In [66]:
```
from IPython.display import Image
```

Riddler Classic - June 26, 2020 (https://fivethirtyeight.com/features/can-you-connect-the-dots/)

Problem

Polly Gawn loves to play "connect the dots." Today, she's playing a particularly challenging version of the game, which has six unlabeled dots on the page. She would like to connect them so that they form the vertices of a hexagon. To her surprise, she finds that there are many different hexagons she can draw, each with the same six vertices.

What is the greatest possible number of unique hexagons Polly can draw using six points?

(Hint: With four points, that answer is three. That is, Polly can draw up to three quadrilaterals, as long as one of the points lies inside the triangle formed by the other three. Otherwise, Polly would only be able to draw one quadrilateral.)

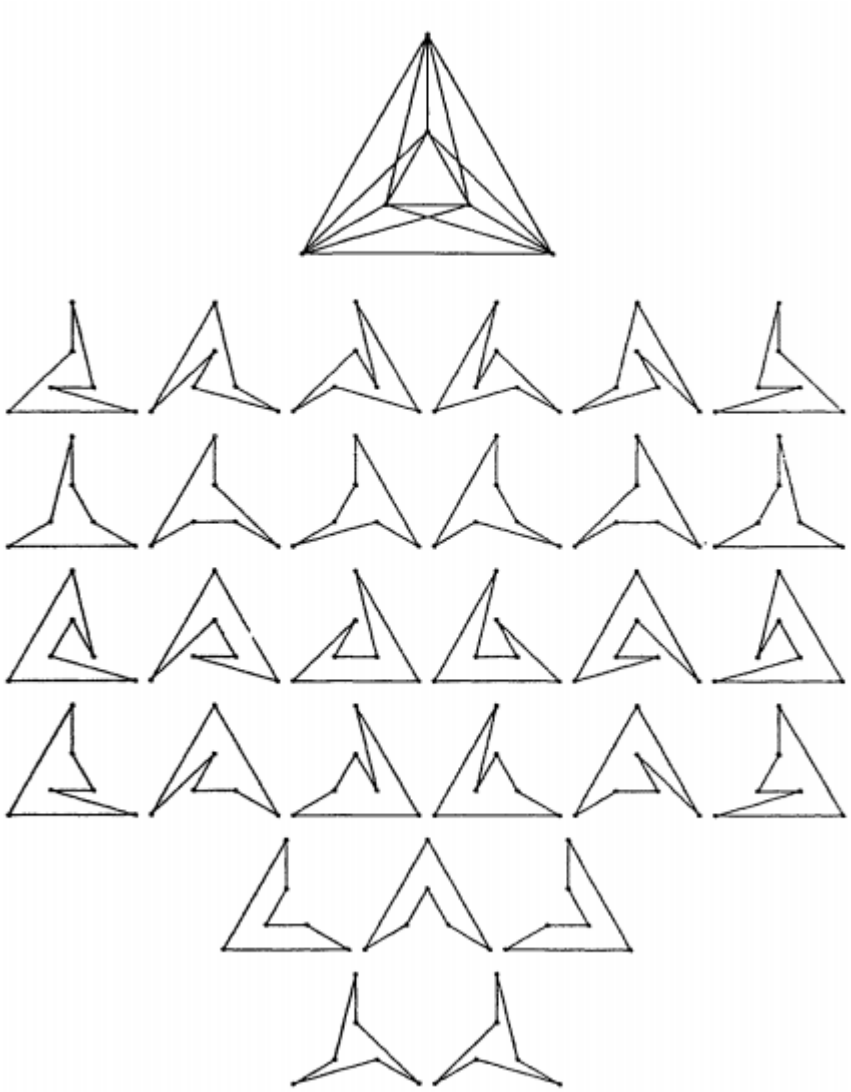
Extra Credit: What is the greatest possible number of unique heptagons Polly can draw using seven points?

Solution

Adimittedly, I did not have go through and write my own solution for this problem. After experimenting with pencil and paper for a while, I began to realize that not only would I have to enumerate all the different simply polygons from a set of n points (a task commonly called *simple polygonalization* or *crossing-free Hamiltonian cycle problem*) but I would have to try a bunch of different point configurations since it is clear from the problem that the number of simple polygonalizations depends on how the points are constructed. I thought better of launching into a brute force computational solution because already with 6 points, the number of different point configurations and hamiltonian paths that would have to be tried is quite large (not to mention I'm not even sure how to appropriatly enumerate all of the point sets the form unique simple polygonalizations). Instead I started digging around to better understand the problem. In my searching, I happened to find the answer and extra credit which comes from Oswin Aichholzer (http://www.ist.tugraz.at/staff/aichholzer//research/rp/triangulations/ordertypes/applications.txt). In his studies, he finds that the most simple polygons that can be formed is 29 and 92 for 6 and 7 points respectively. Below is an image of the different combinations for 6 points from Hayward (https://link.springer.com/content/pdf/10.1007/BF02187887.pdf).

In [67]:
```
Image("6point_polygonalizations.png")
```

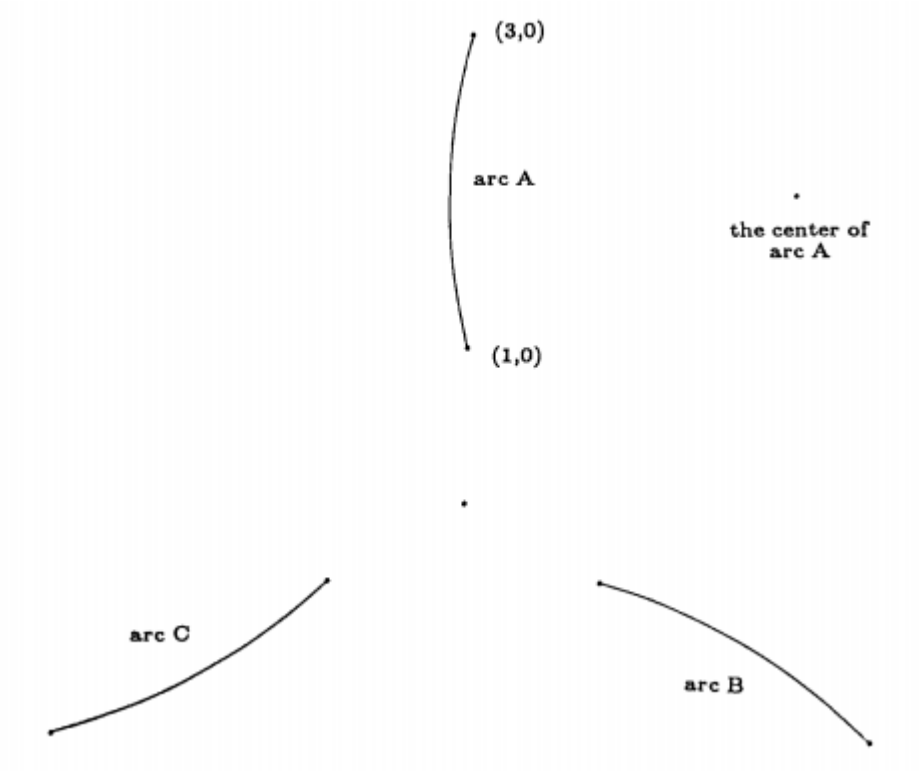
Out[67]:



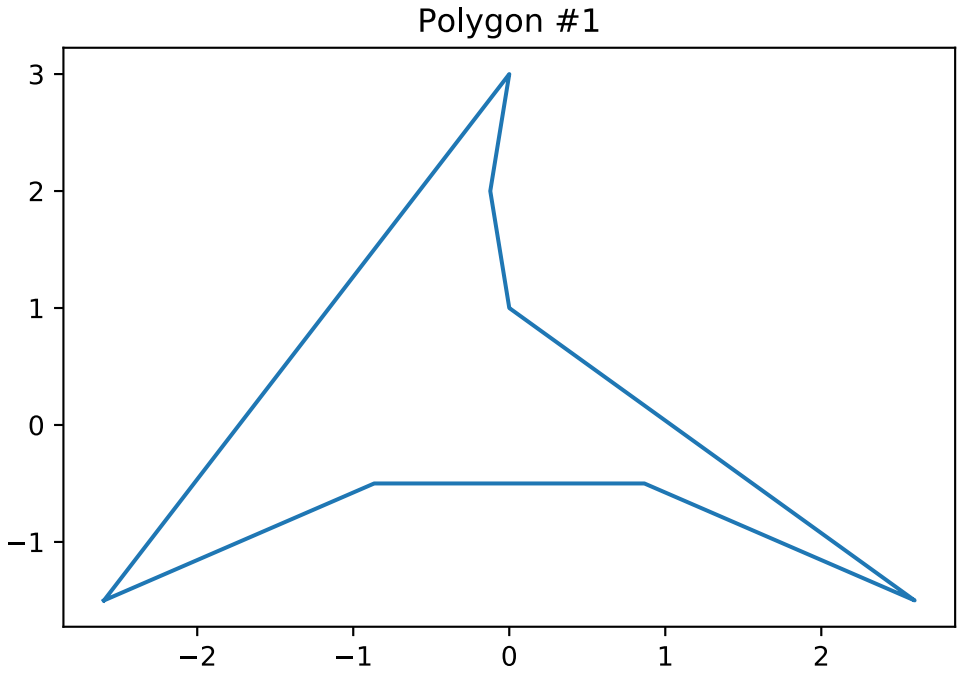
As Hayward's method for selecting points makes clear, its not obvious how you place points to maximize the number of simple polynomializations however it turns out, the optimal Haywards method for selecting points to maximize the number of simply polygonalizations works for up to 8 point. So we could construct the points set that maximizes the number of polygonalizations for 7 points by placing those points on the ends of each of the three arcs (see below) as well as one point in the middle of one of the arcs to get a total of 7 points. With these points, we can maximize the number of simple polygonalizations.

