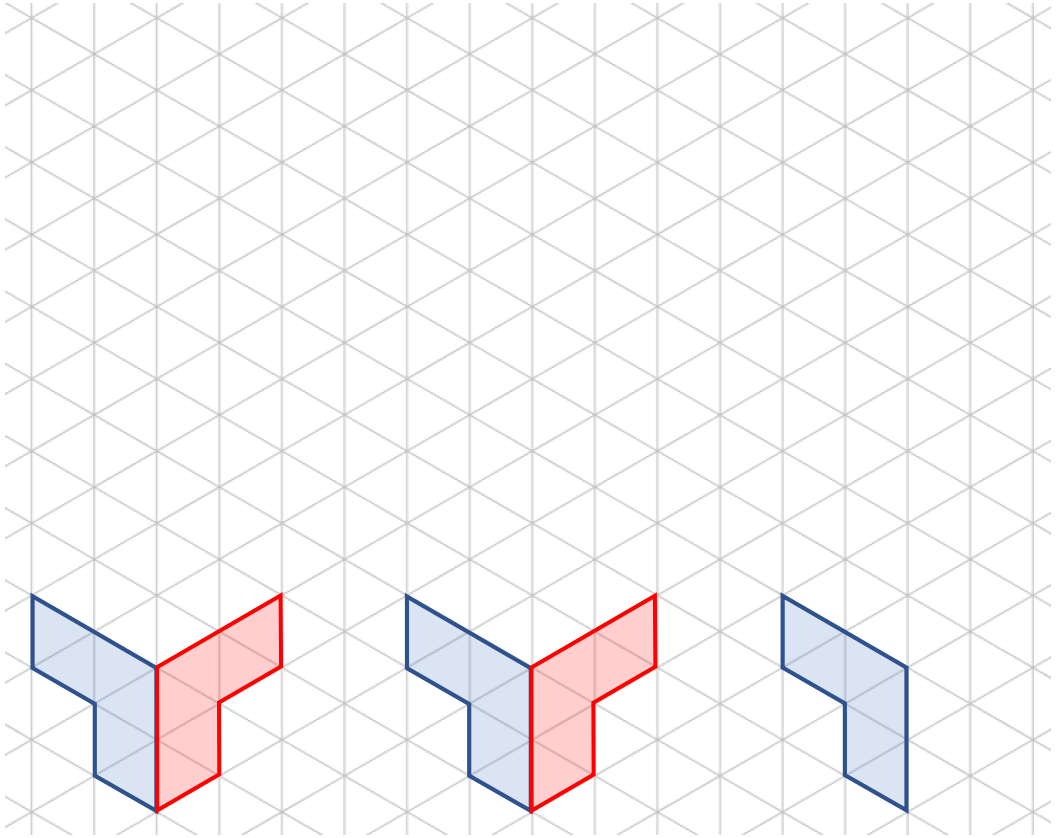
T-shapes



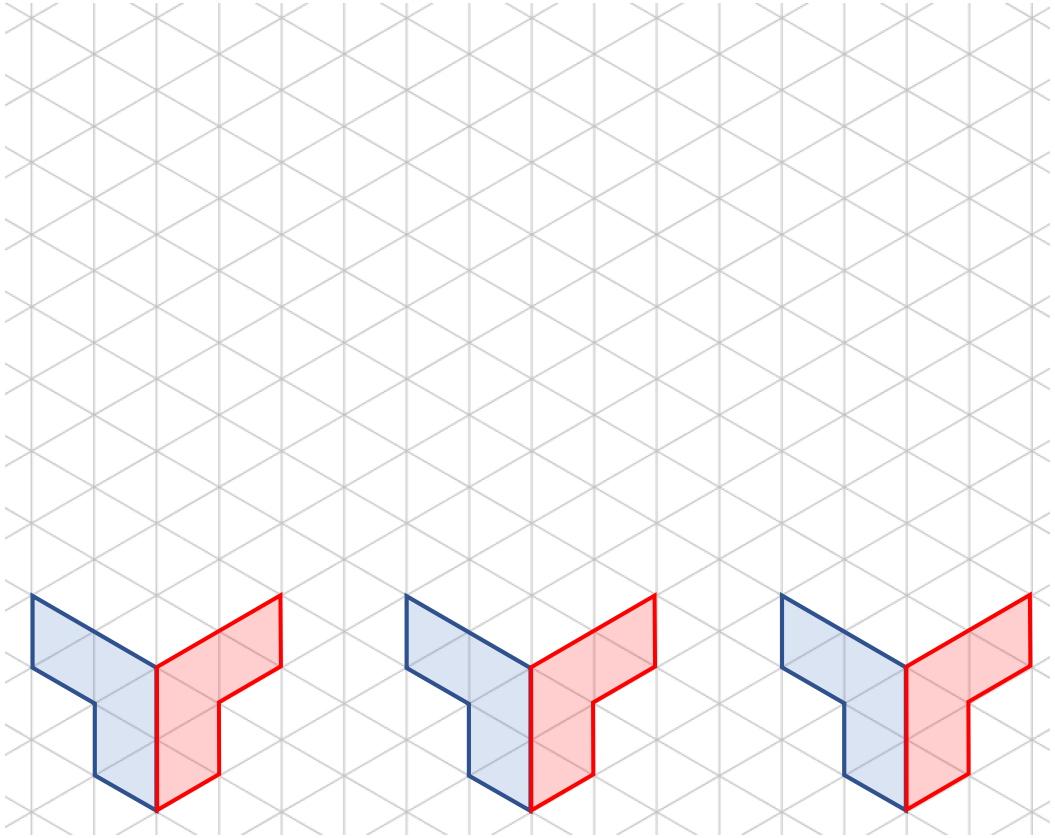
T-shapes



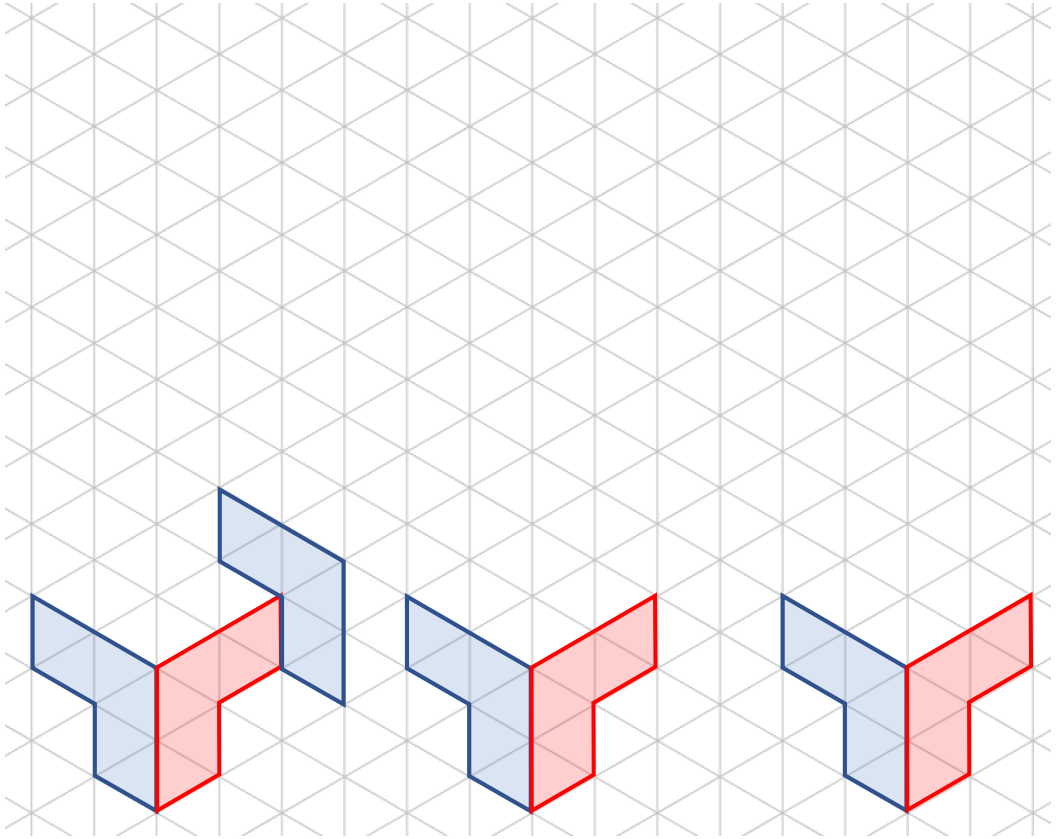
T-shapes



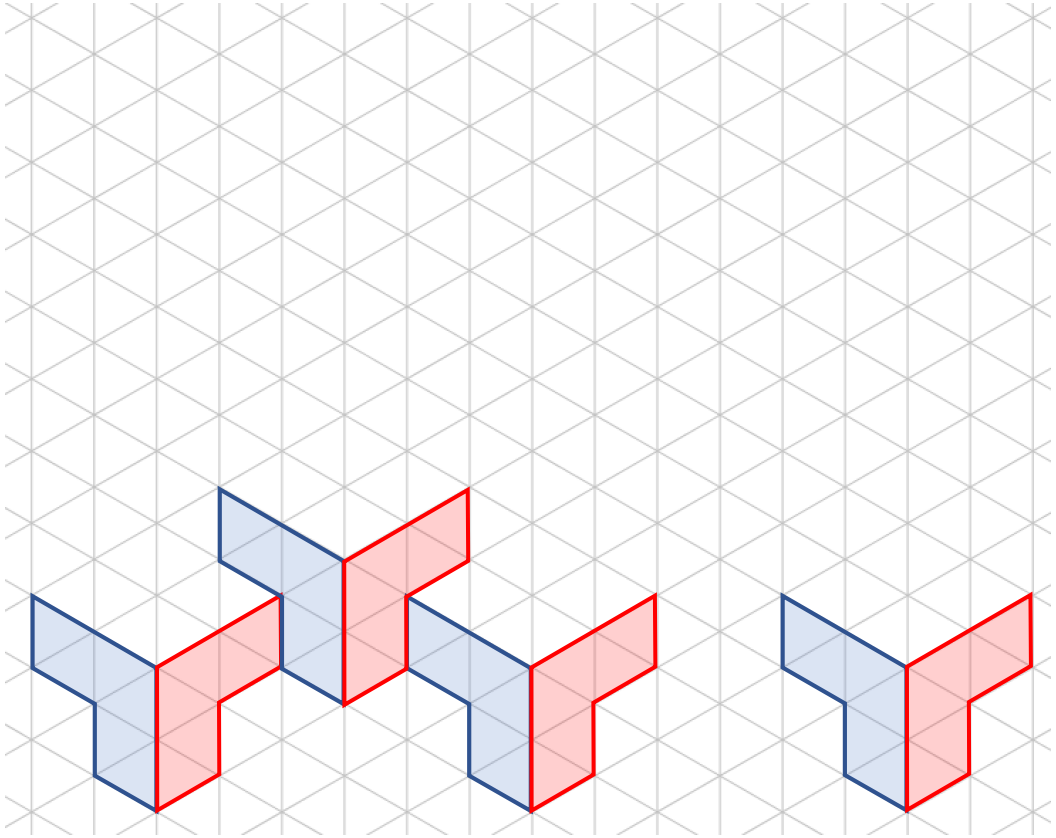
T-shapes



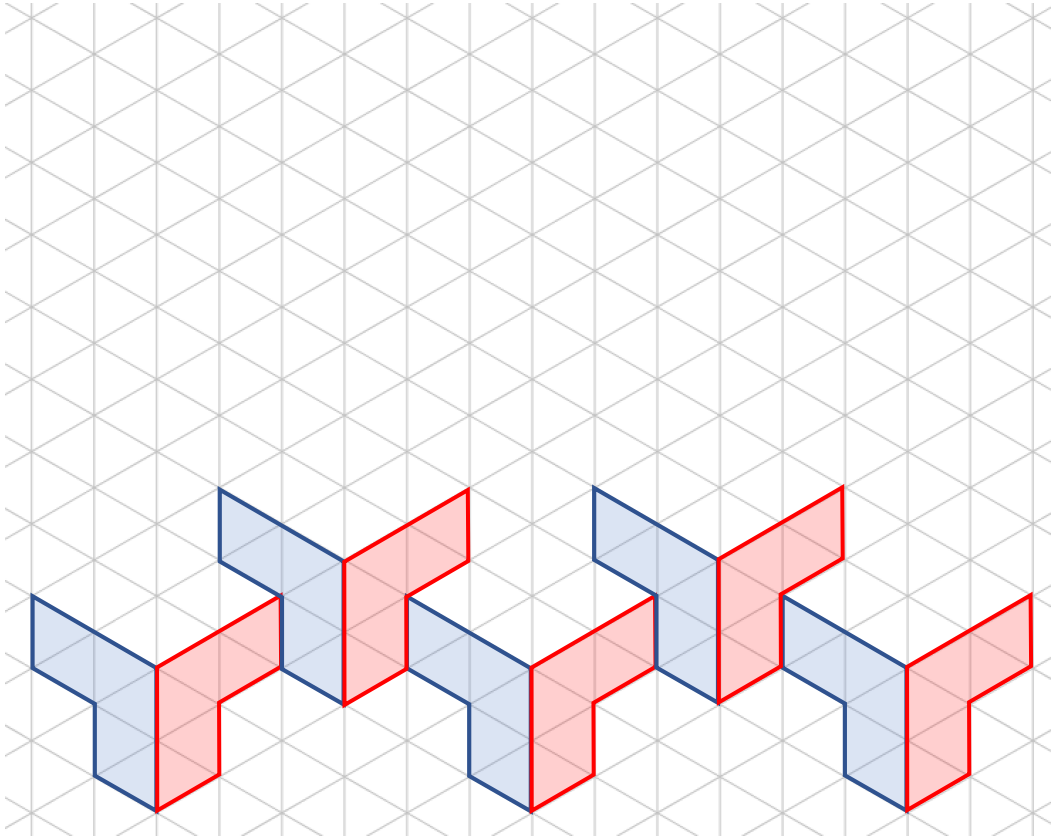
T-shapes



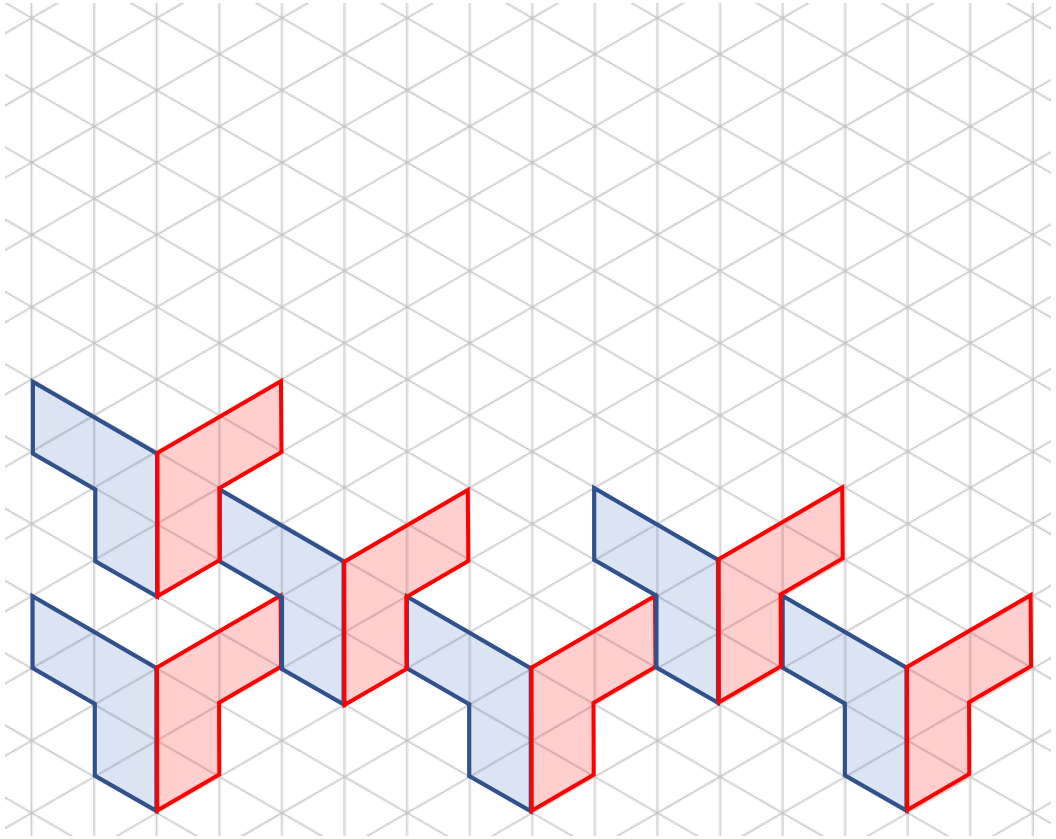
T-shapes



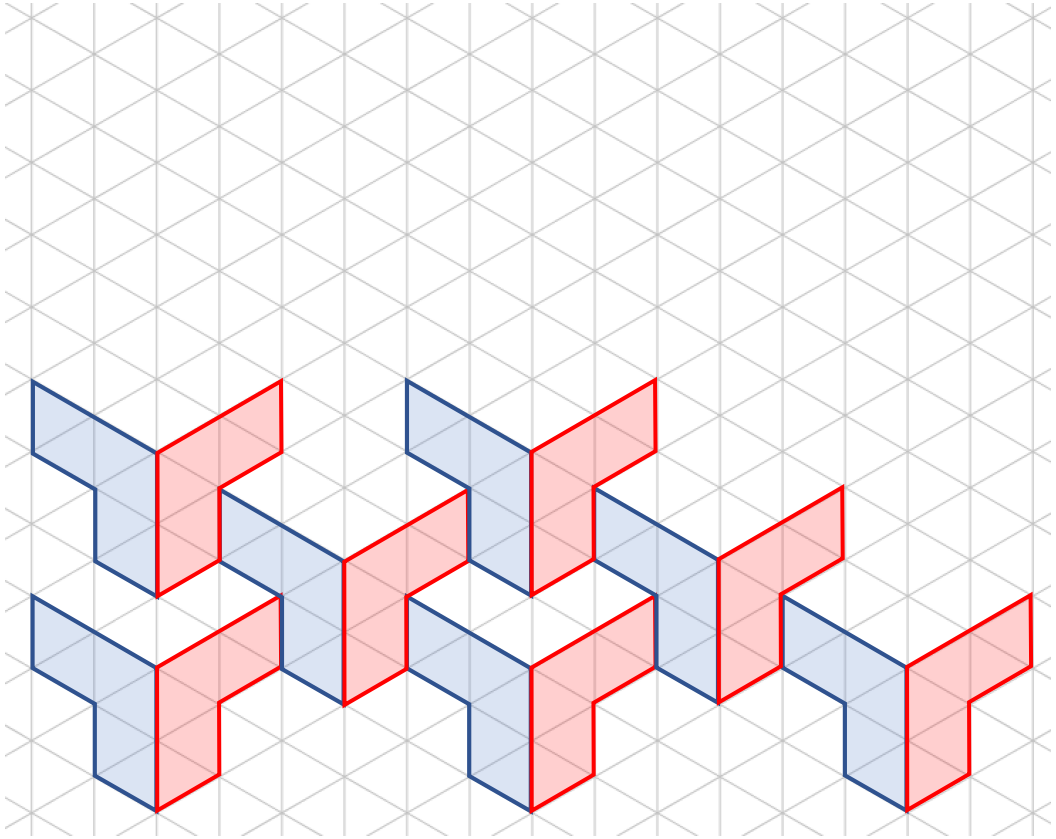
T-shapes



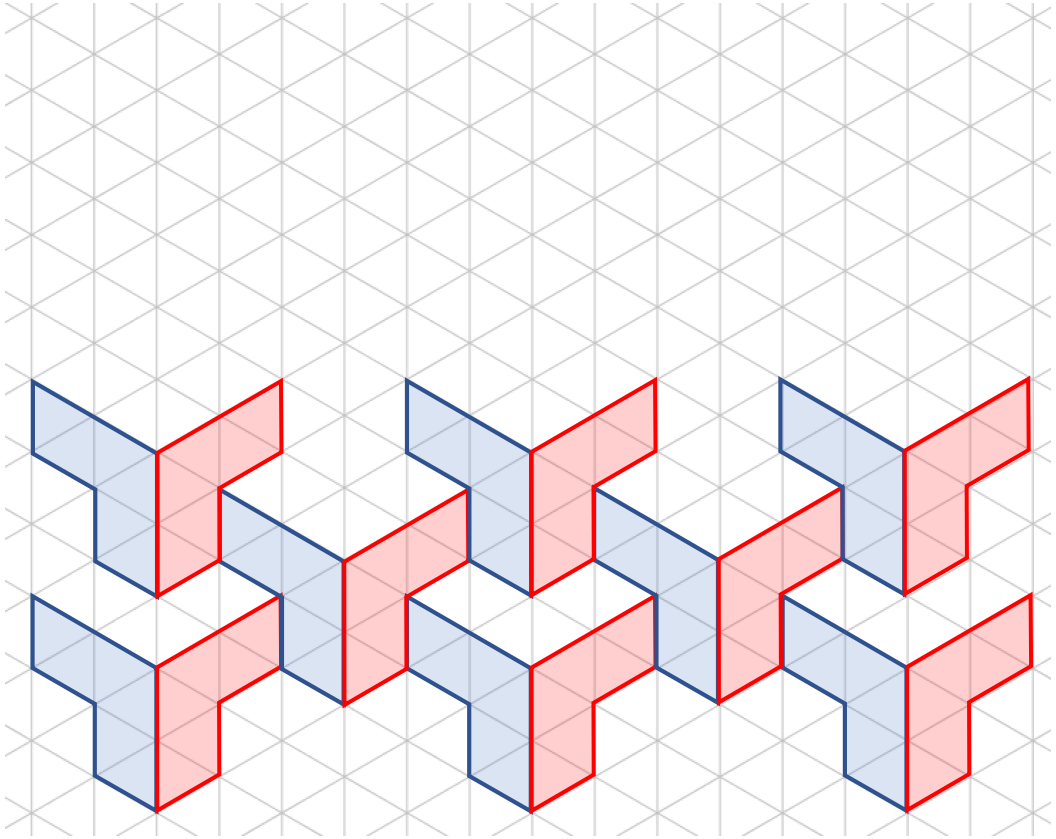