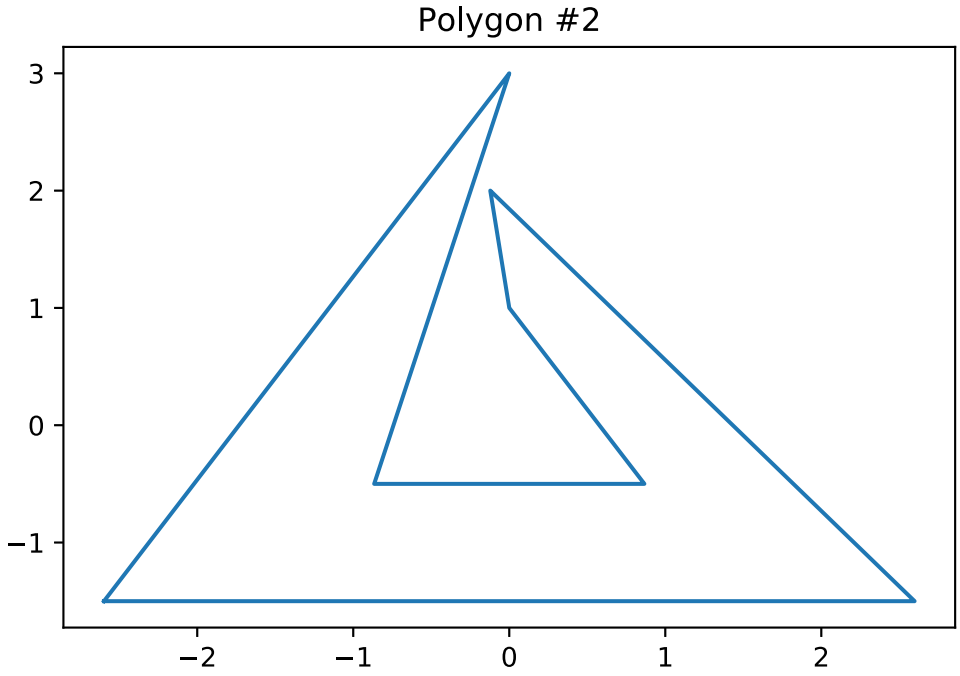
In [68]:
```
Image("arcs.png")
```

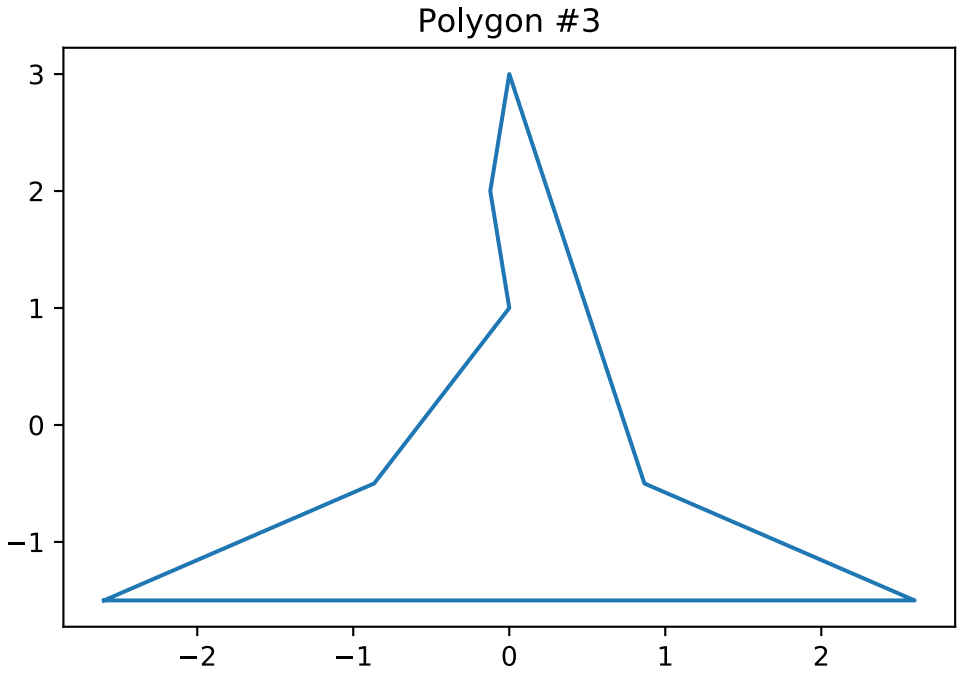
Out[68]:

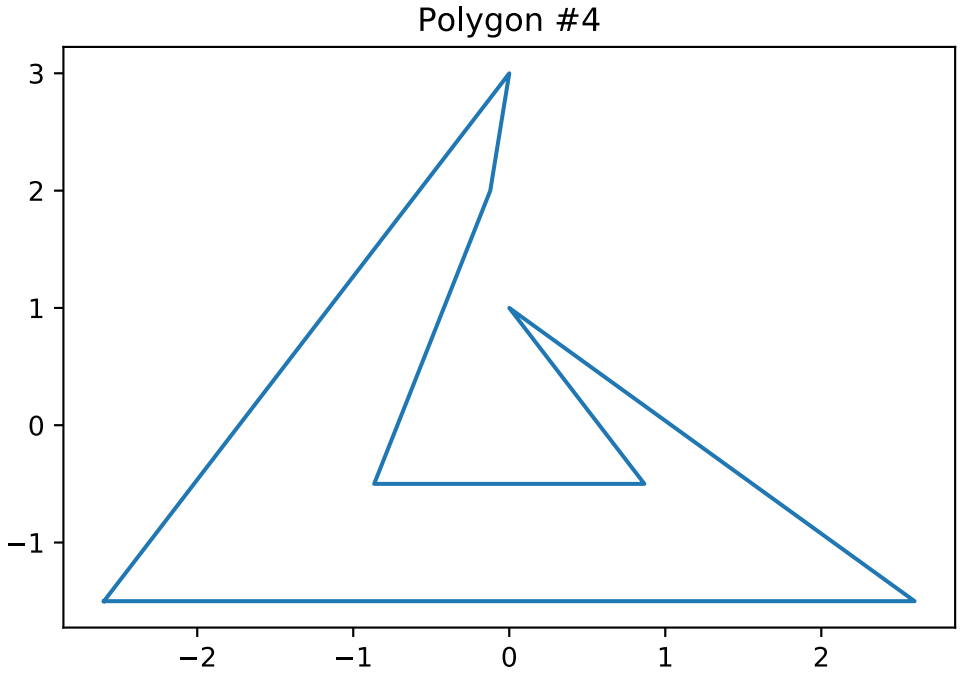


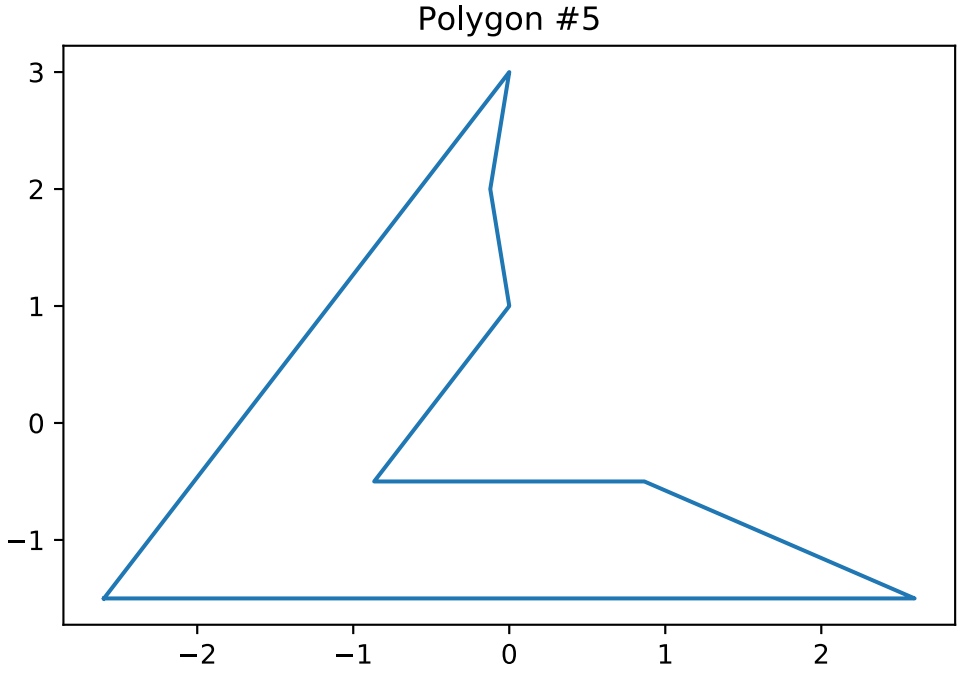
In [69]:
```
import matplotlib.pyplot as plt
from glob import glob
import re
data_files = glob("/home/xavier/ProgramingProjects/Riddler/2020-06-2
6/genpoly-rpg/results.out-*line")
polygons = {}
i = 0
for file_name in data_files:
    xs = []
    ys = []
    with open(file_name) as f:
        next(f)
        for row in f:
            x, y = (float(s) for s in row.split())
            xs.append(x)
            ys.append(y)
    data = (tuple(xs), tuple(ys))
    if data in polygons:
        continue
    else:
        i += 1
        polygons[data] = i
        plt.figure()
        plt.title("Polygon #{}".format(i))
        plt.plot(xs,ys)
```