T-shapes



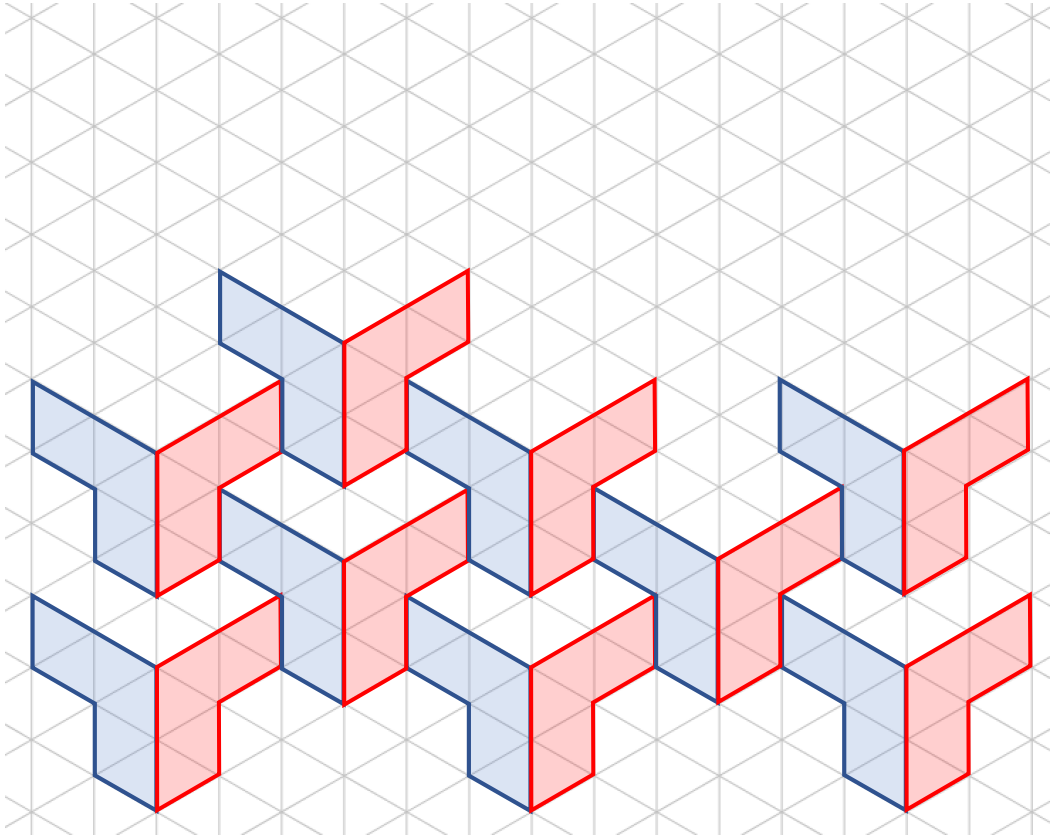
T-shapes



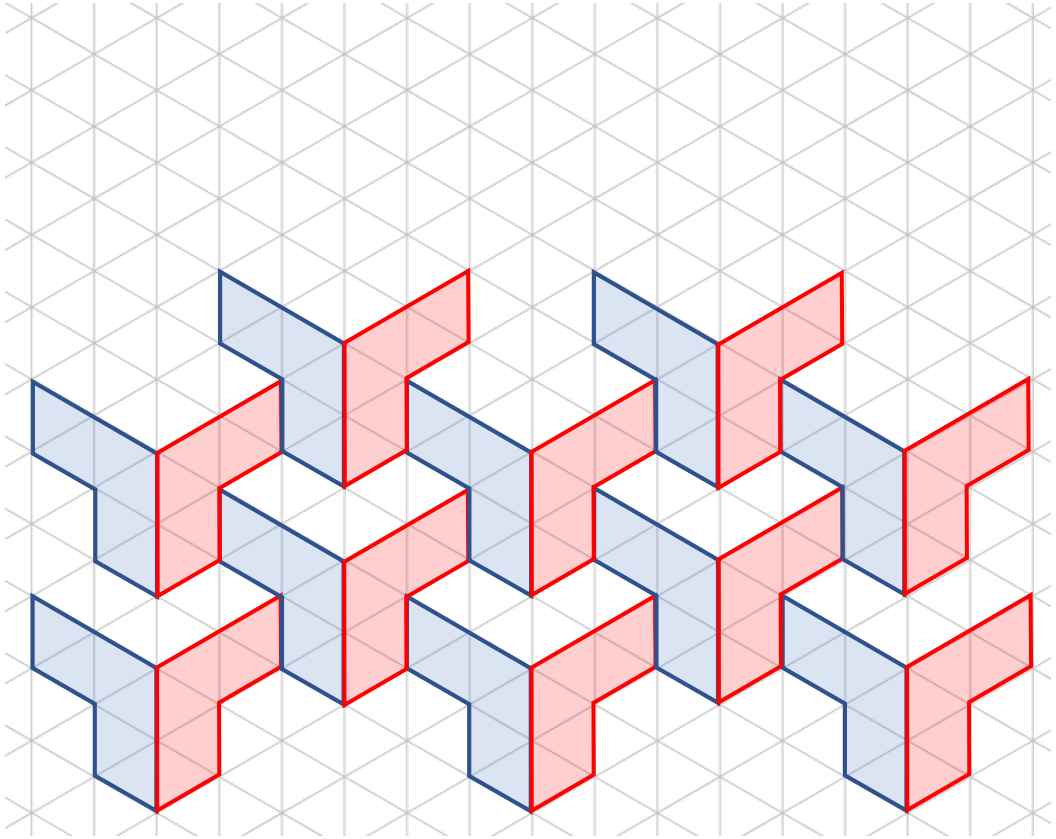
T-shapes



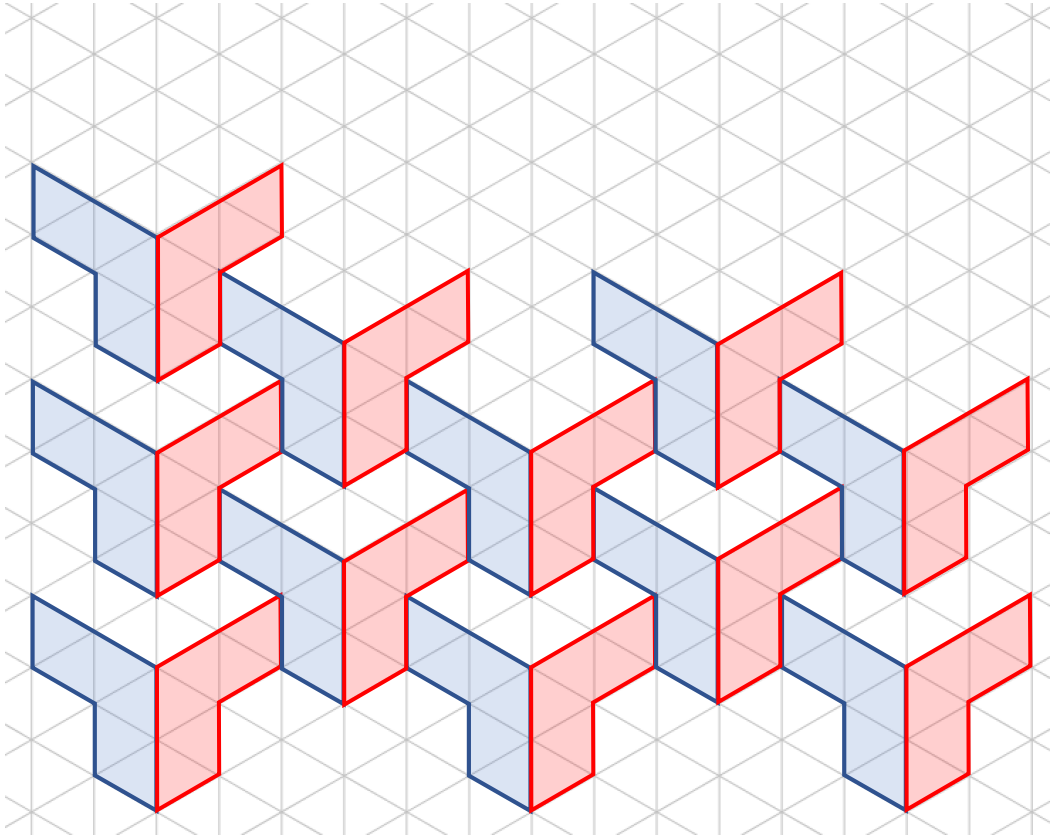
T-shapes



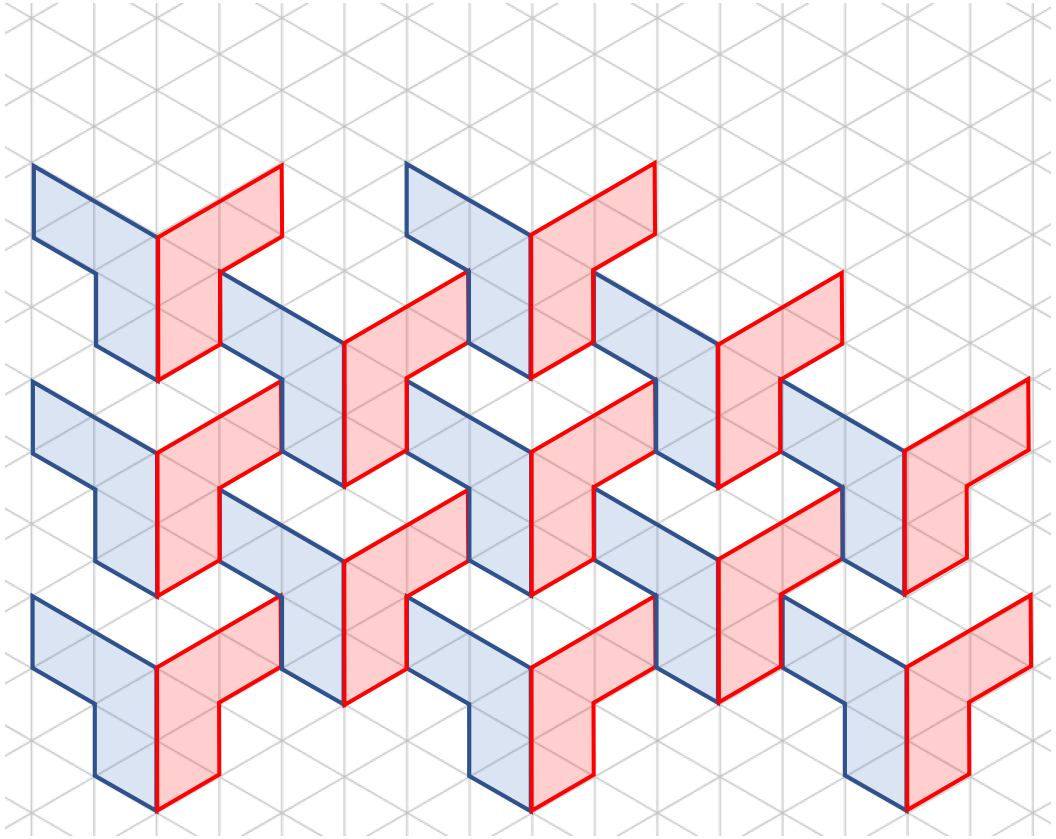
T-shapes



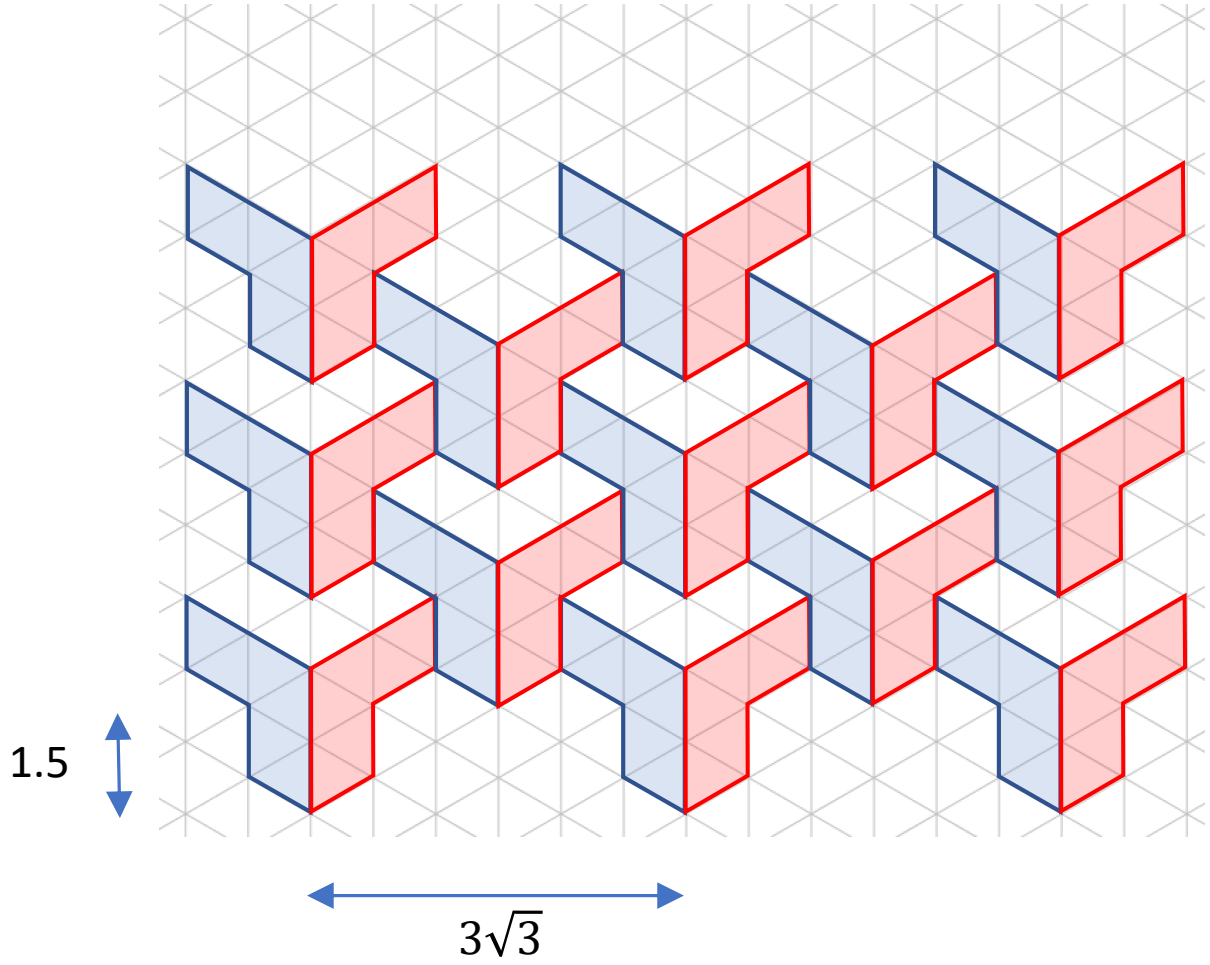
T-shapes



T-shapes



T-shapes



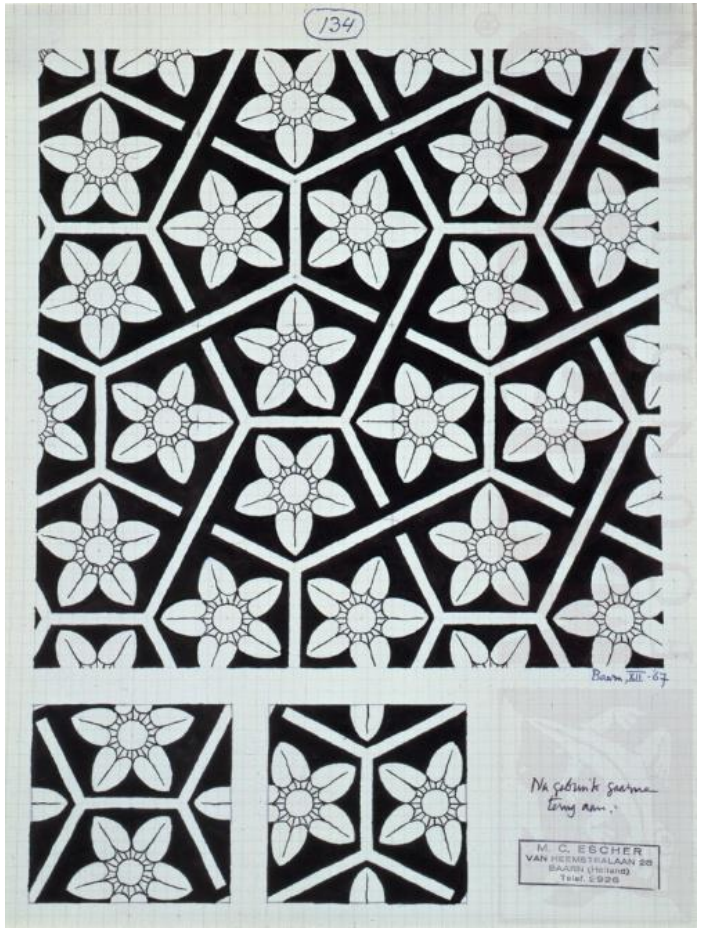
Tessellations

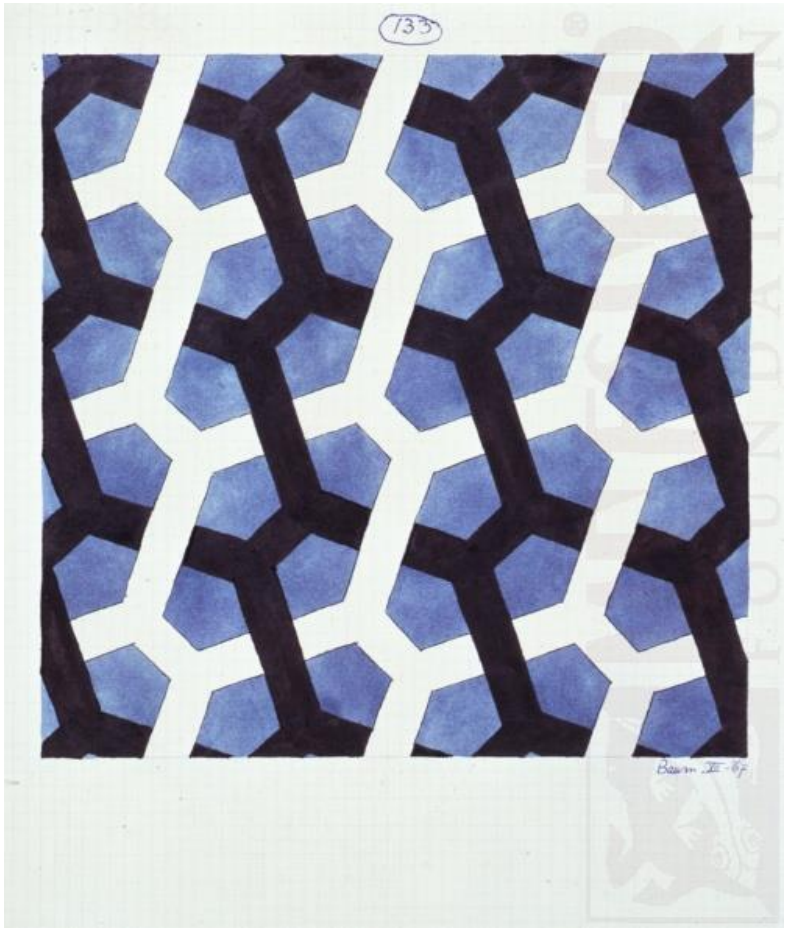
Tessellations

- Cairo
- Fish
- Circular Fish
- Penrose
- End

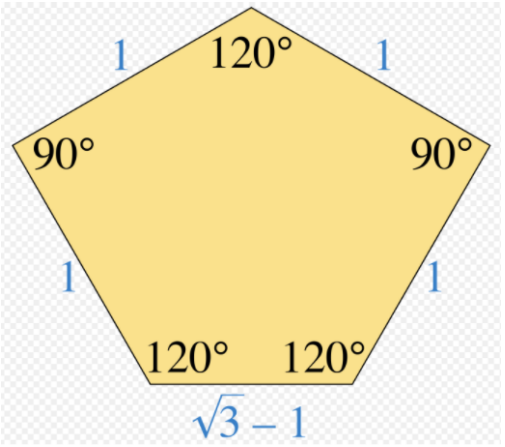
Cairo

Cairo

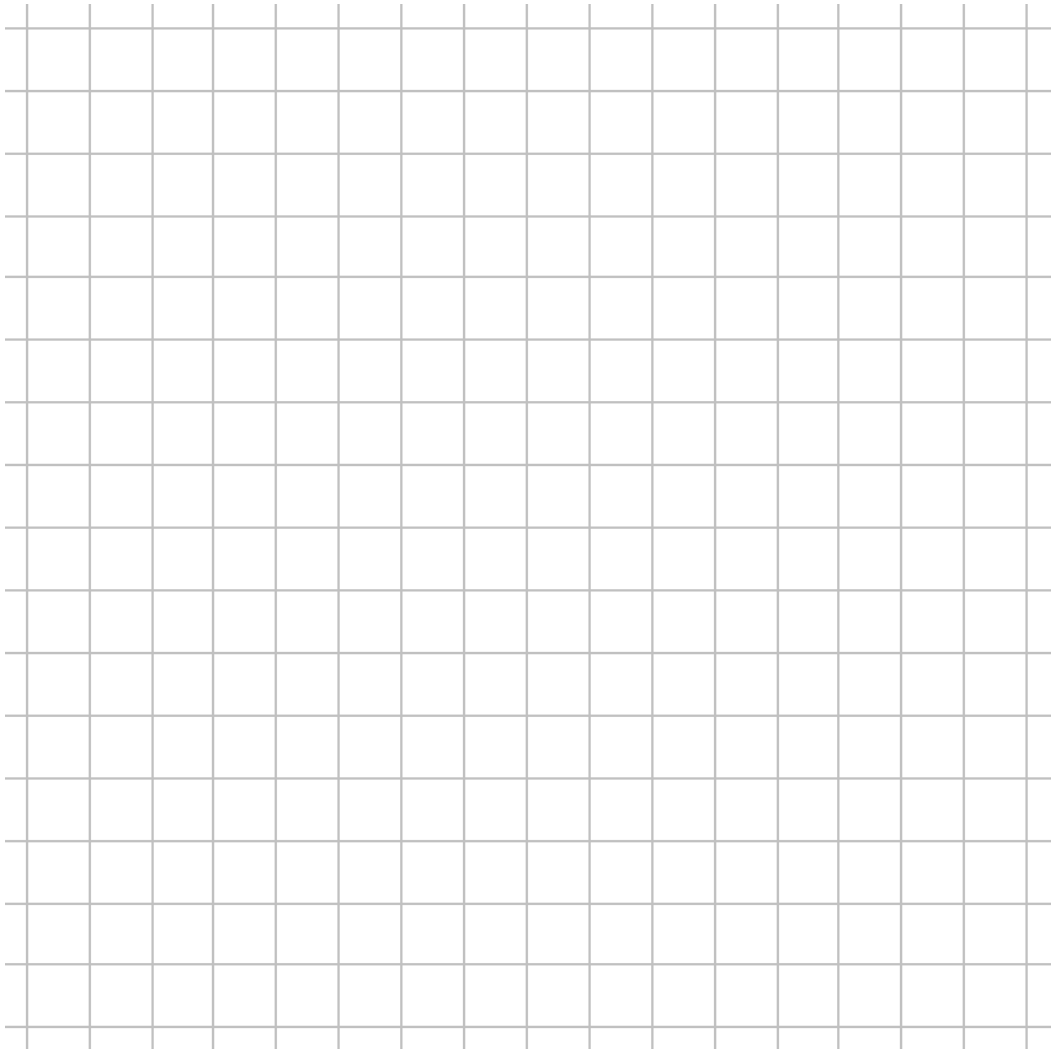




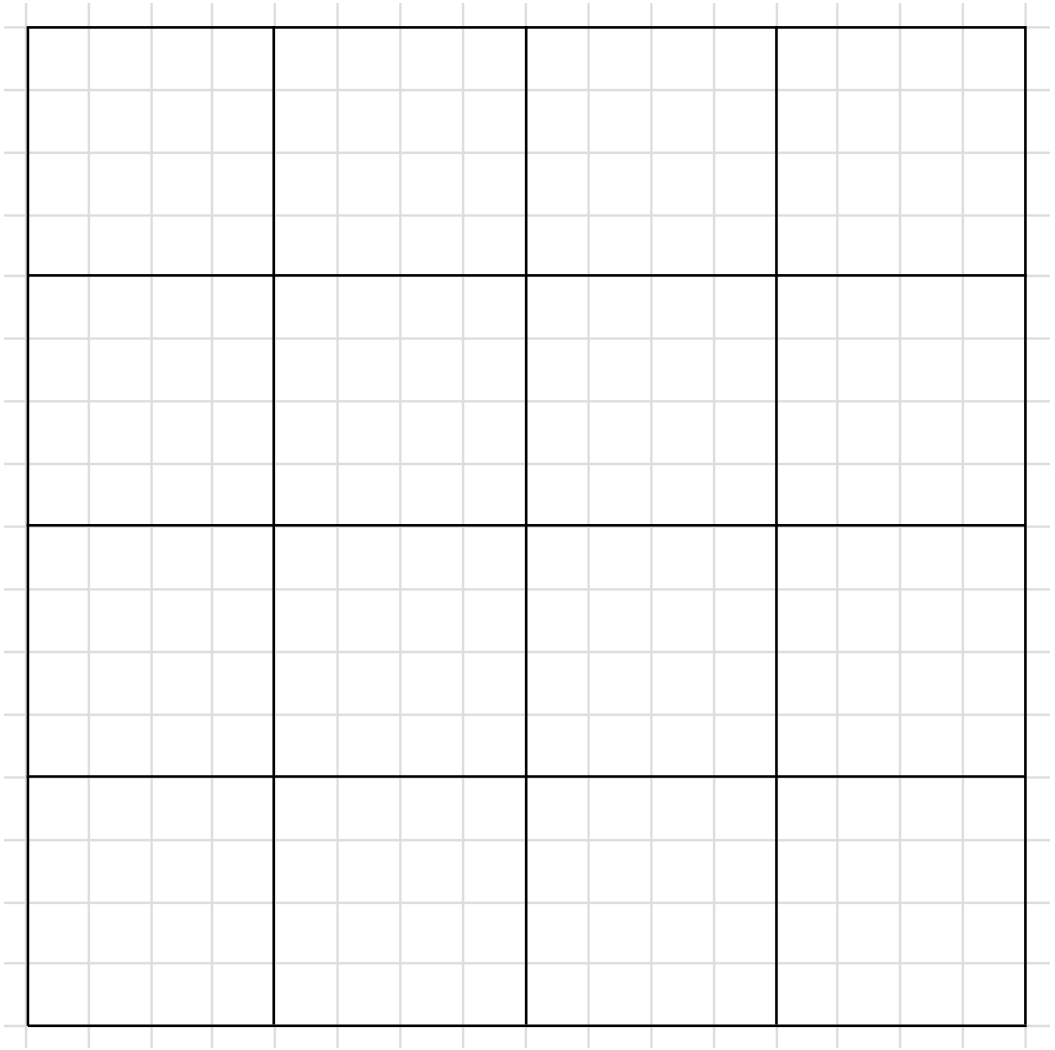
Cairo



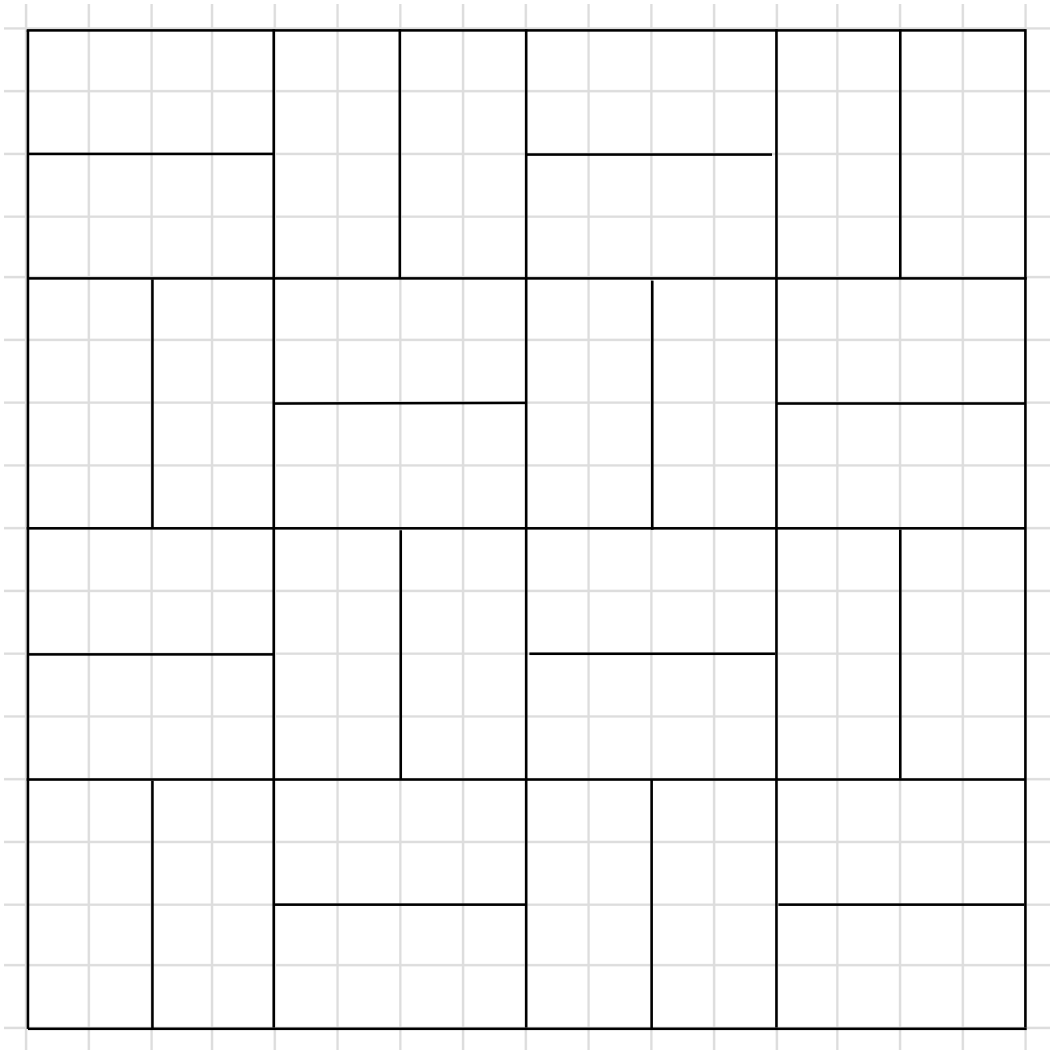
Cairo



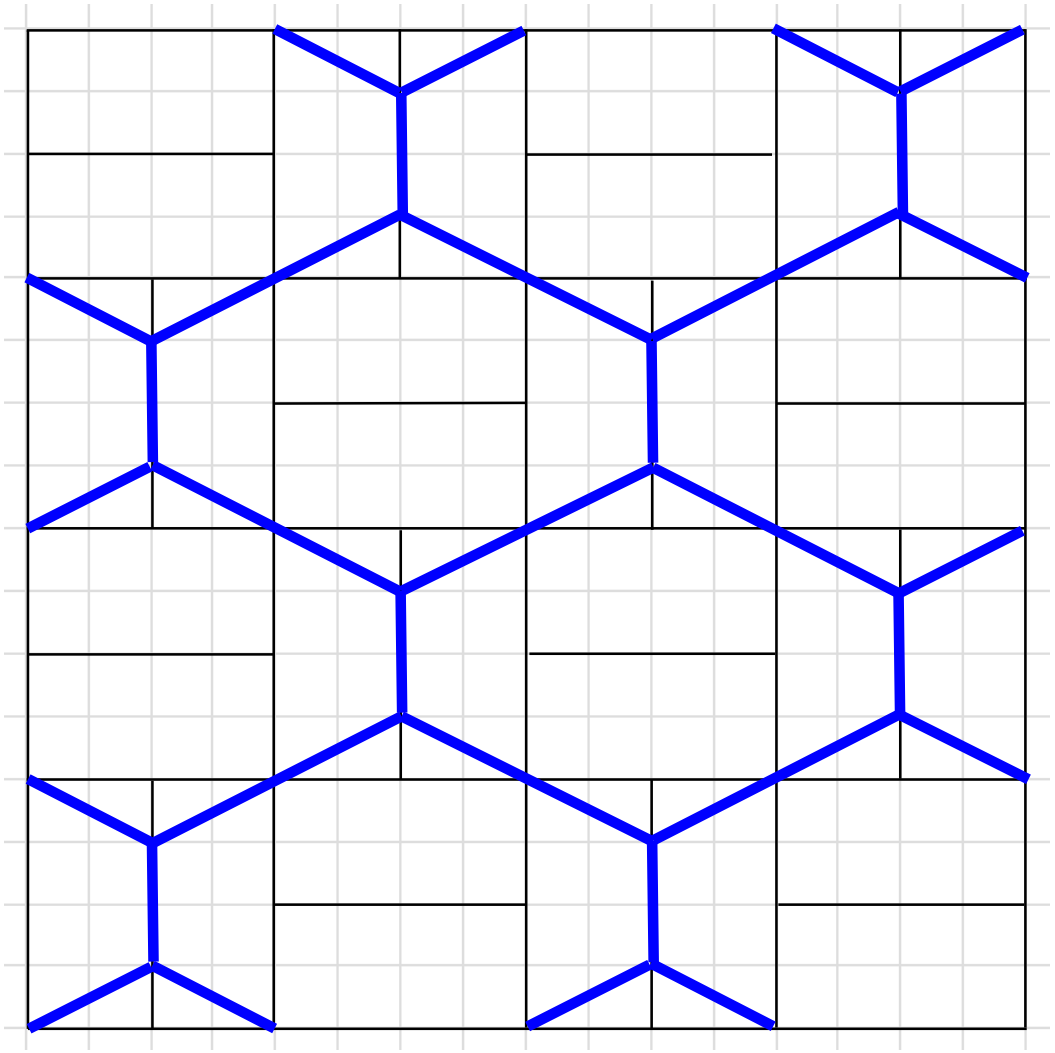
Cairo



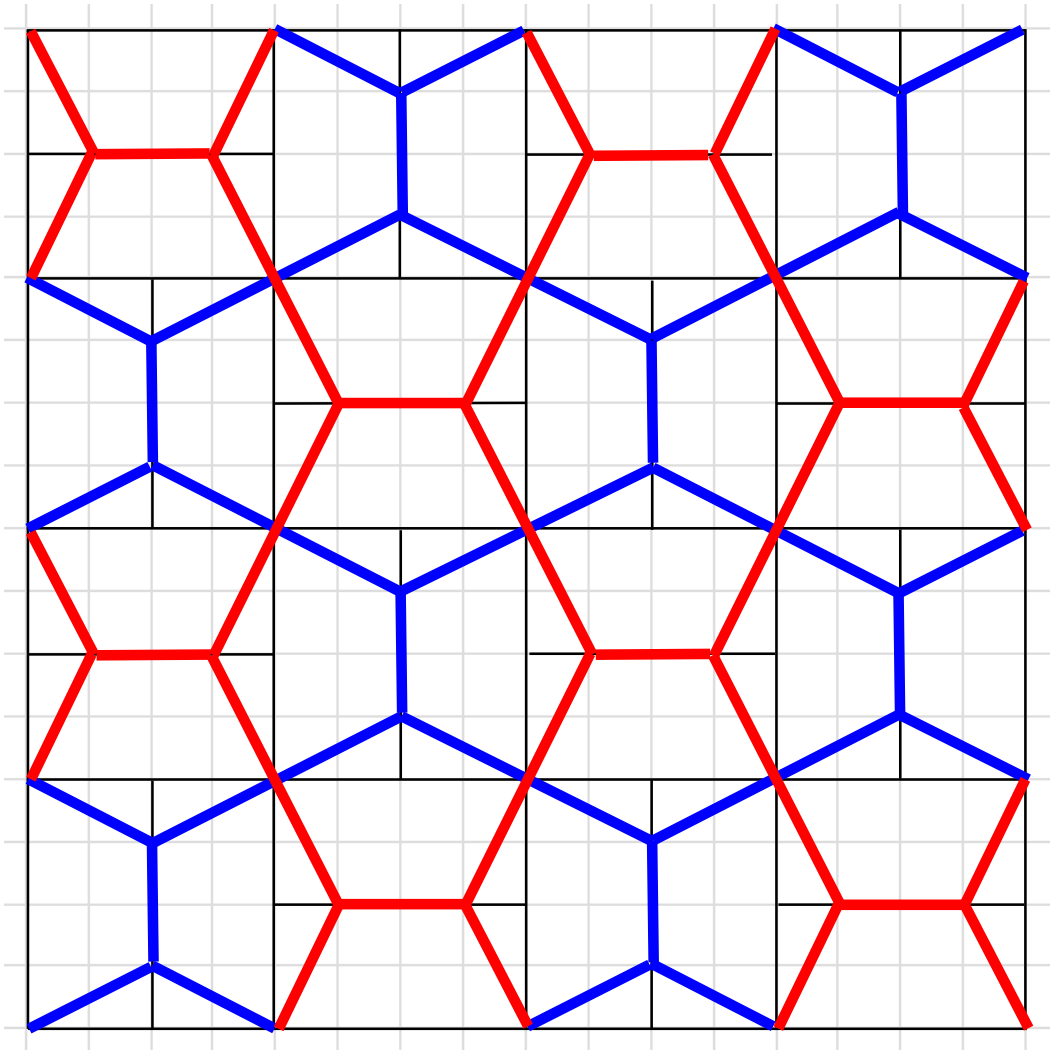
Cairo