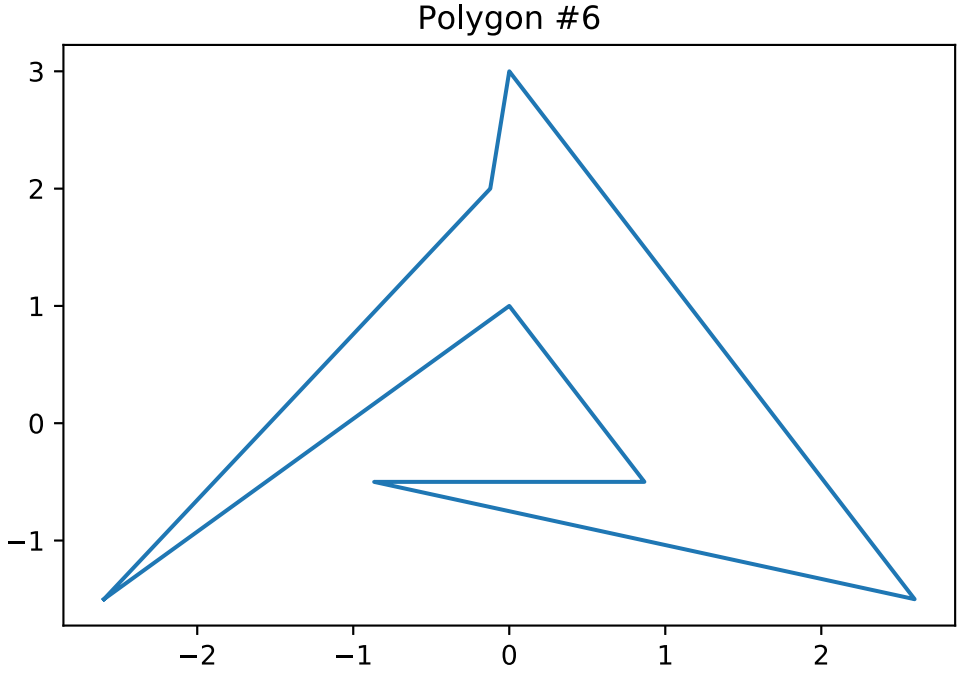

Polygon #1
3
2
1
0
−1
−2
−1
0
1
2

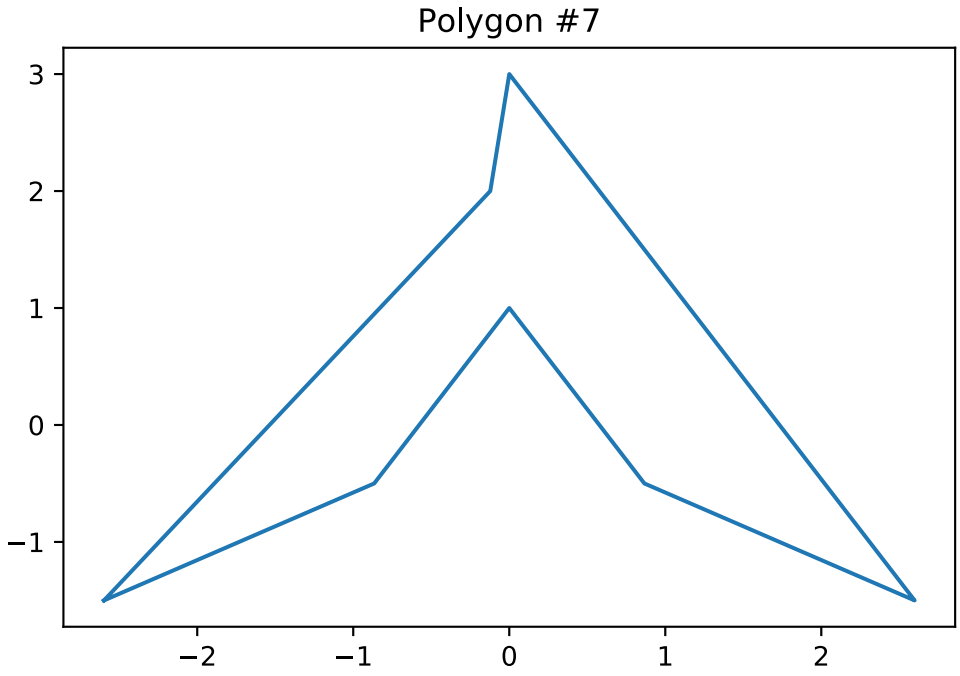

Polygon #2
3
2
1
0
−1
−2
−1
0
1
2

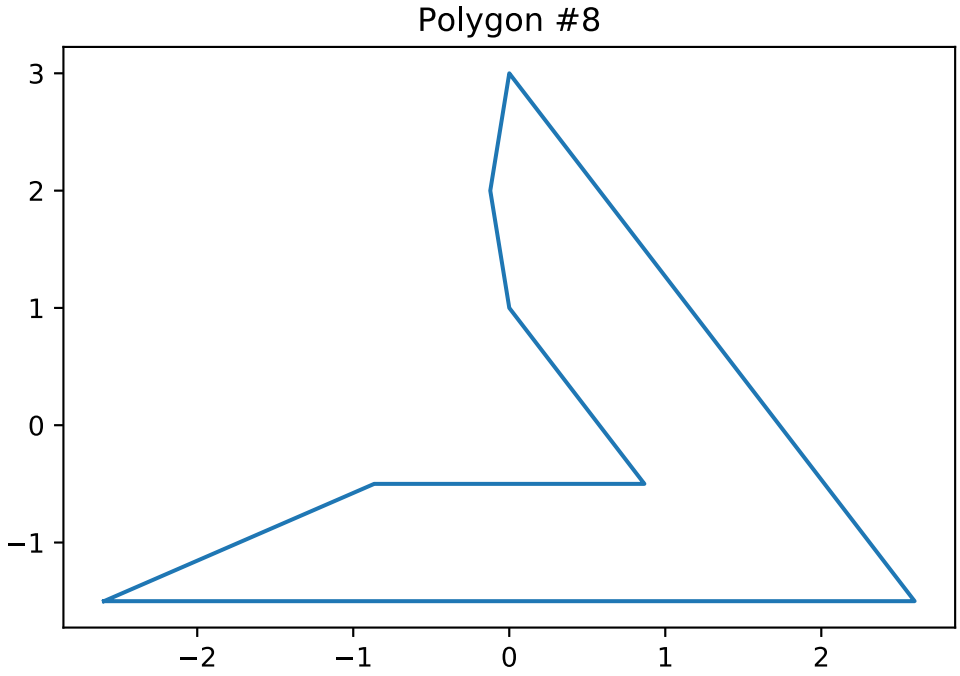

Polygon #3
3
2
1
0
−1
−2
−1
0
1
2

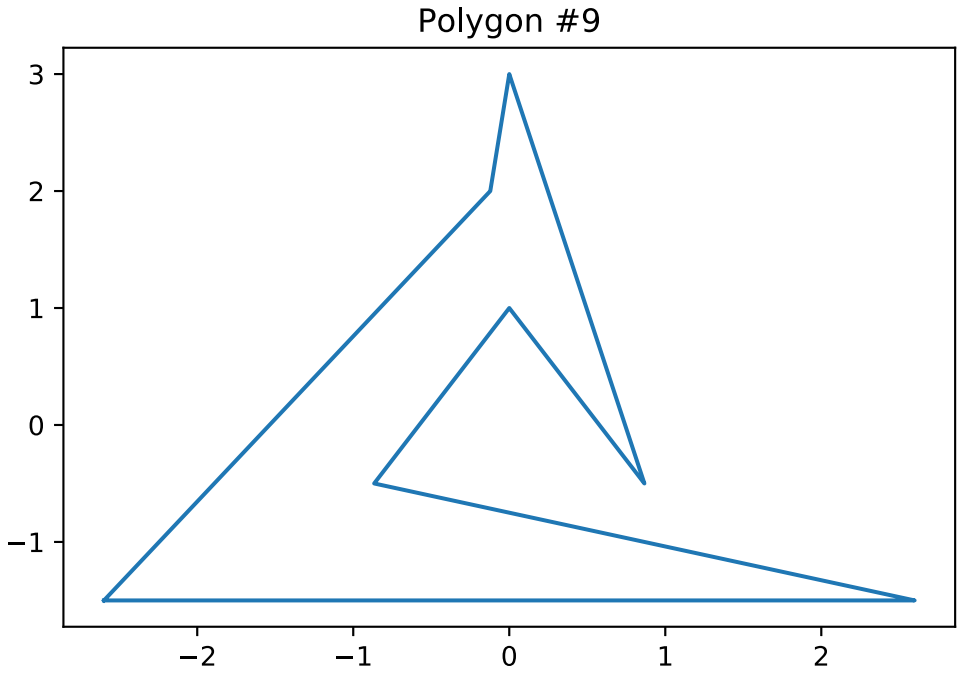

Polygon #4
3
2
1
0
−1
−2
−1
0
1
2

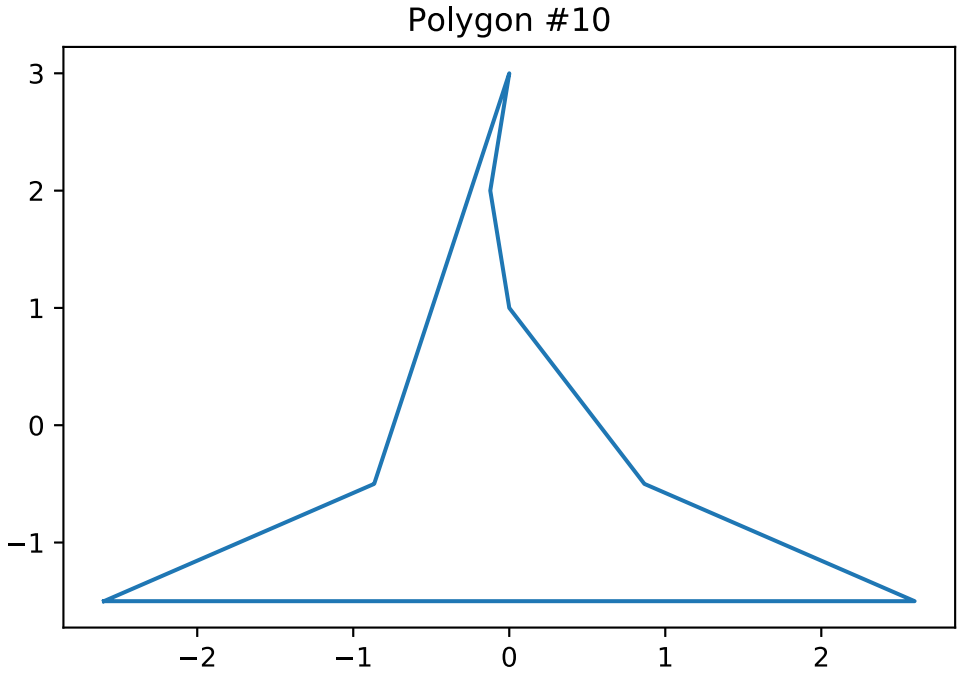

Polygon #5
3
2
1
0
−1
−2
−1
0
1
2

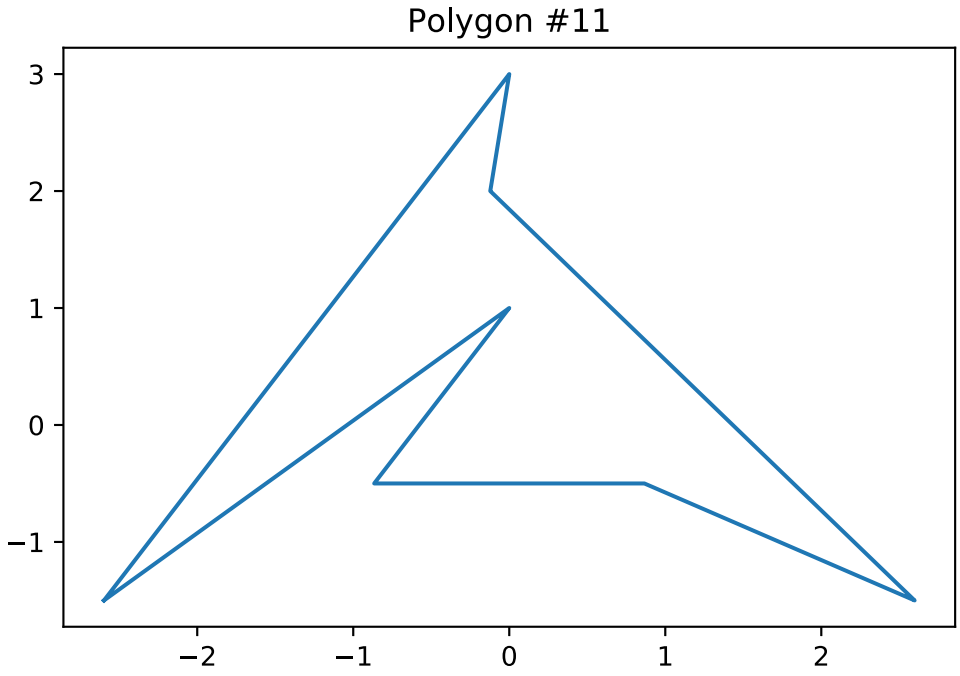

Polygon #6
3
2
1
0
−1
−2
−1
0
1
2

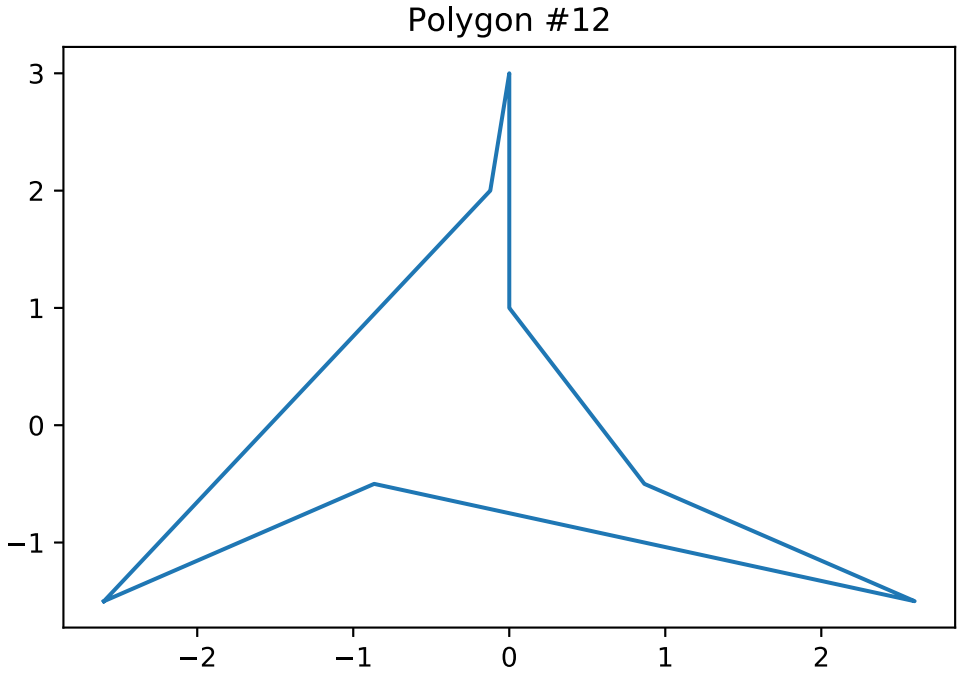

Polygon #7
3
2
1
0
−1
−2
−1
0
1
2

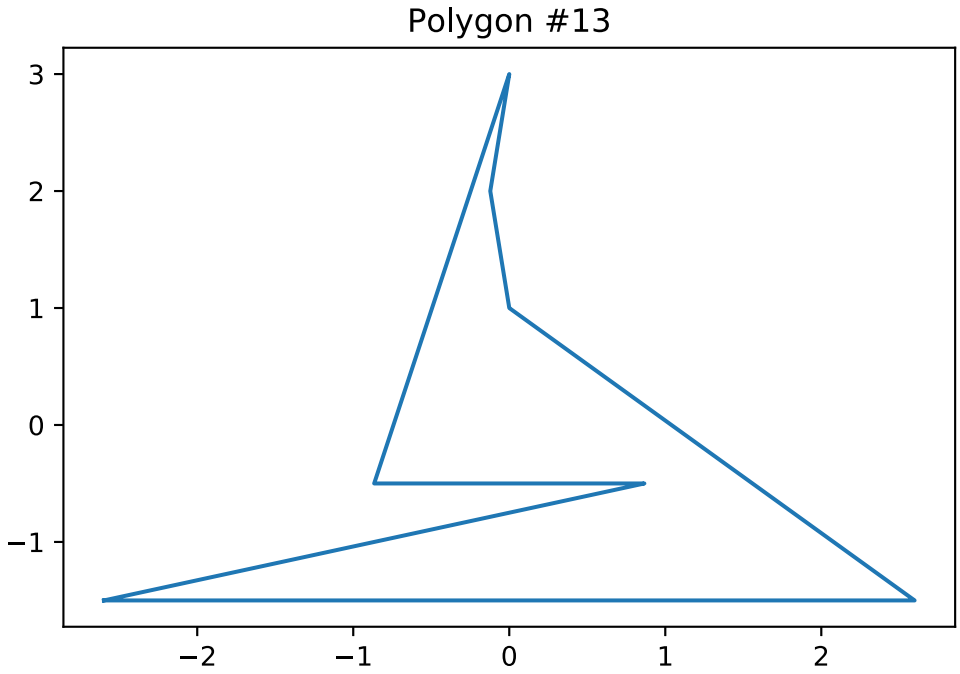

Polygon #8
3
2
1
0
−1
−2
−1
0
1
2

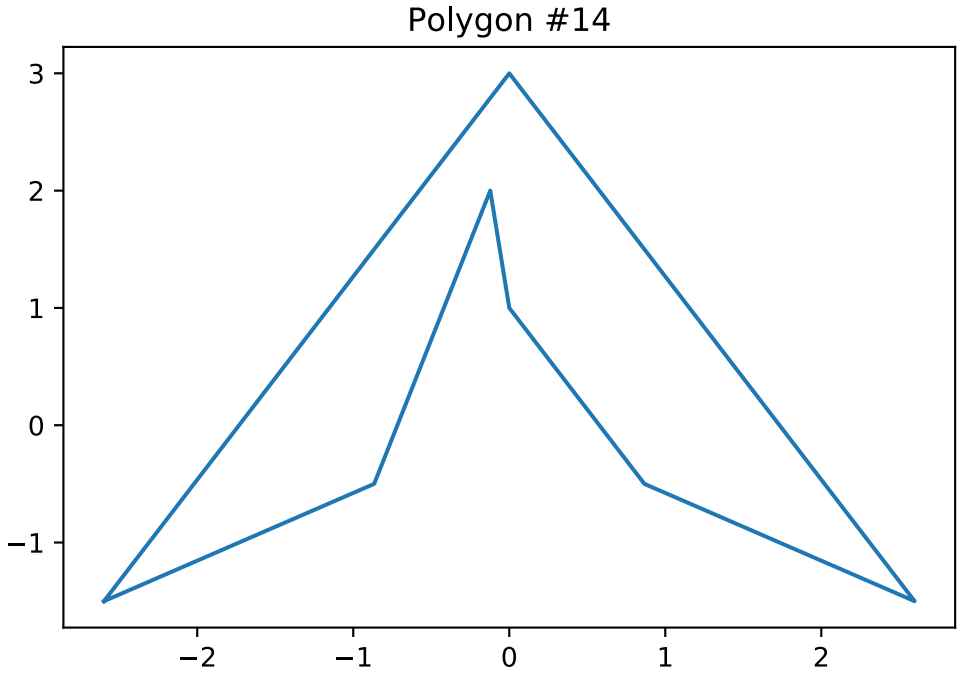

Polygon #9
3
2
1
0
−1
−2
−1
0
1
2

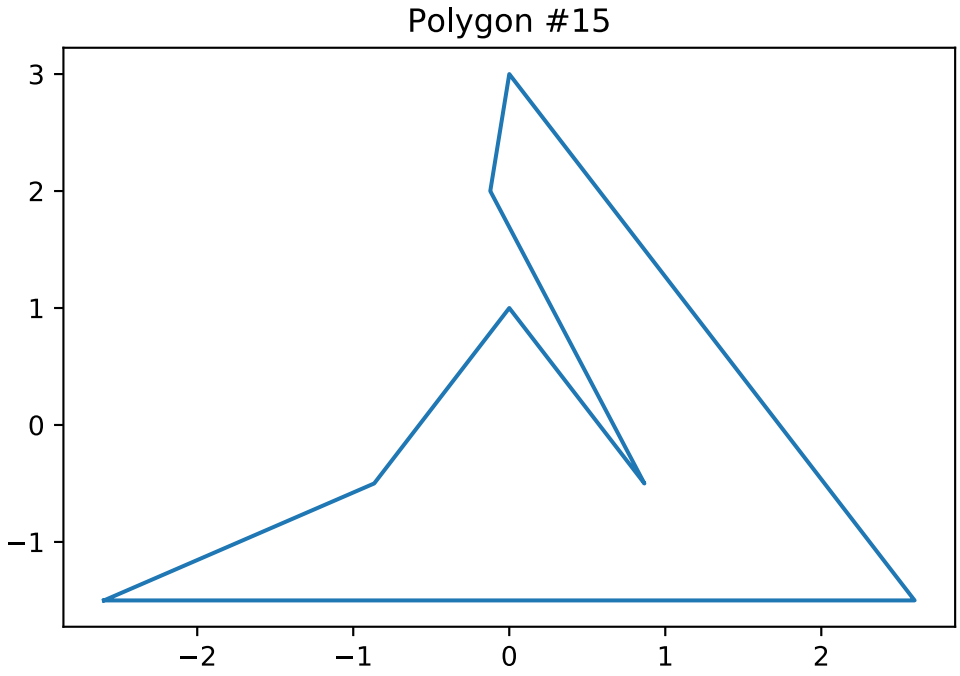

Polygon #10
3
2
1
0
−1
−2
−1
0
1
2

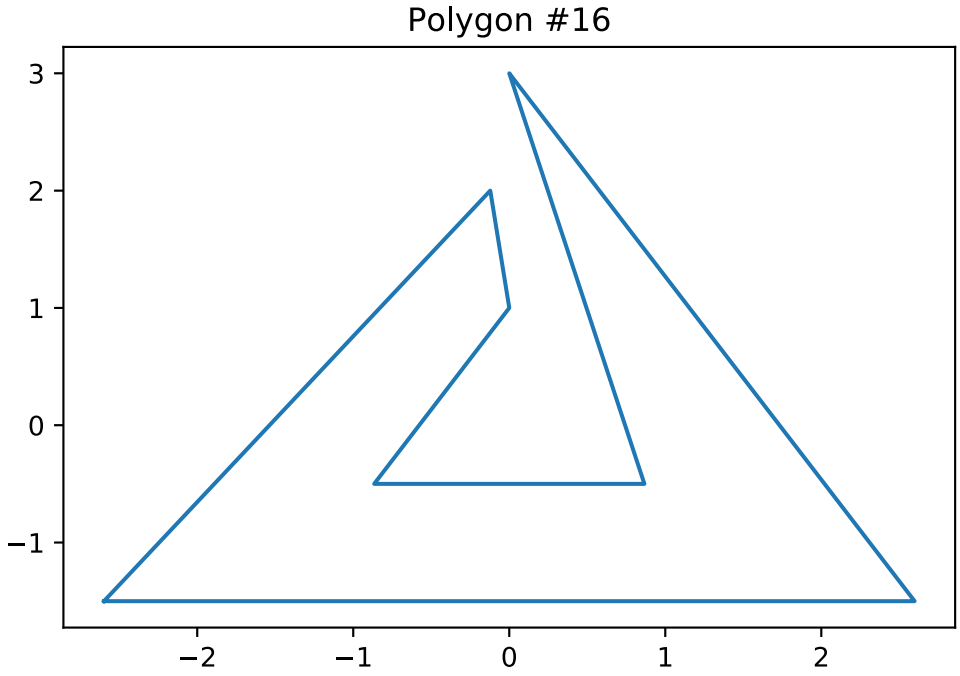

Polygon #11
3
2