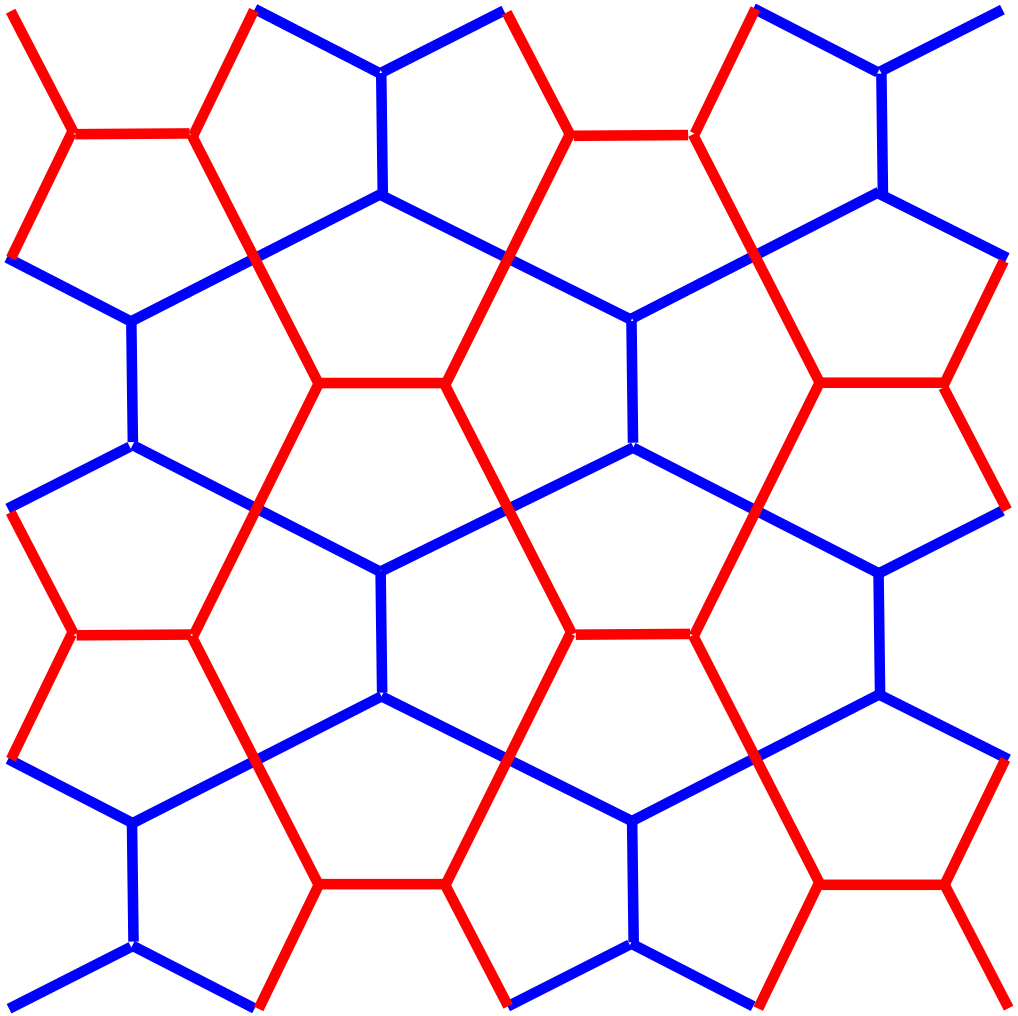
Cairo



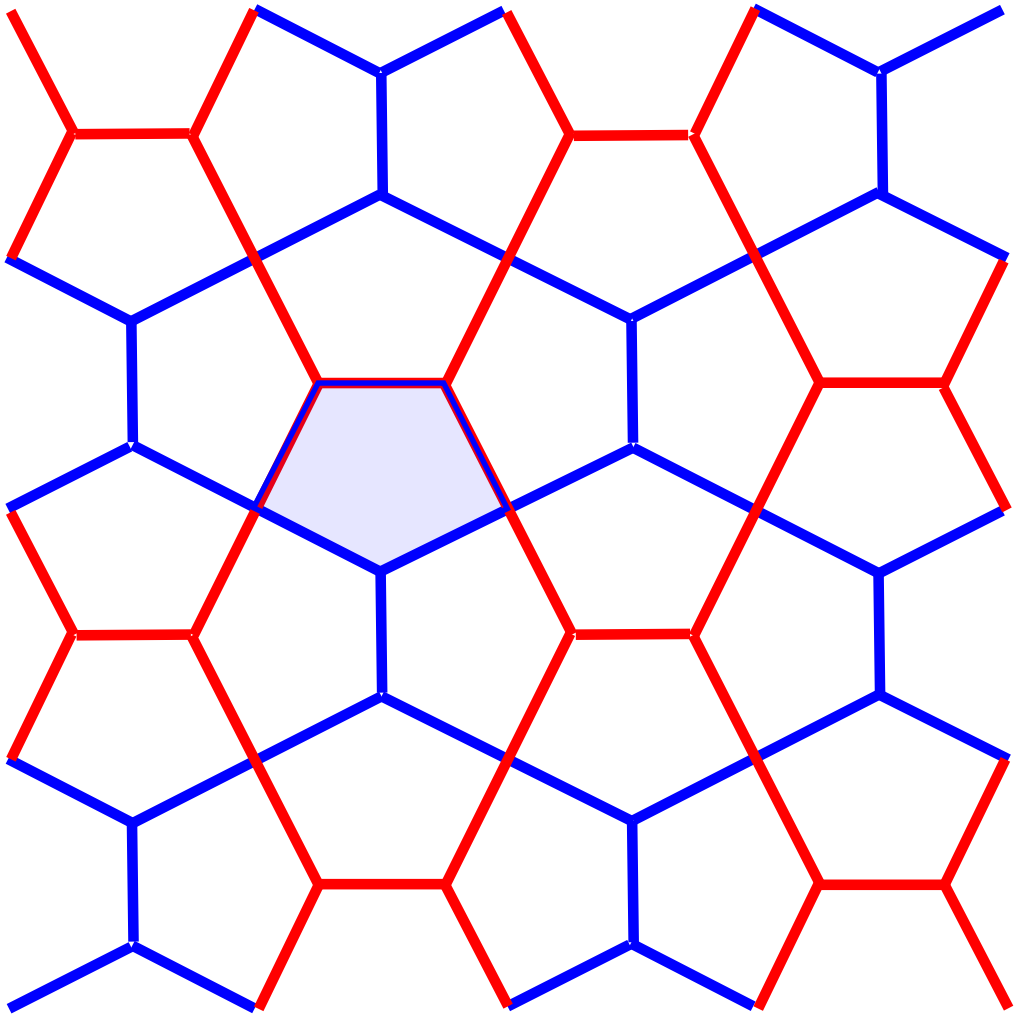
Cairo



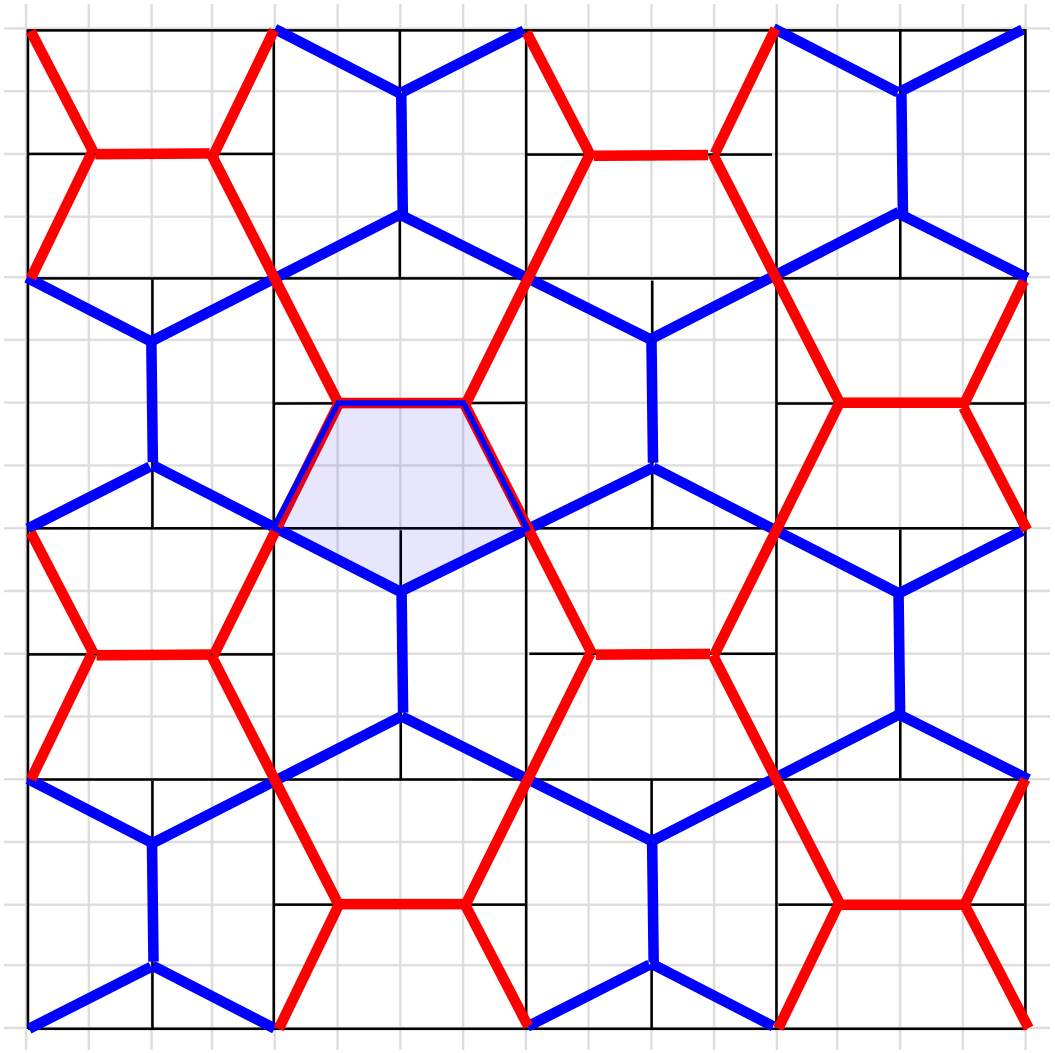
Cairo



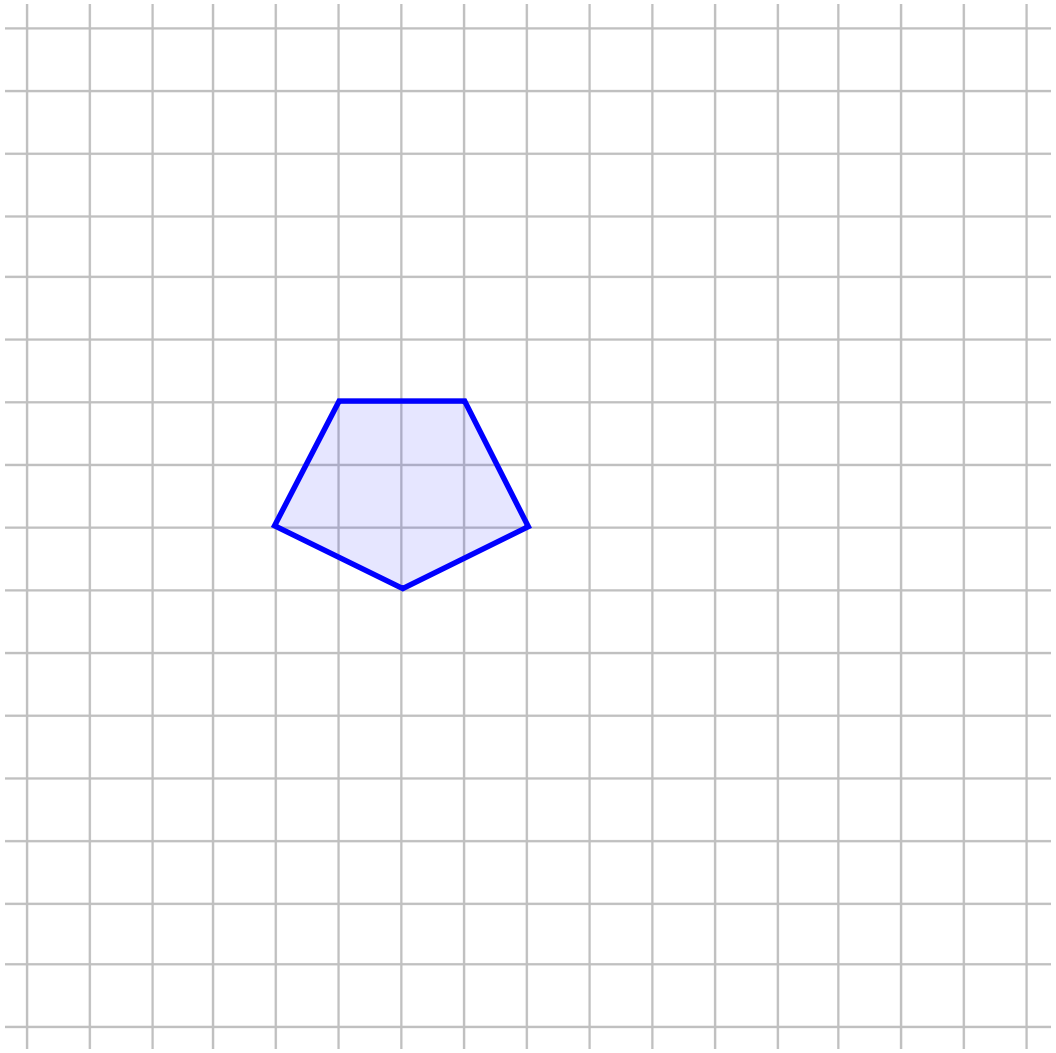
Cairo



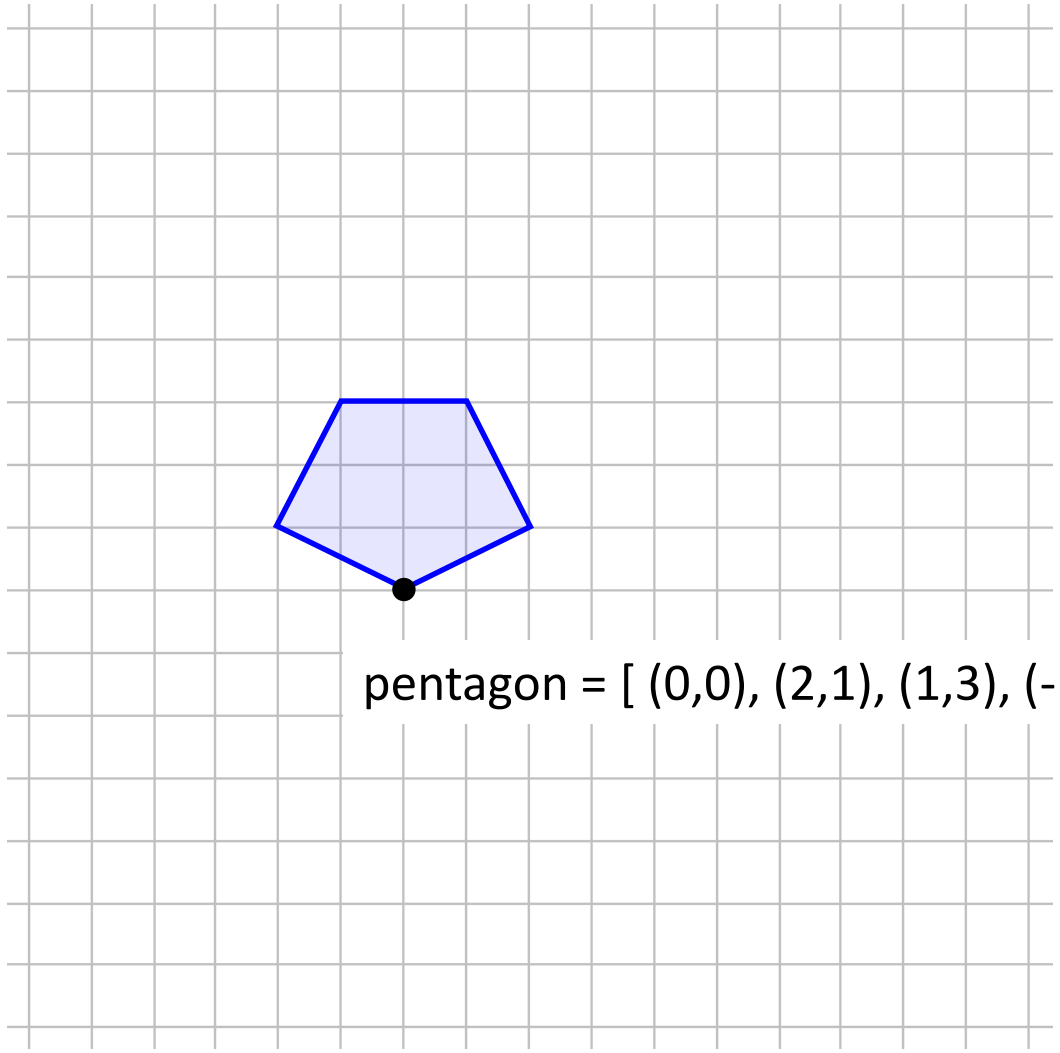
Cairo



Cairo

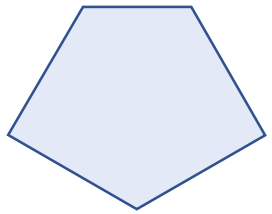


Cairo

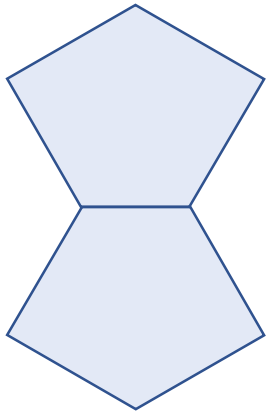


pentagon = [(0,0), (2,1), (1,3), (-1,3), (-2,1), (0,0)]