1
0
−1
−2
−1
0
1
2

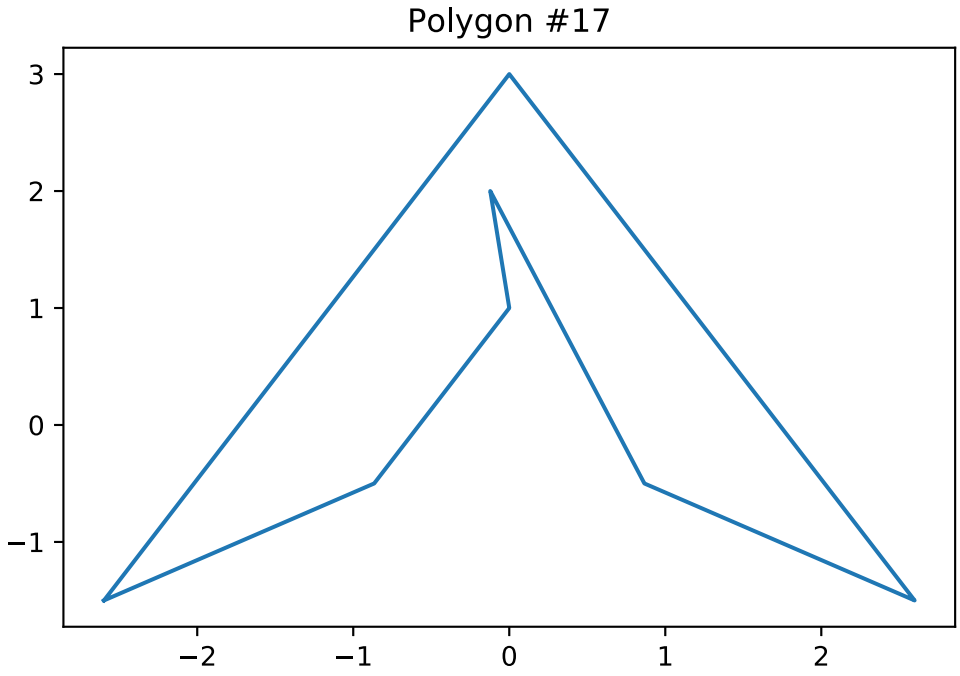

Polygon #12
3
2
1
0
−1
−2
−1
0
1
2

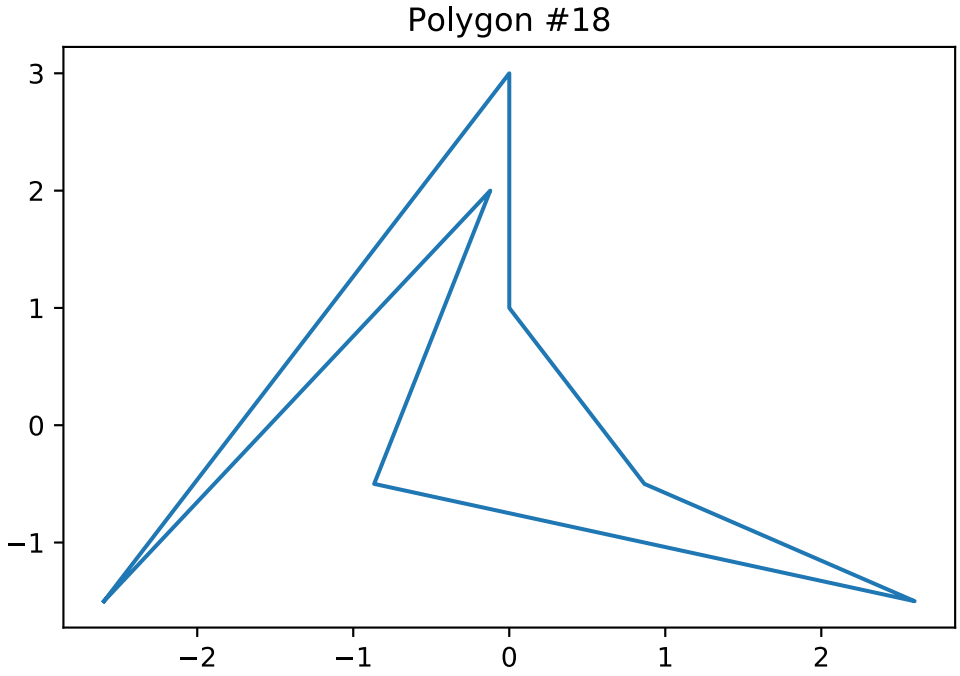

Polygon #13
3
2
1
0
−1
−2
−1
0
1
2

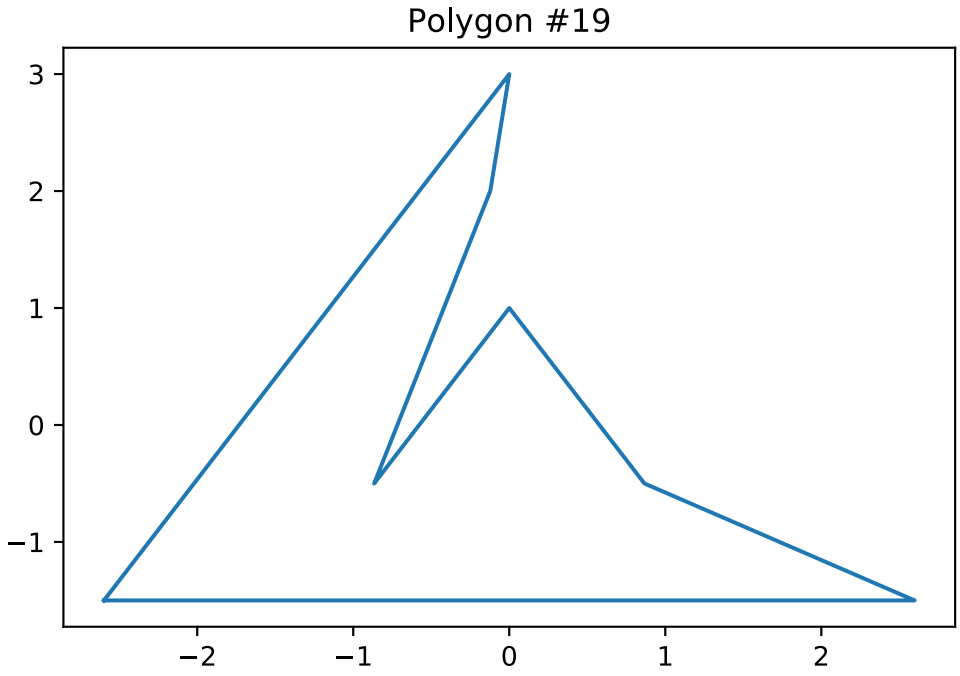

Polygon #14
3
2
1
0
−1
−2
−1
0
1
2

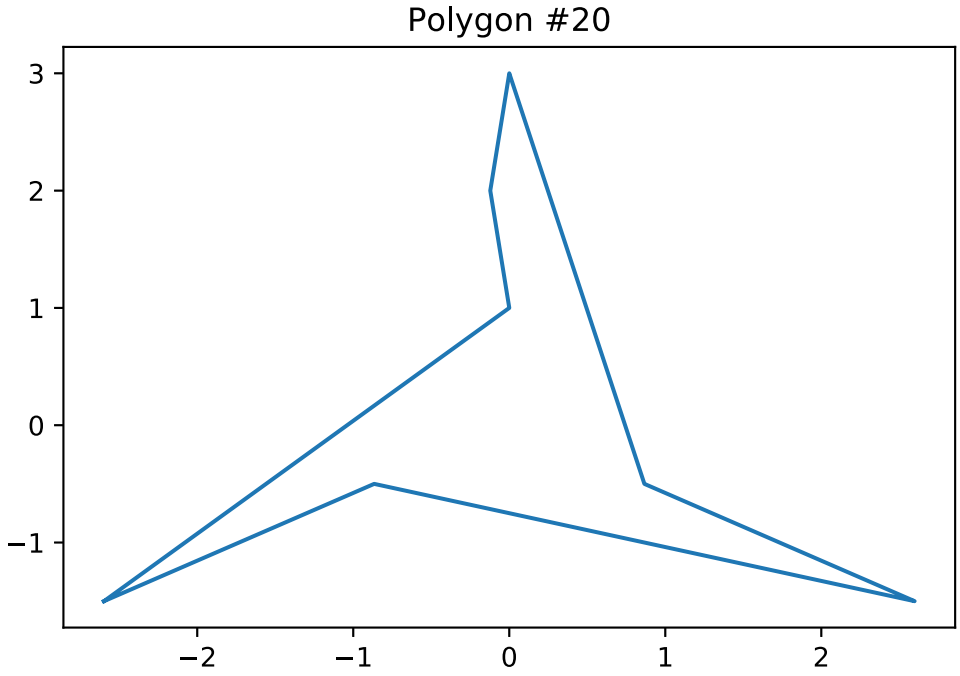

Polygon #15
3
2
1
0
−1
−2
−1
0
1
2

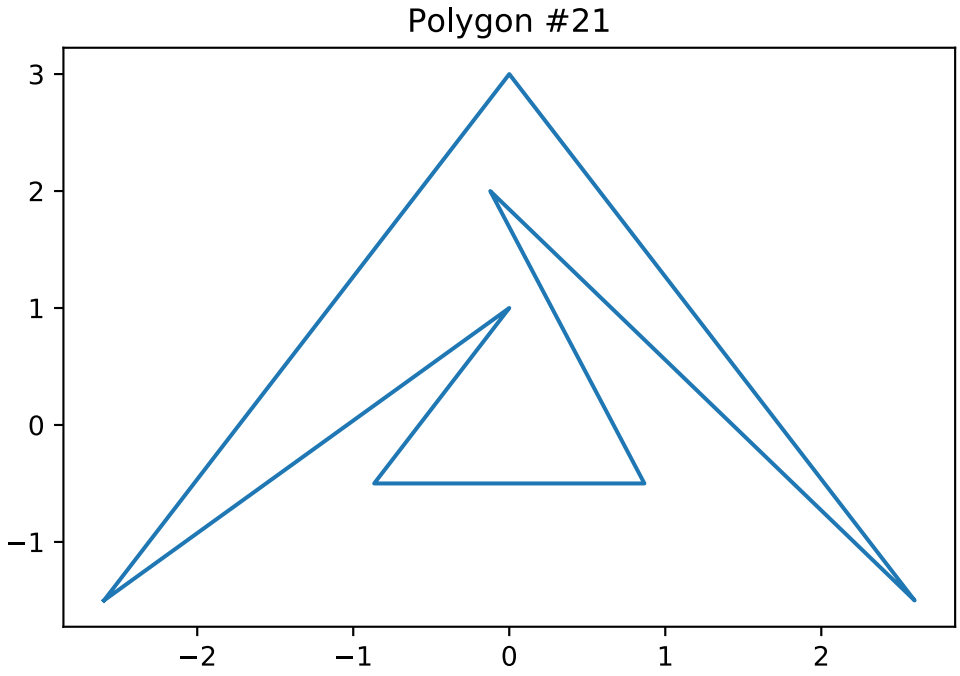

Polygon #16
3
2
1
0
−1
−2
−1
0
1
2