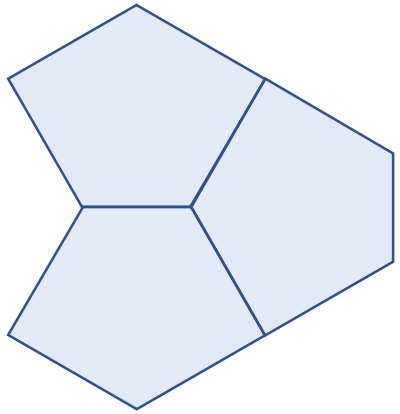
Cairo



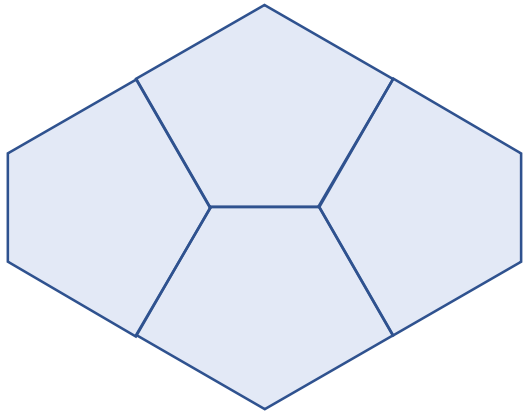
Cairo



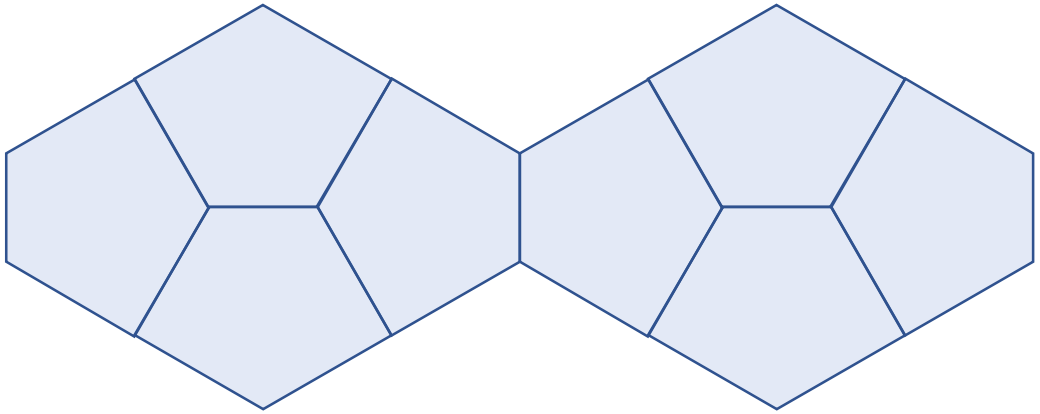
Cairo



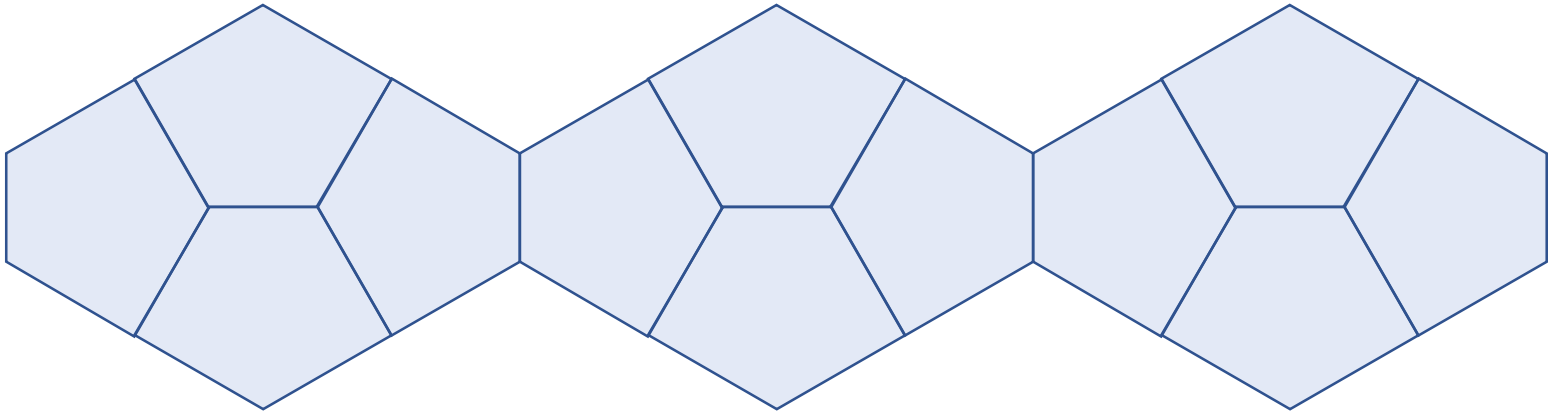
Cairo



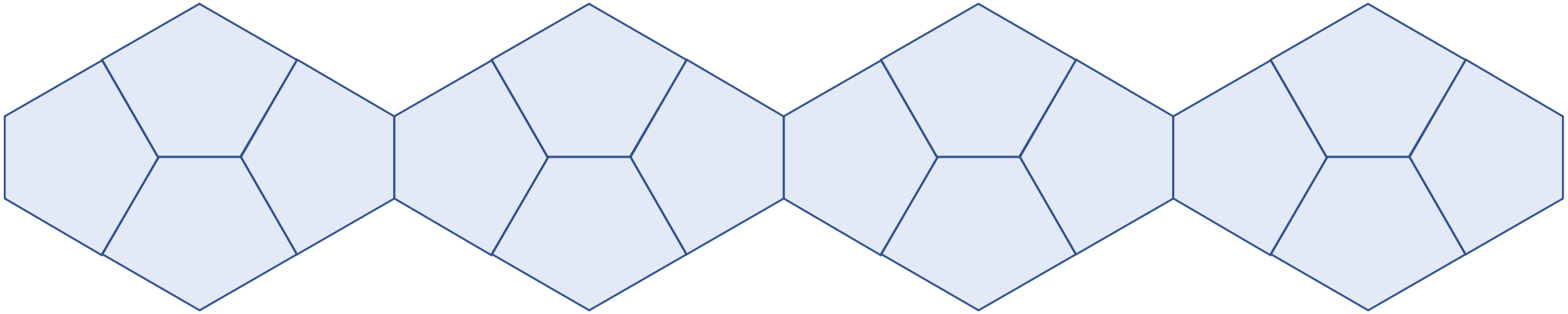
Cairo



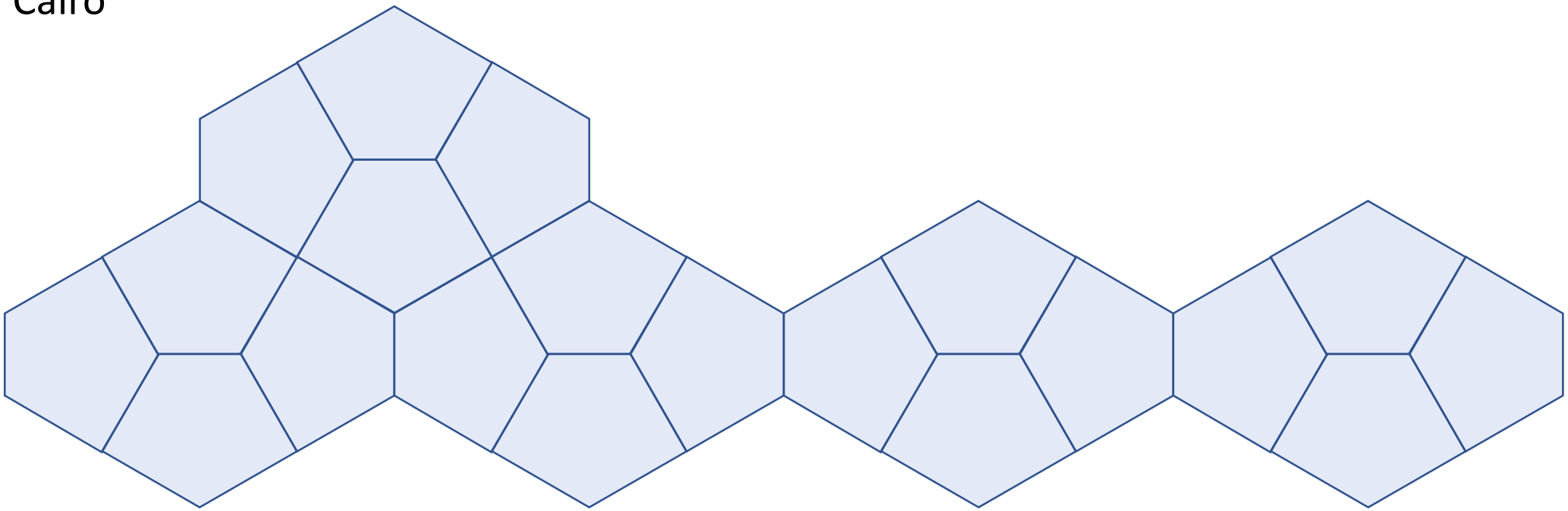
Cairo



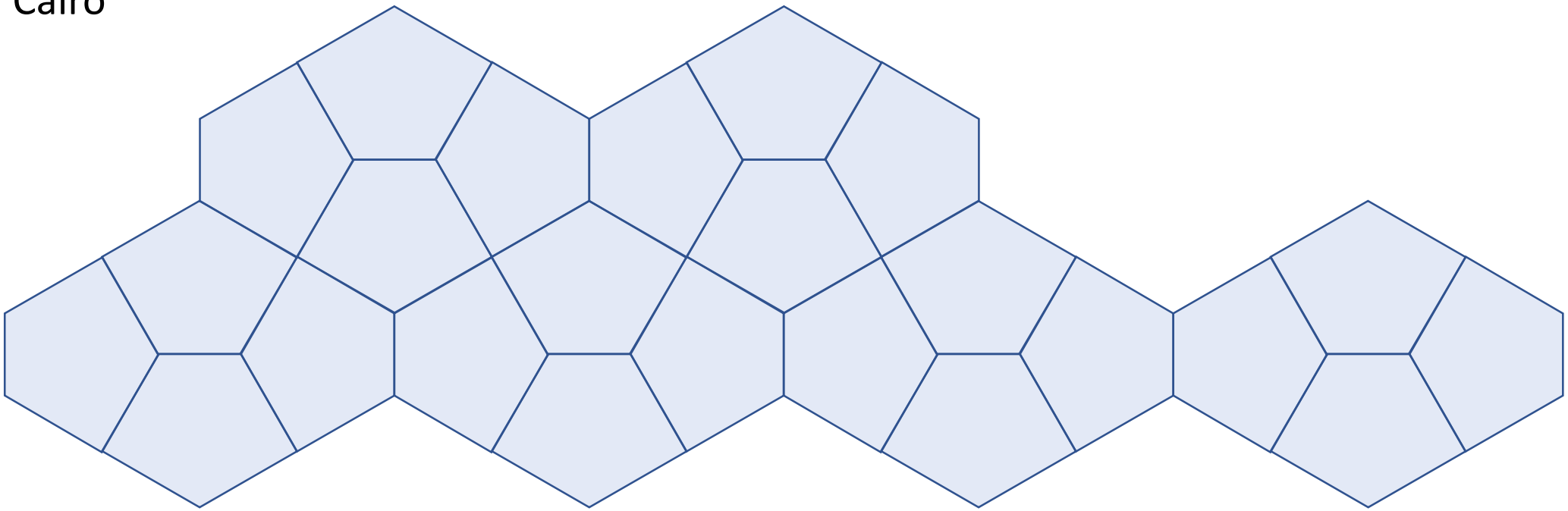
Cairo



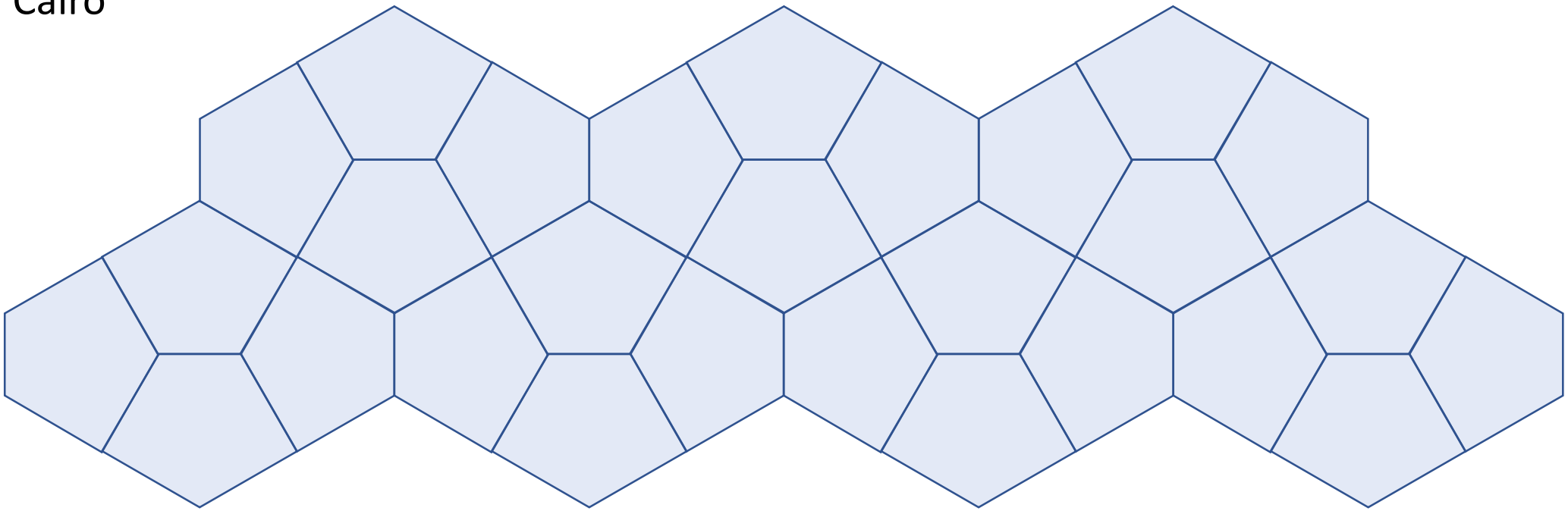
Cairo



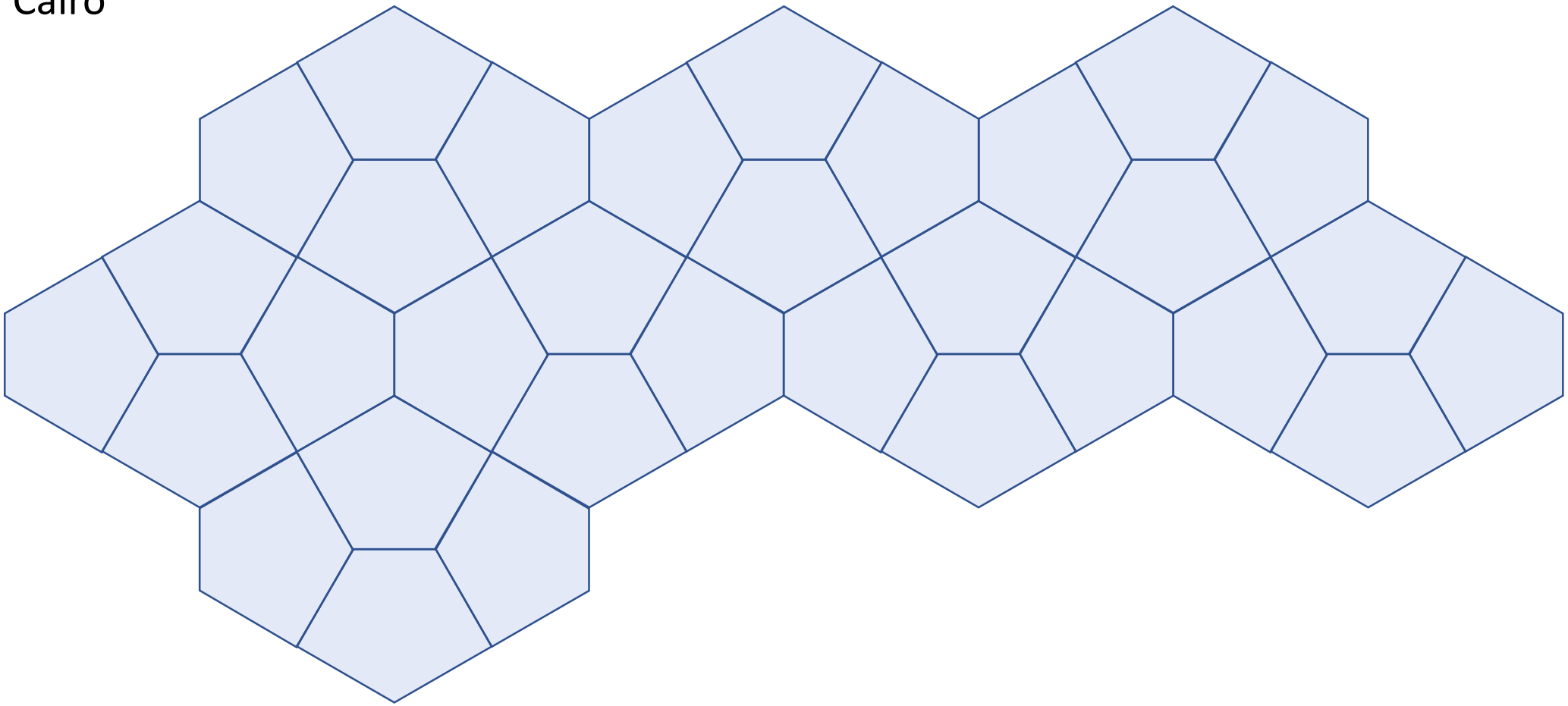