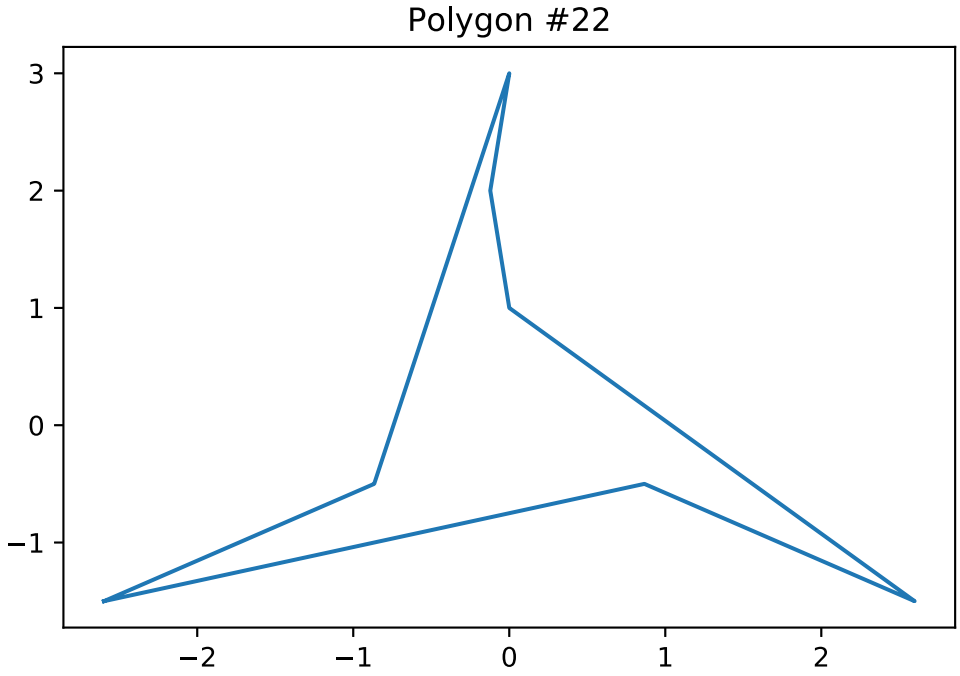

Polygon #17
3
2
1
0
−1
−2
−1
0
1
2

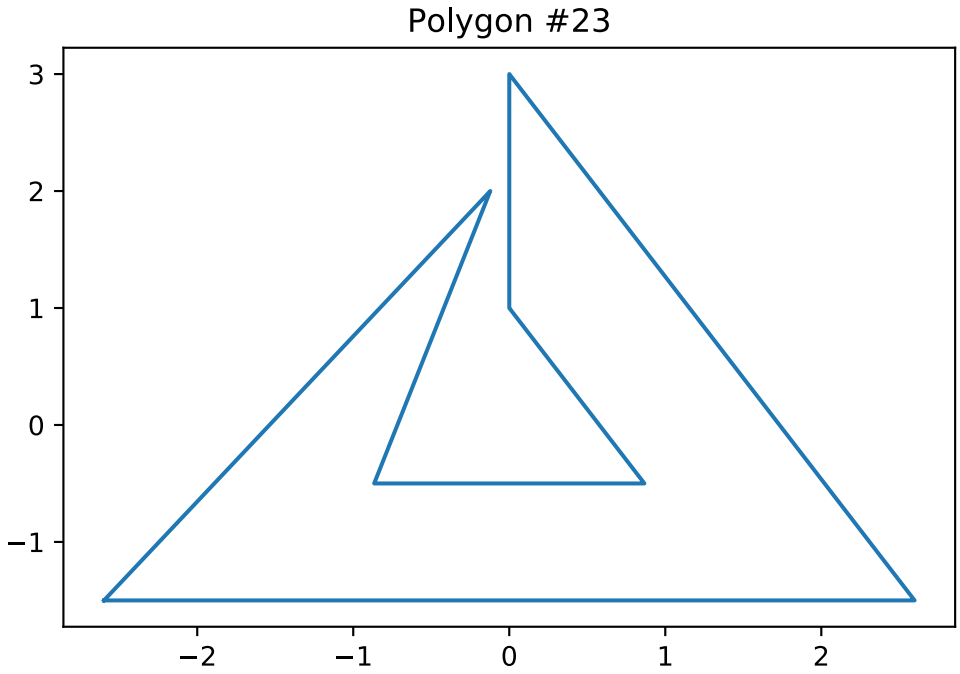

Polygon #18
3
2
1
0
−1
−2
−1
0
1
2

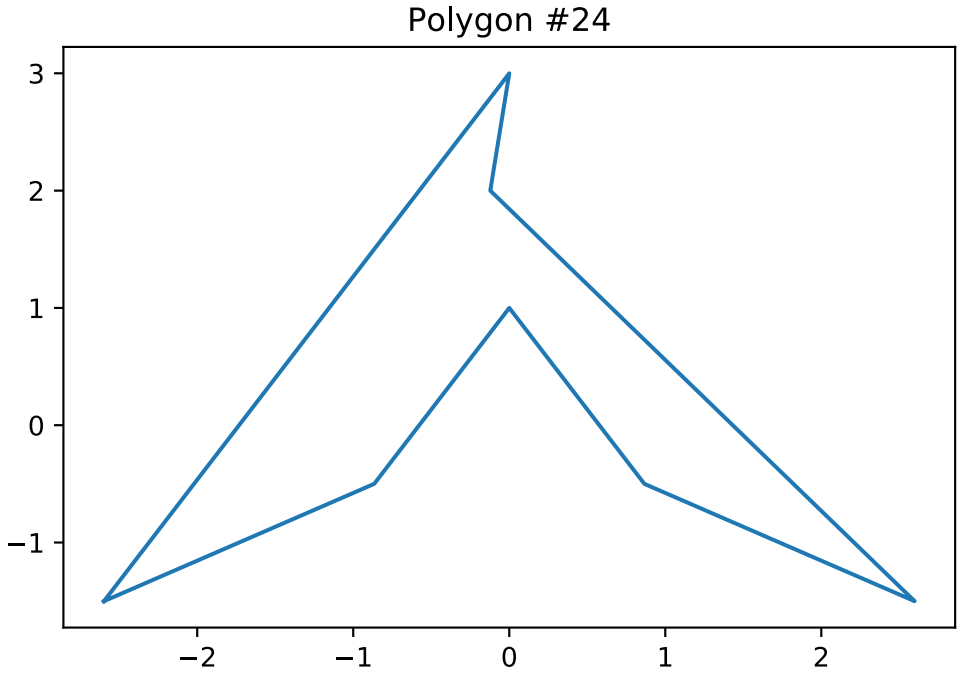

Polygon #19
3
2
1
0
−1
−2
−1
0
1
2

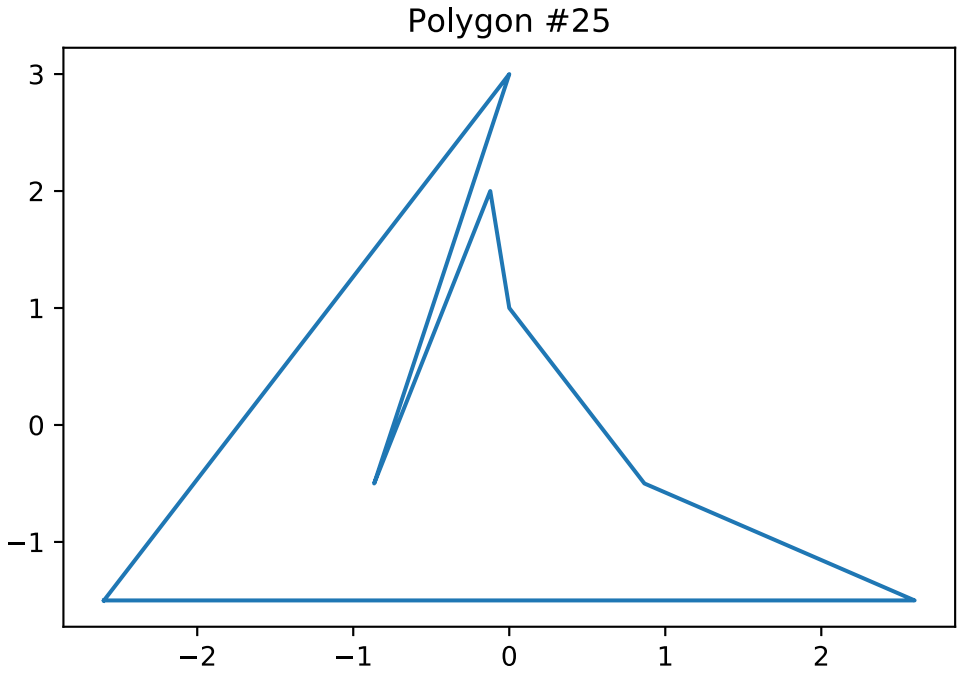

Polygon #20
3
2
1
0
−1
−2
−1
0
1
2

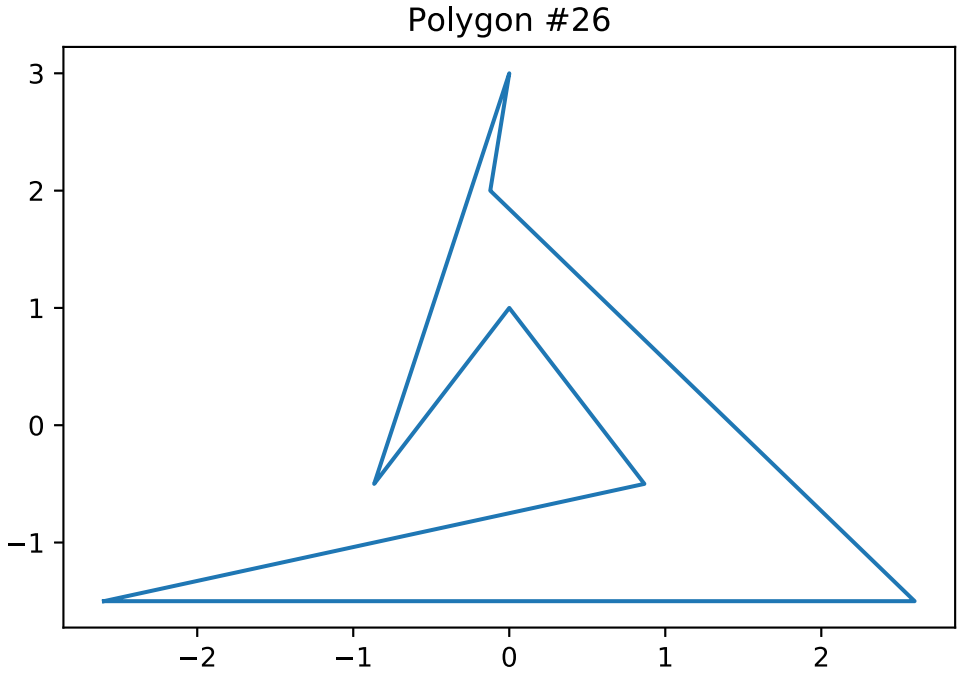

Polygon #21
3
2
1
0
−1
−2
−1
0
1
2

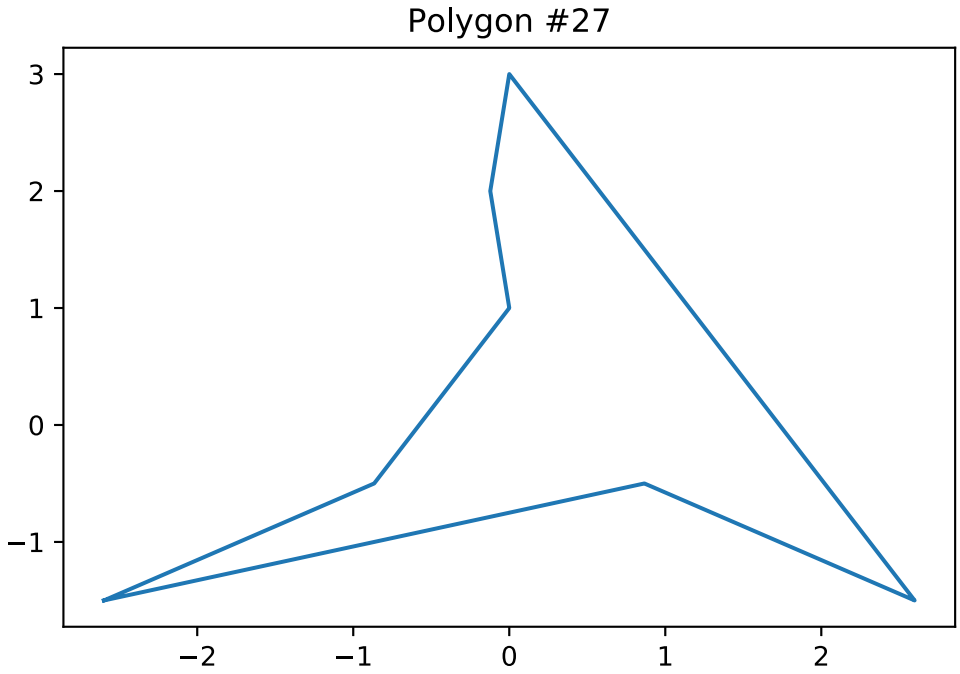