Cairo



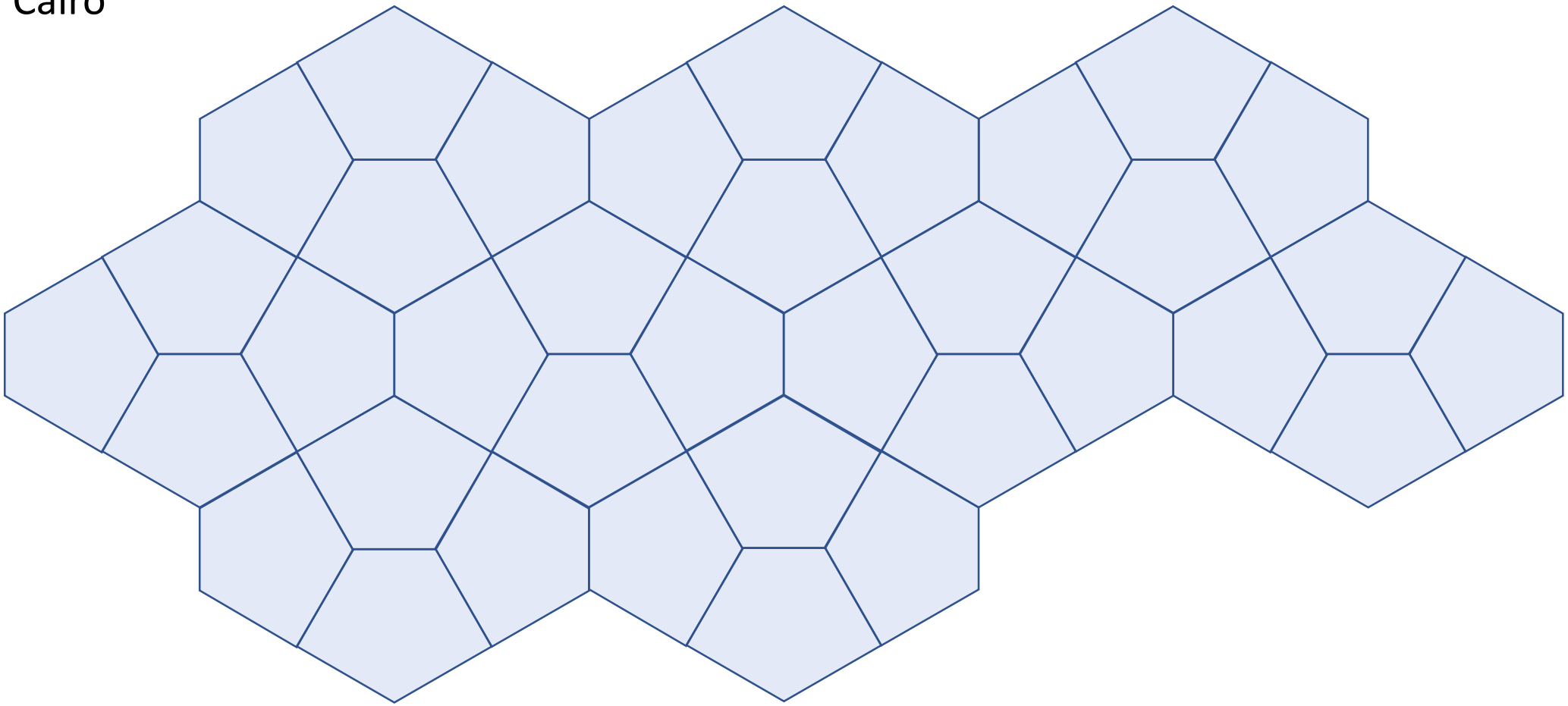
Cairo



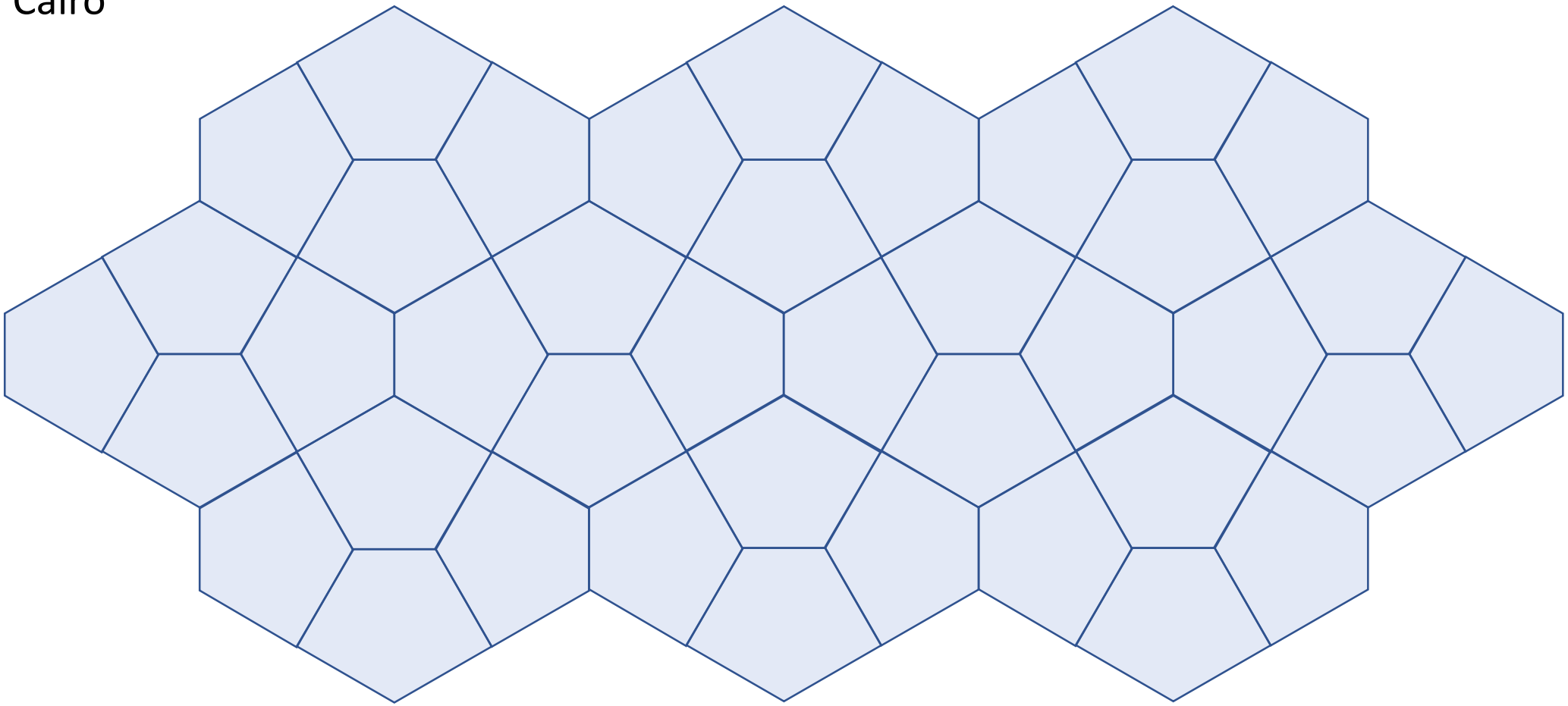
Cairo



Cairo



Cairo

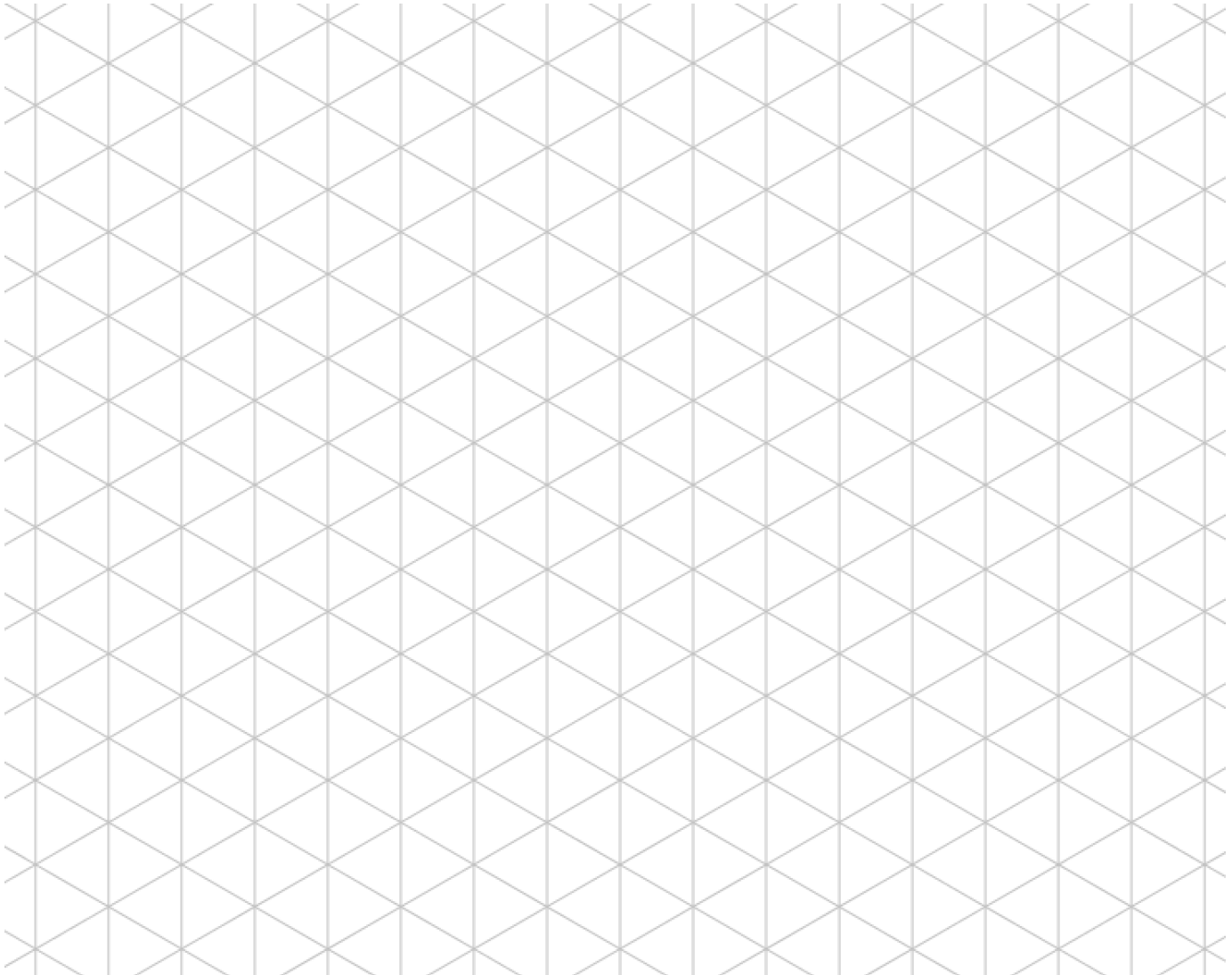


Cairo

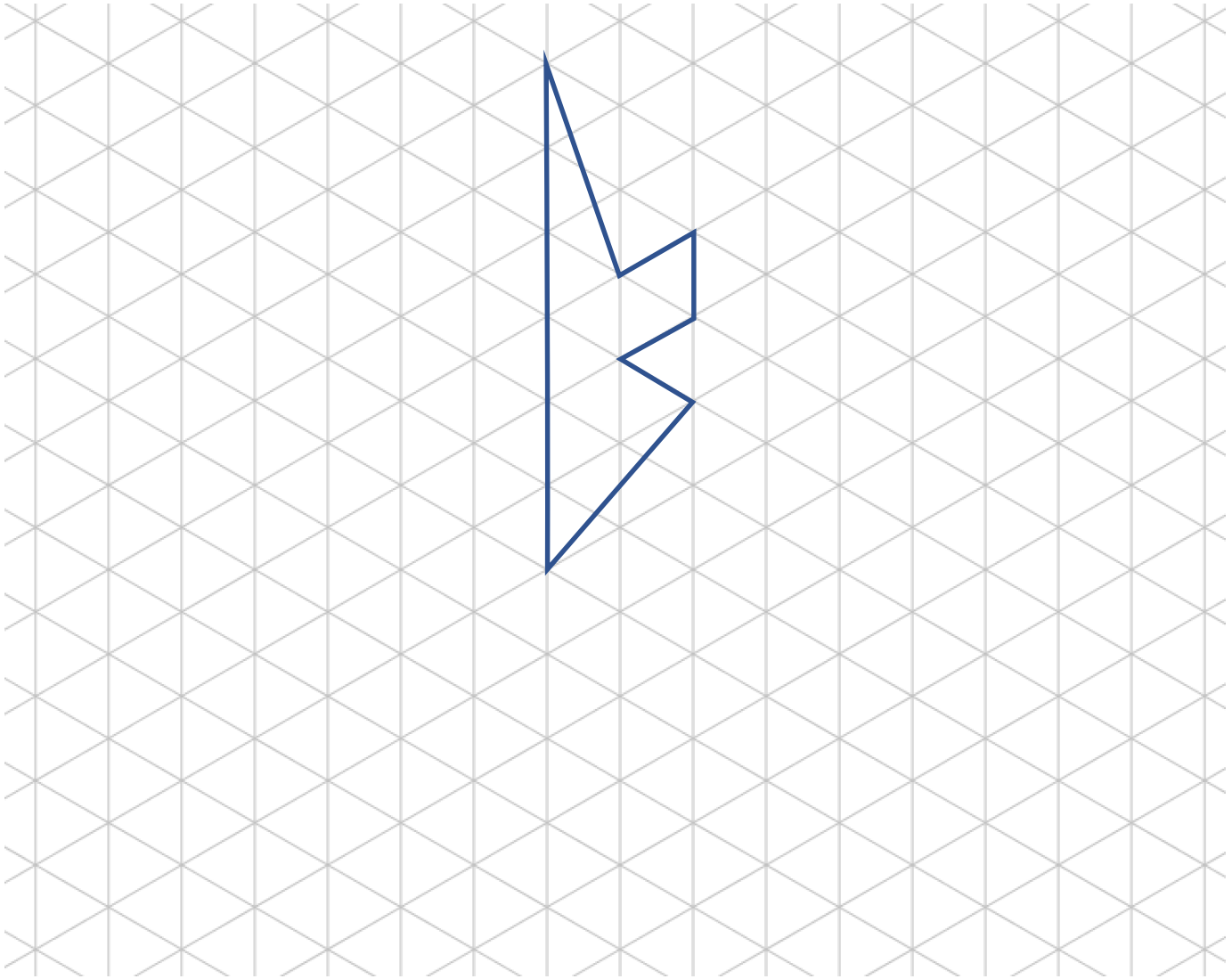
Back

Fish

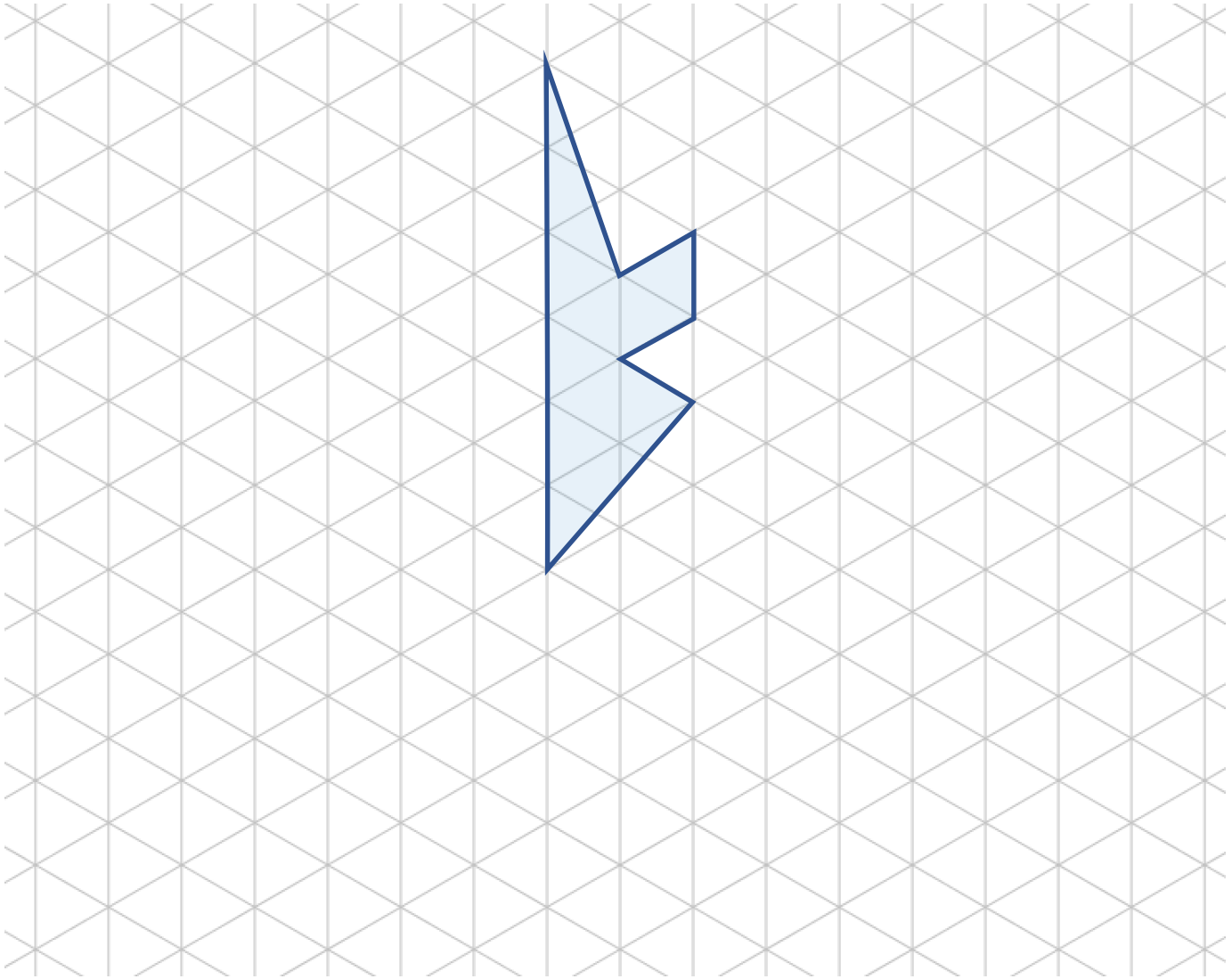
Fish



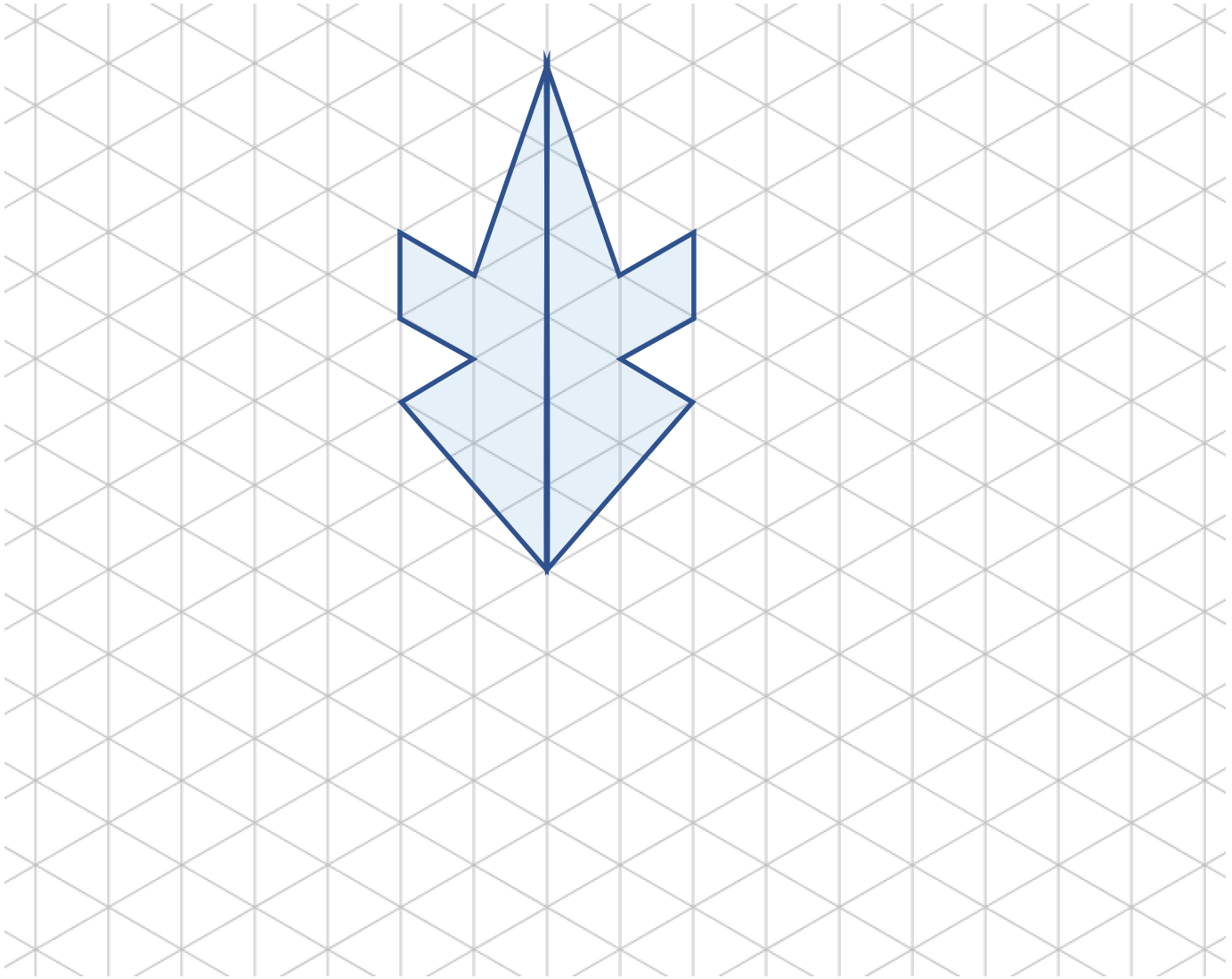
Fish



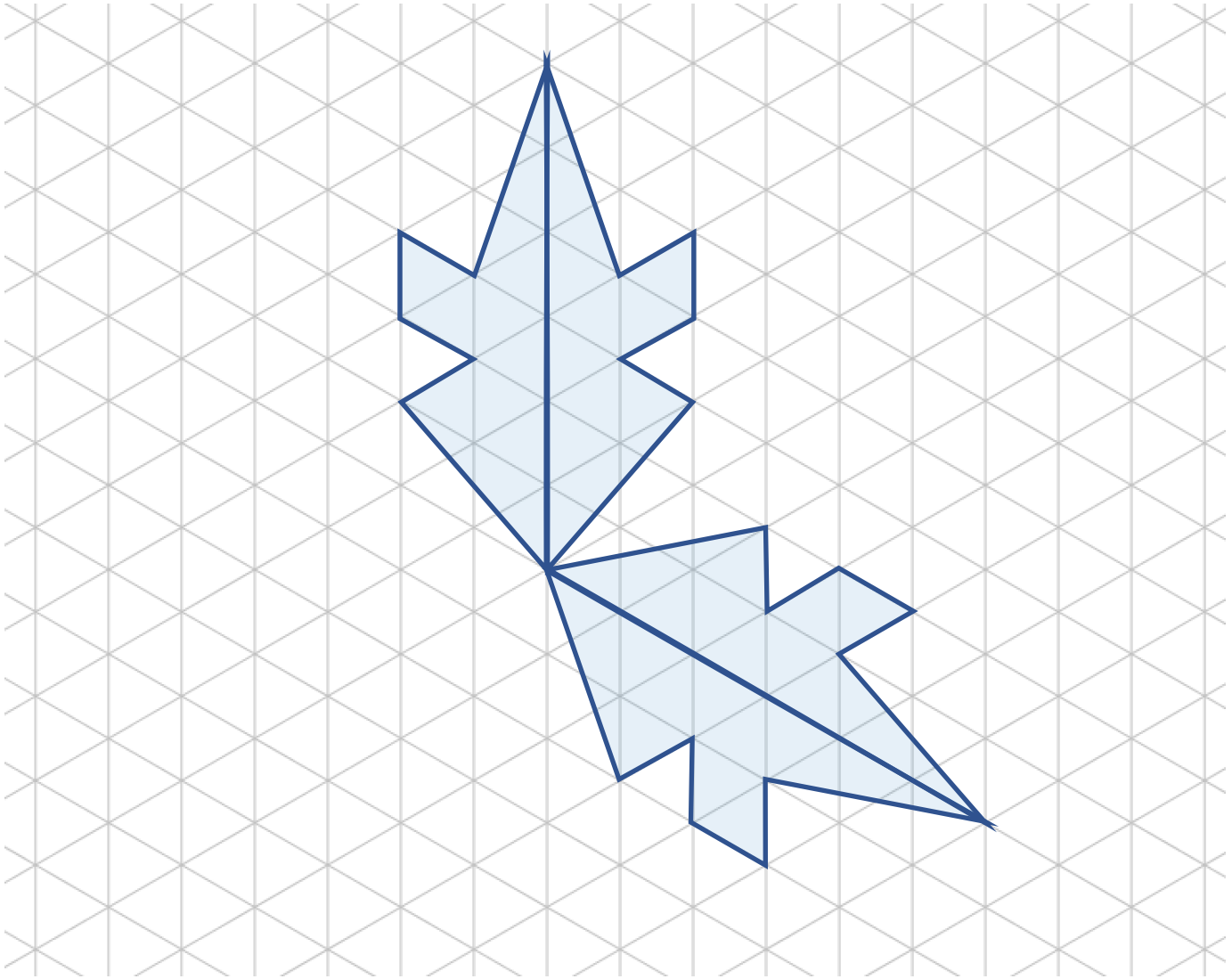
Fish



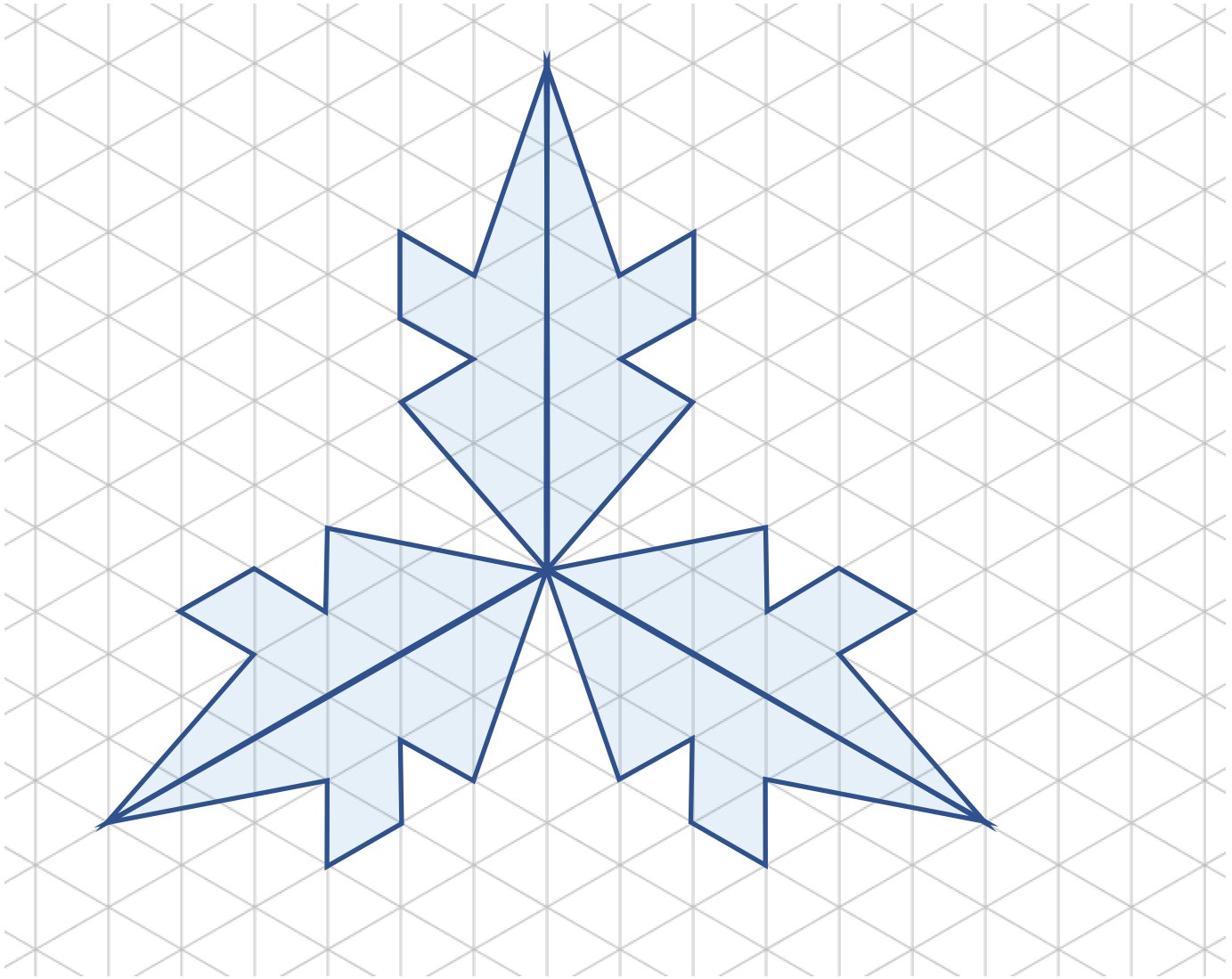
Fish



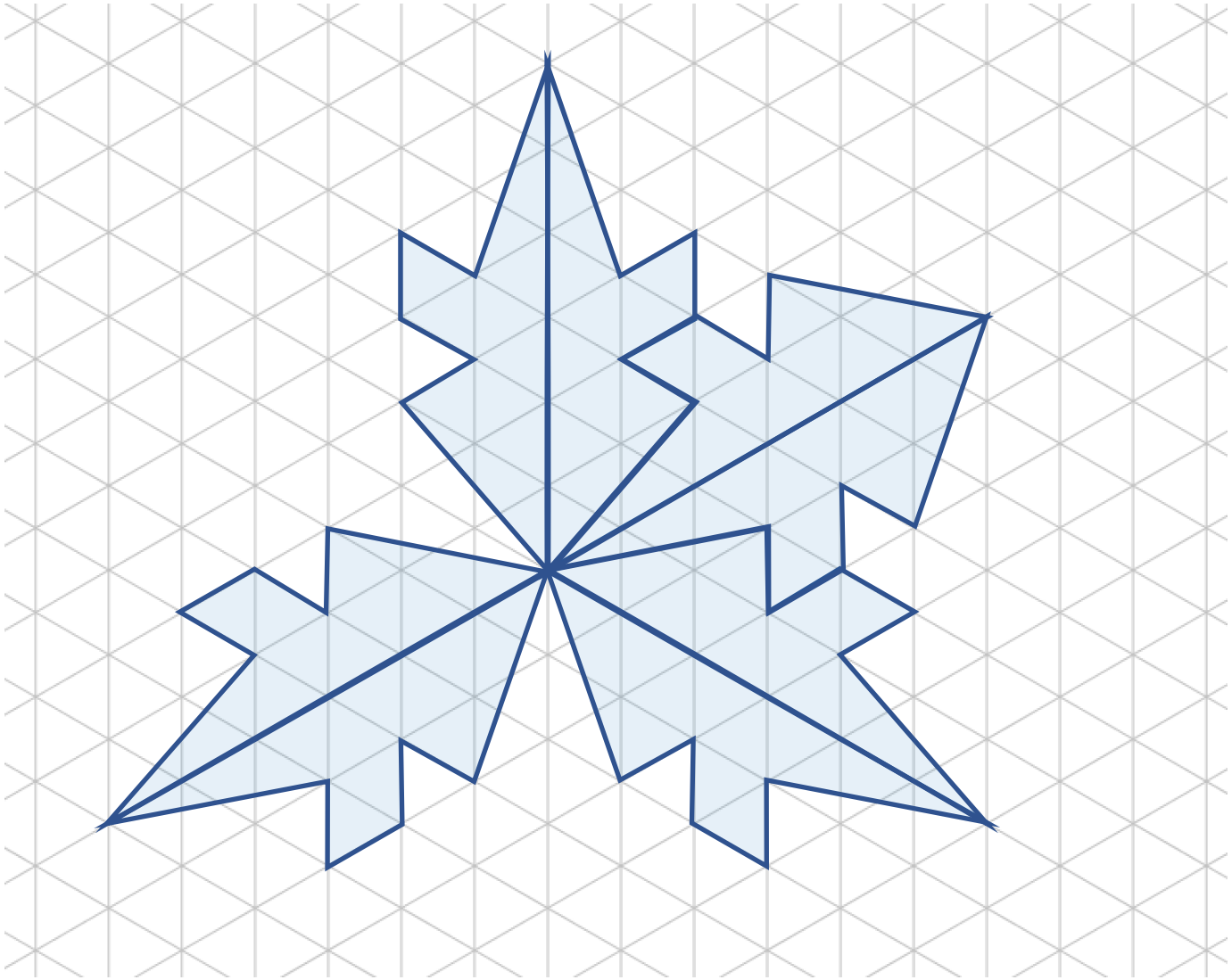
Fish



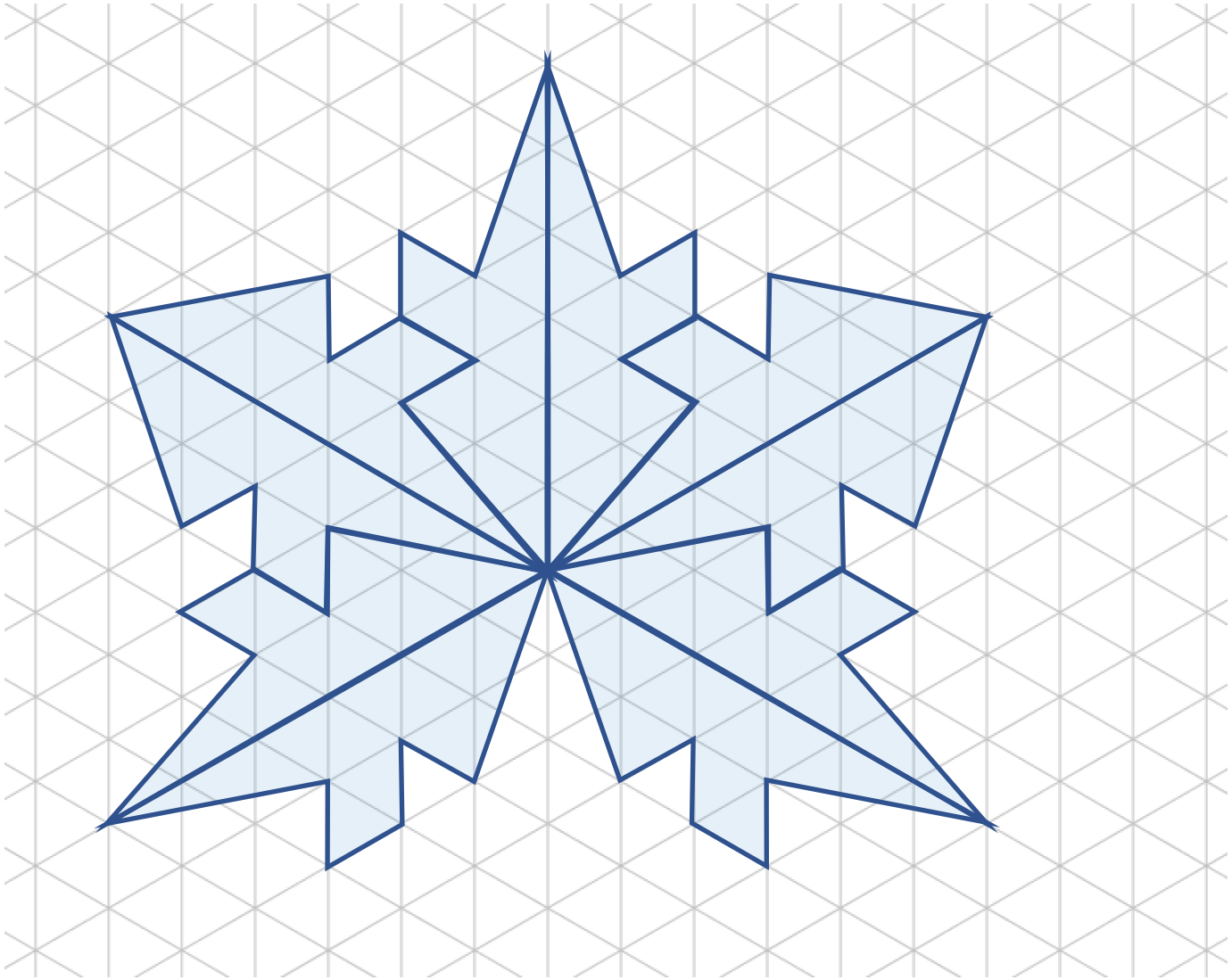
Fish



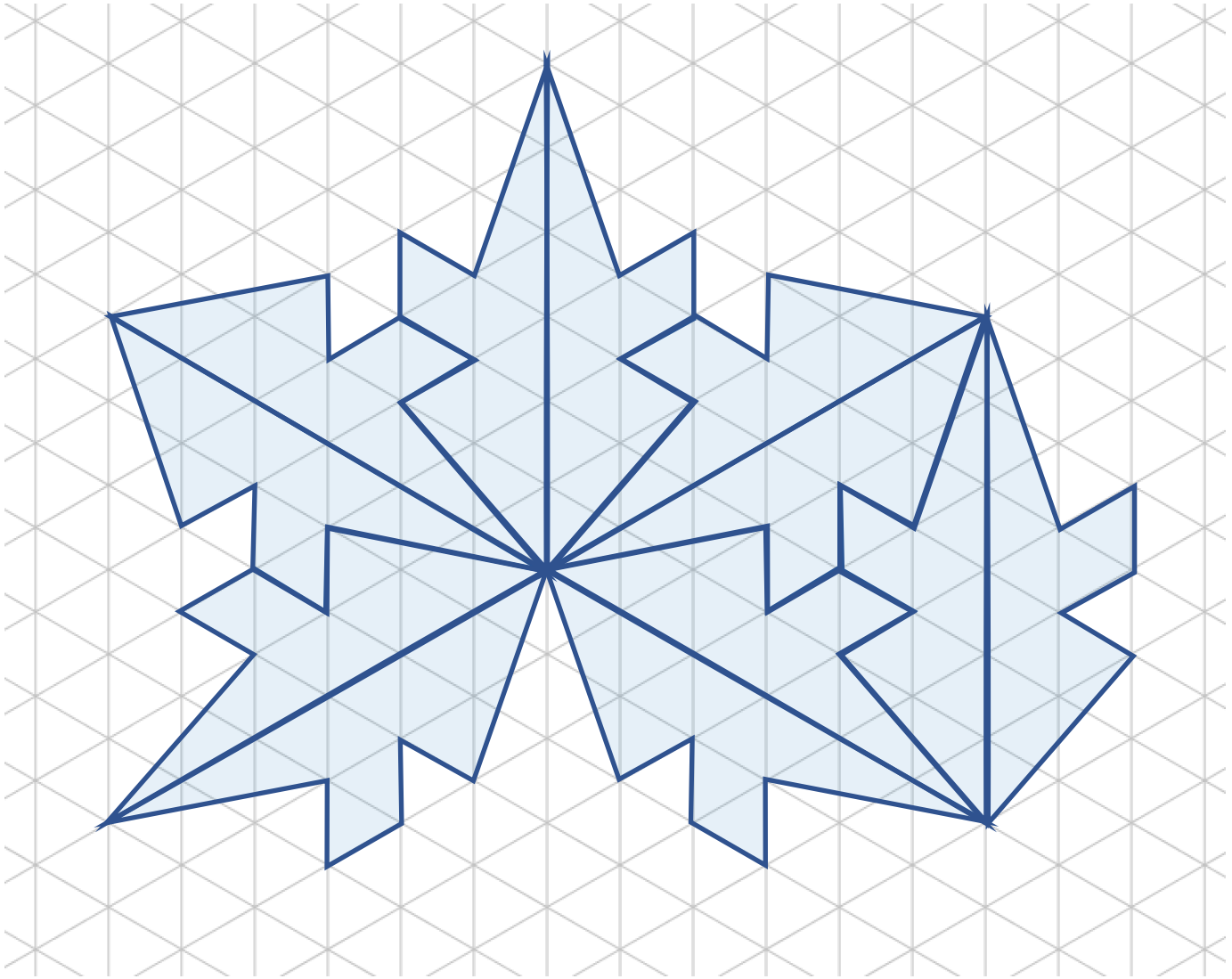
Fish



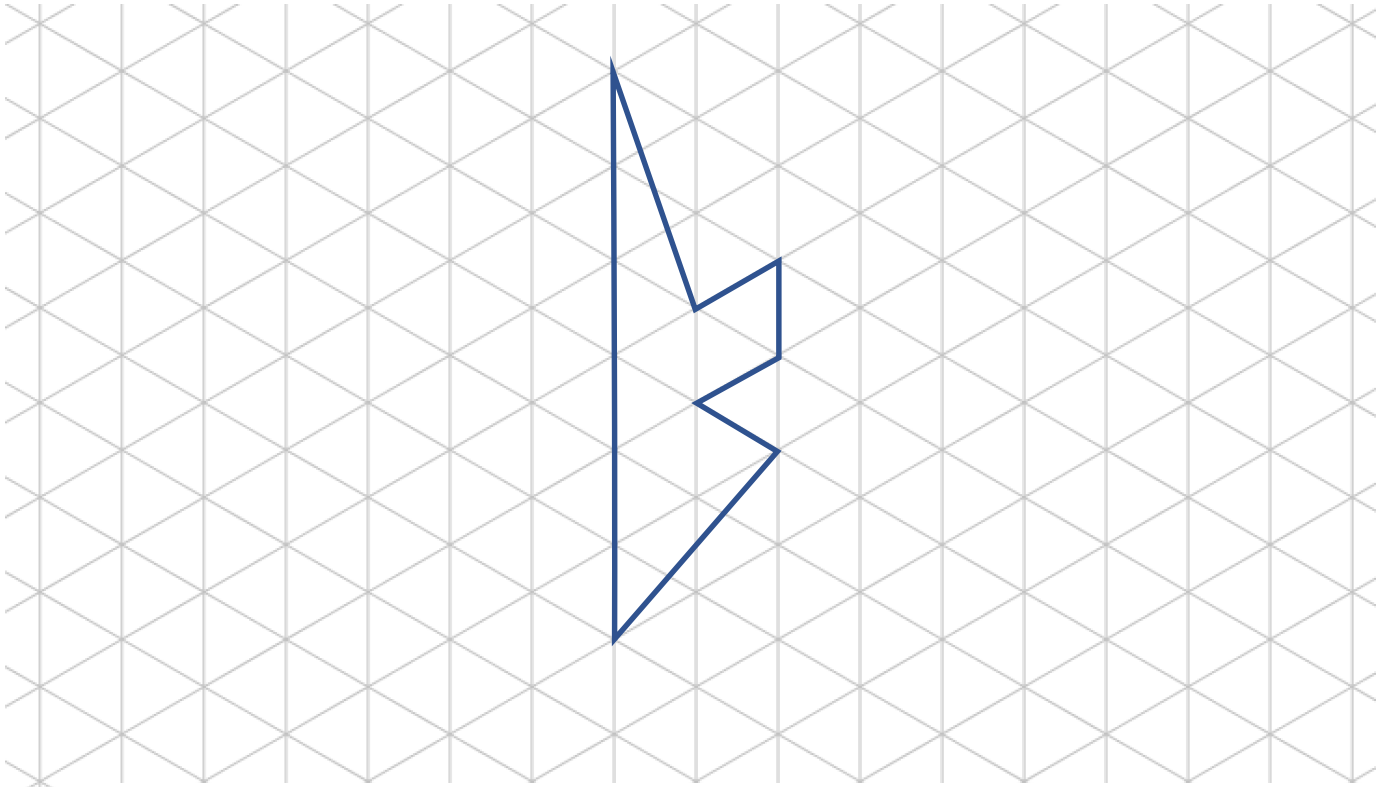
Fish



Fish



Fish



shape = [(0,0), (2,2), (1,2.5), (2,3), (2,4), 1,3.5), (0,6), (0,0)]

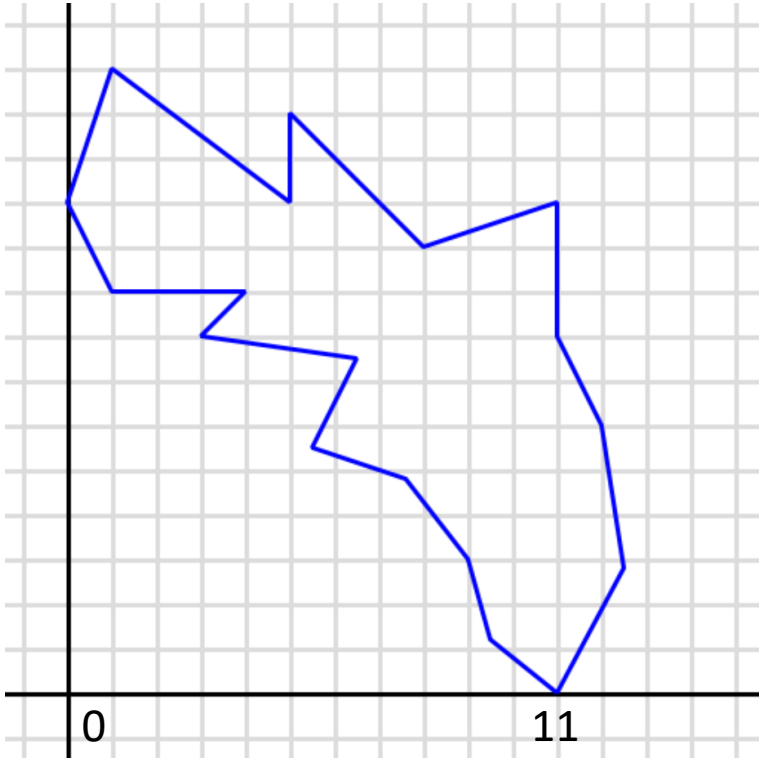
Fish

Back

Circular Fish

"Circular Fish", 1956 woodcut

Circular Fish