Polygon #22
3
2
1
0
−1
−2
−1
0
1
2

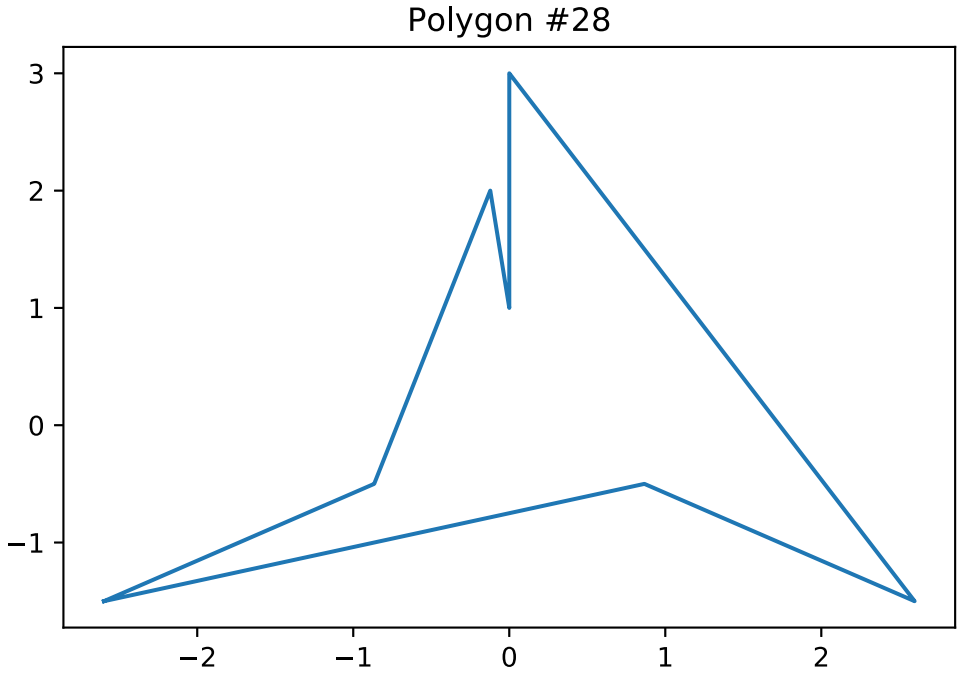

Polygon #23
3
2
1
0
−1
−2
−1
0
1
2

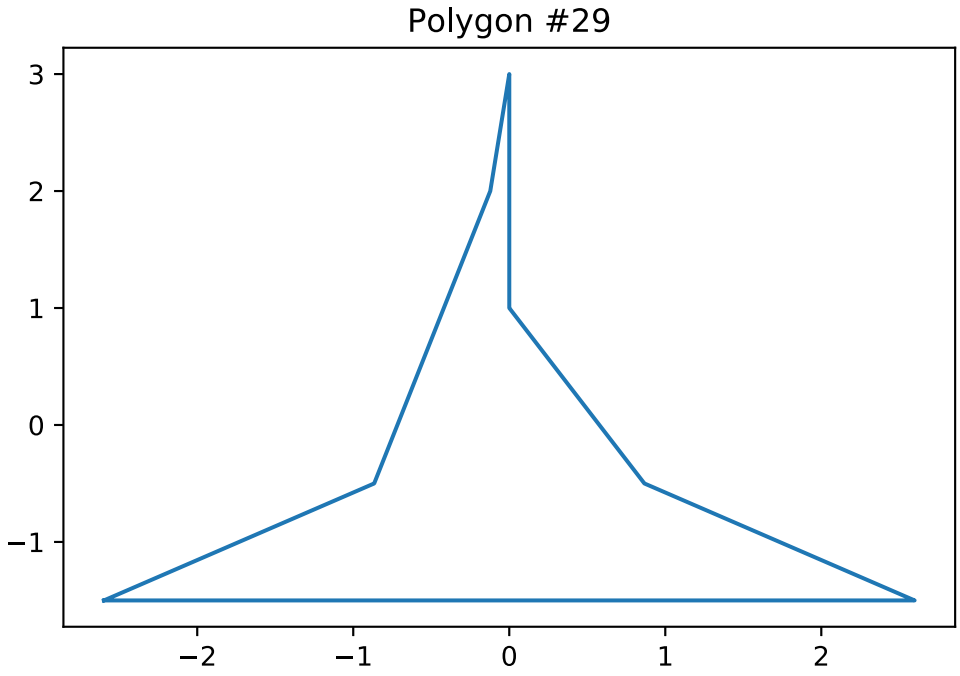

Polygon #24
3
2
1
0
−1
−2
−1
0
1
2

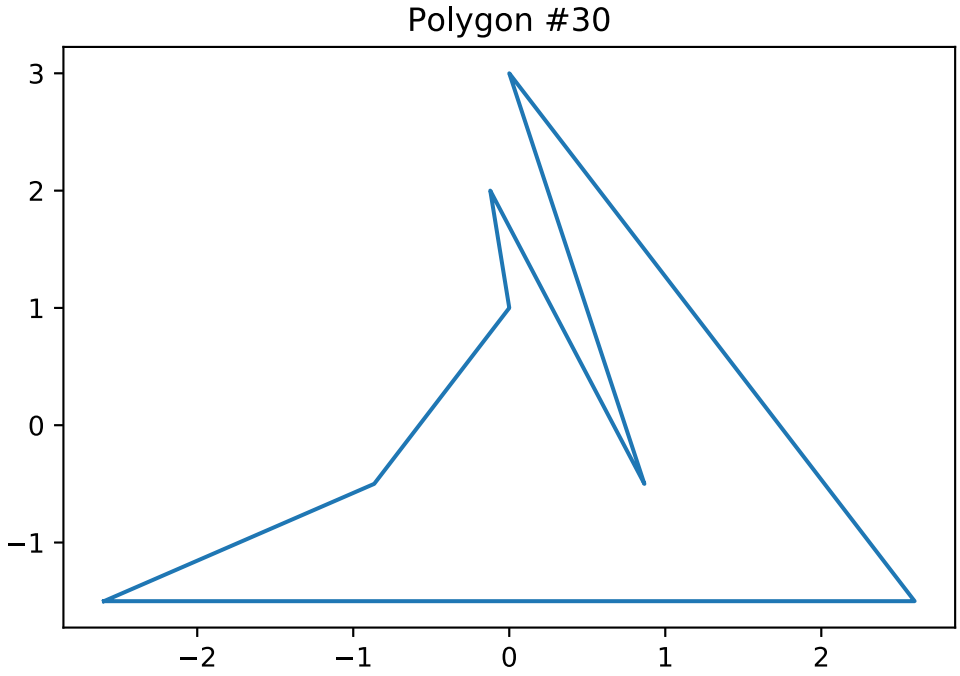

Polygon #25
3
2
1
0
−1
−2
−1
0
1
2

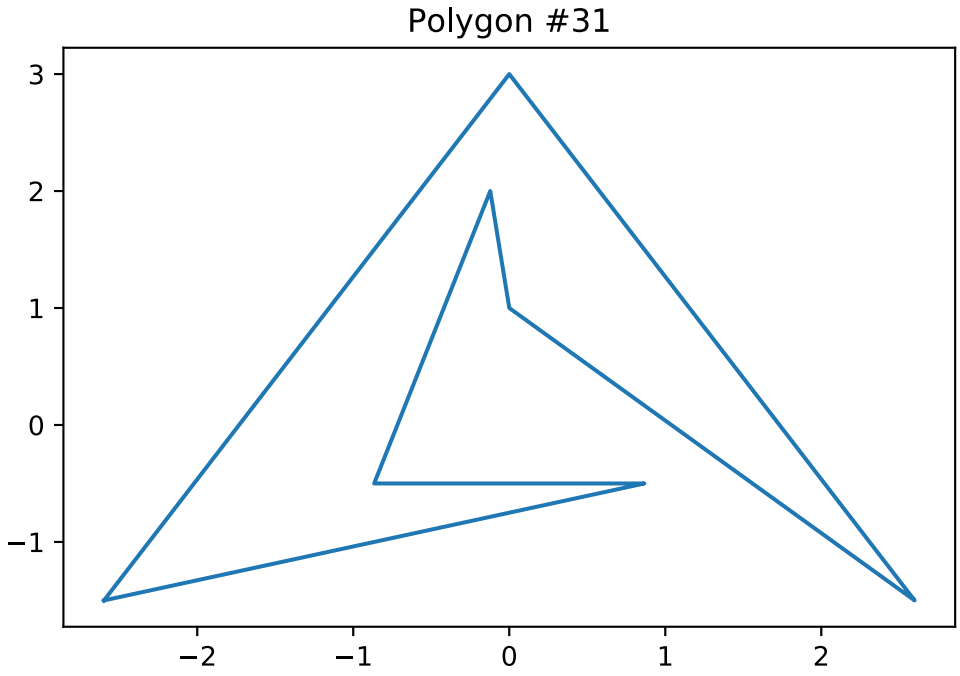

Polygon #26
3
2
1
0
−1
−2
−1
0
1
2

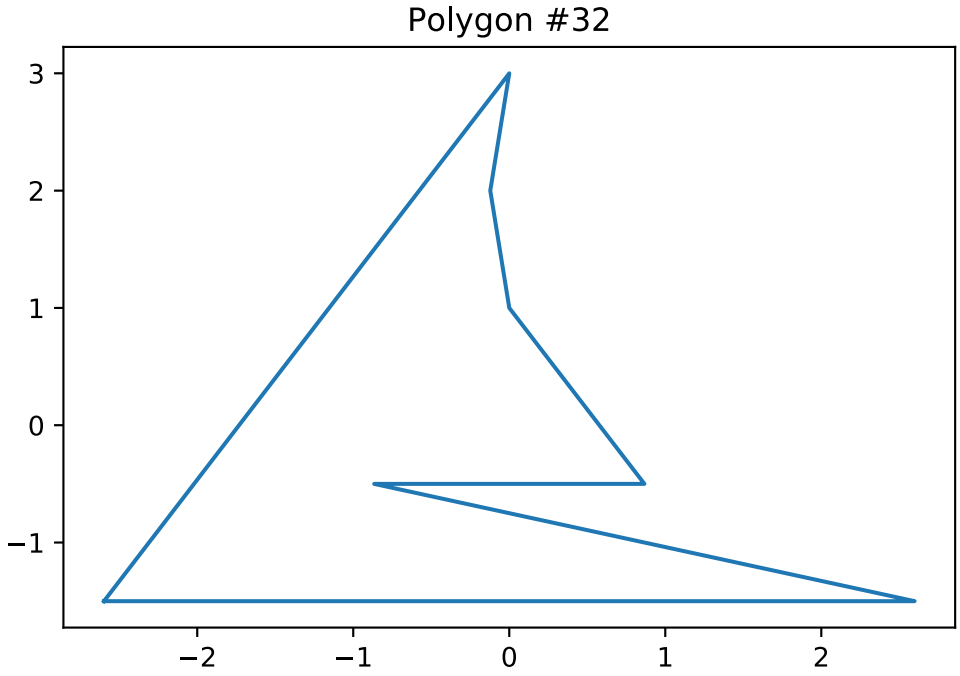

Polygon #27