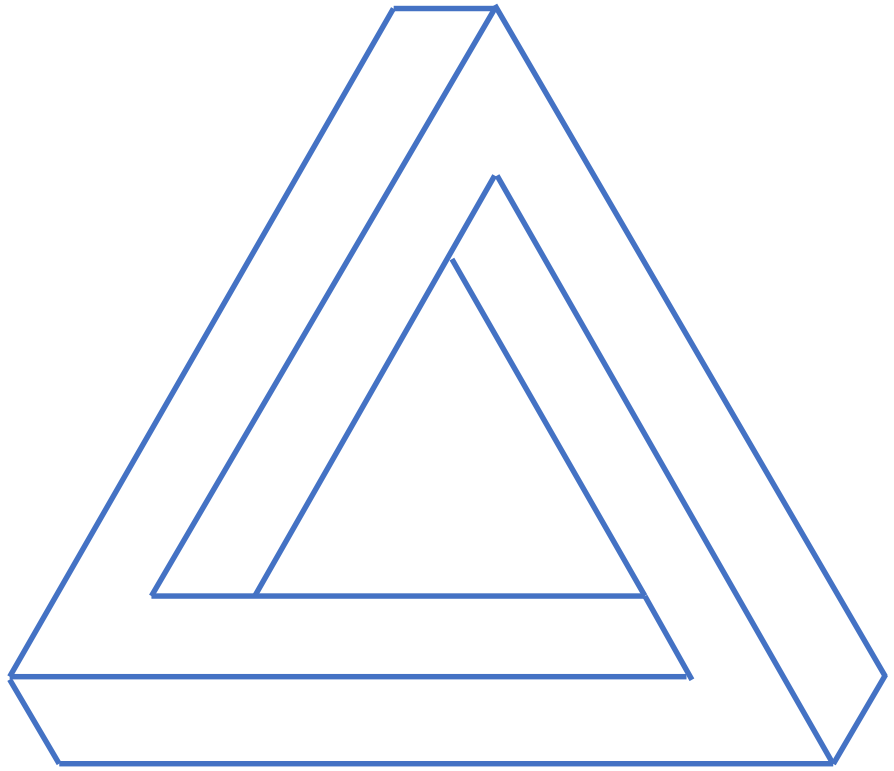


fish = [(11, 0), (12.5, 2.8), (12, 6), (11, 8), (11, 11), (8, 10), (5, 13), (5, 11), (1, 14), (0, 11), (1, 9), (4, 9),
(3, 8), (6.5, 7.5), (5.5, 5.5), (7.55, 4.8), (9, 3), (9.5, 1.2), (11, 0)]

Circular Fish

Back

Penrose



Penrose

Penrose

Penrose

Penrose

Penrose

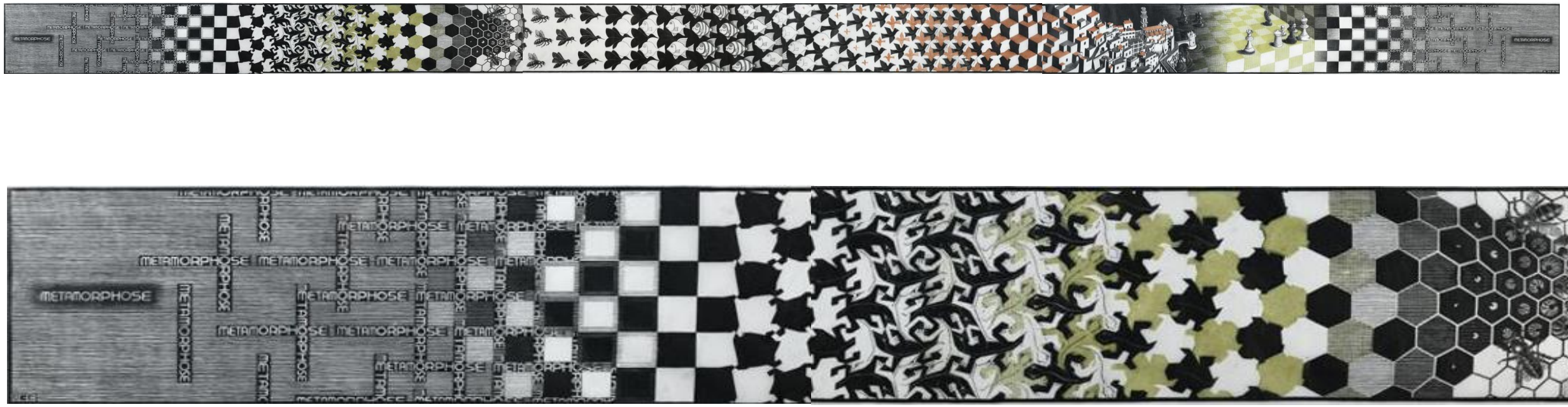
Penrose

Back

Examples

Triangles

Birds



"Metamorphose II", 1940 woodcut