3
2
1
0
−1
−2
−1
0
1
2

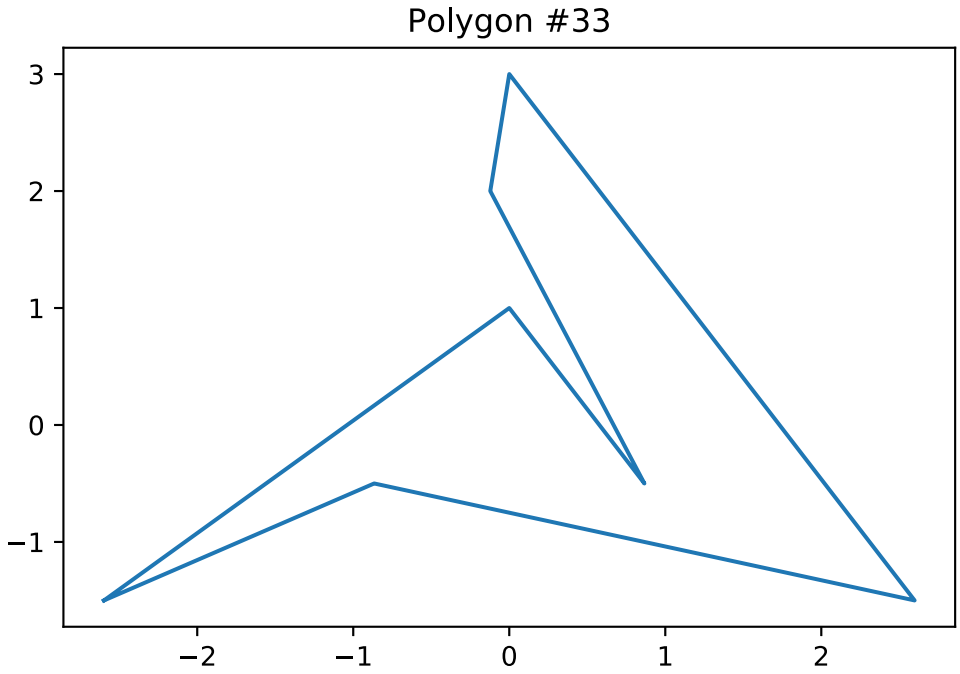

Polygon #28
3
2
1
0
−1
−2
−1
0
1
2

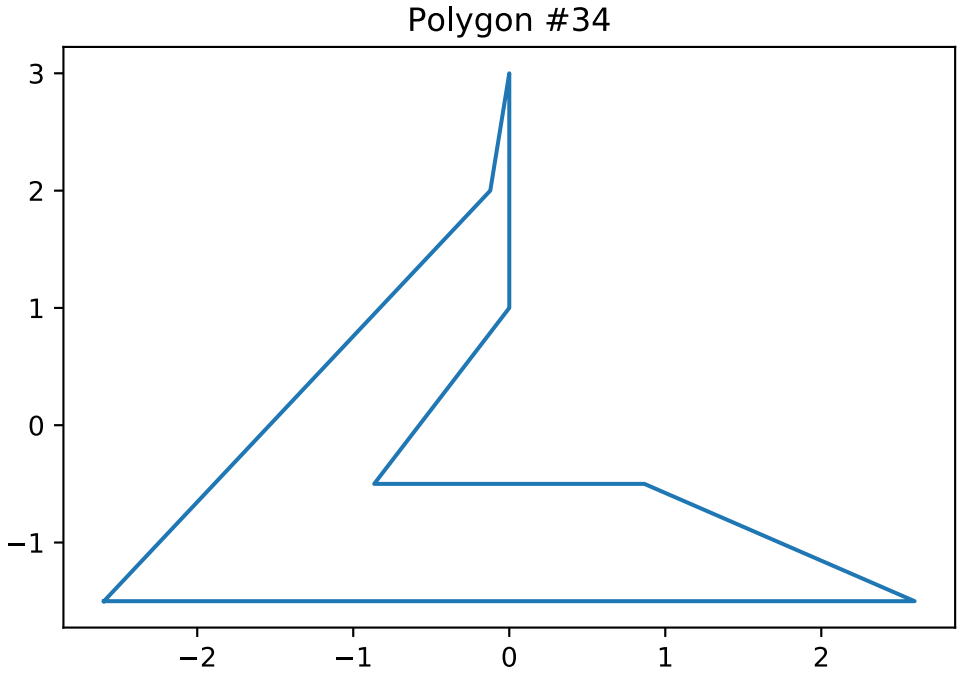

Polygon #29
3
2
1
0
−1
−2
−1
0
1
2

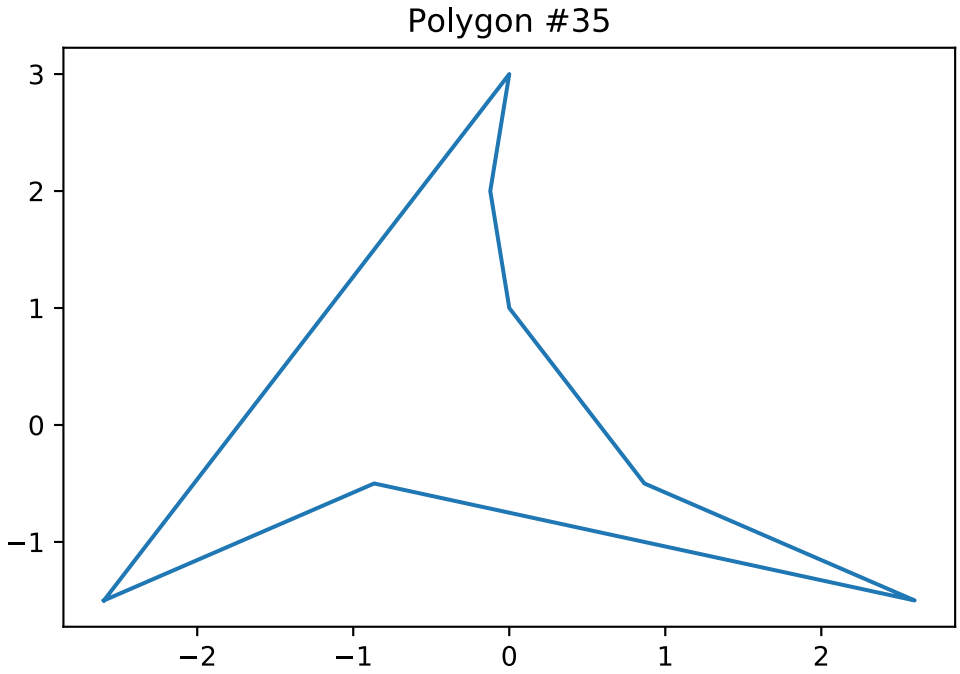

Polygon #30
3
2
1
0
−1
−2
−1
0
1
2

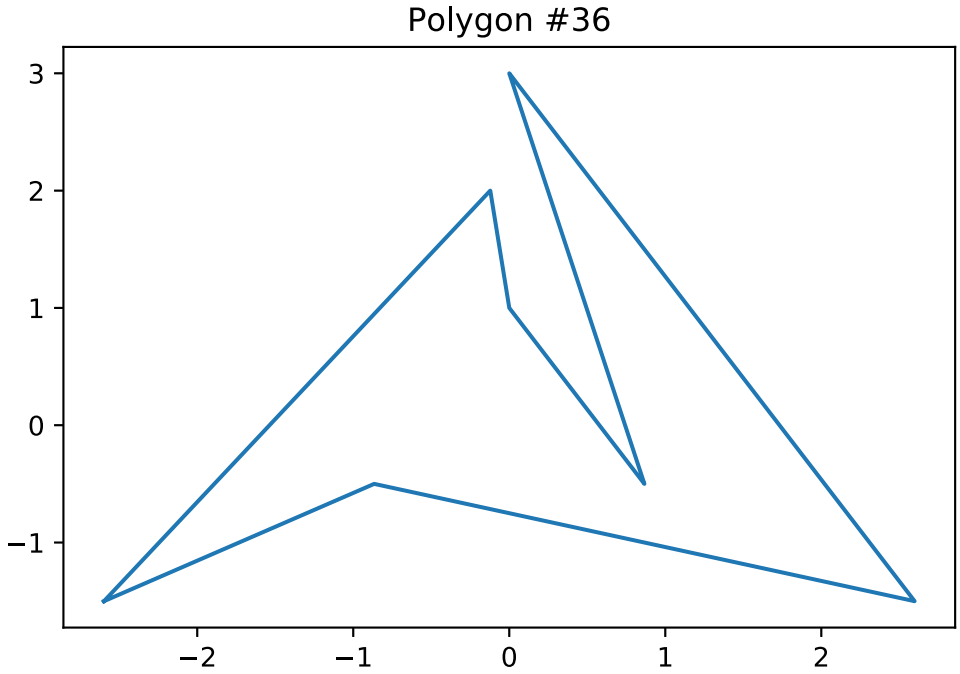

Polygon #31
3
2
1
0
−1
−2
−1
0
1
2

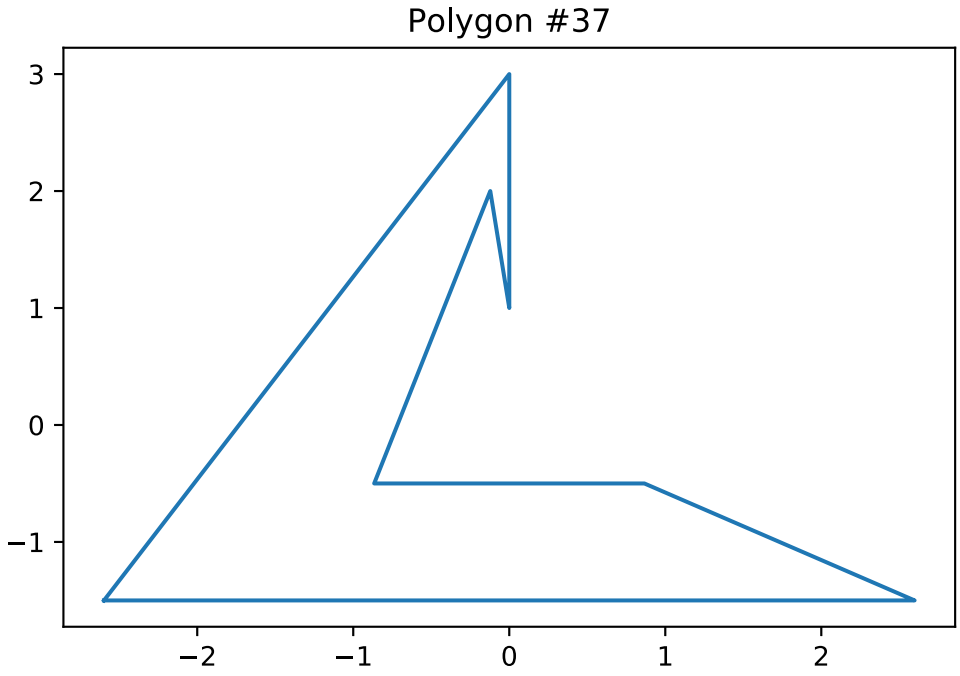

Polygon #32
3
2
1
0
−1
−2
−1
0
1
2

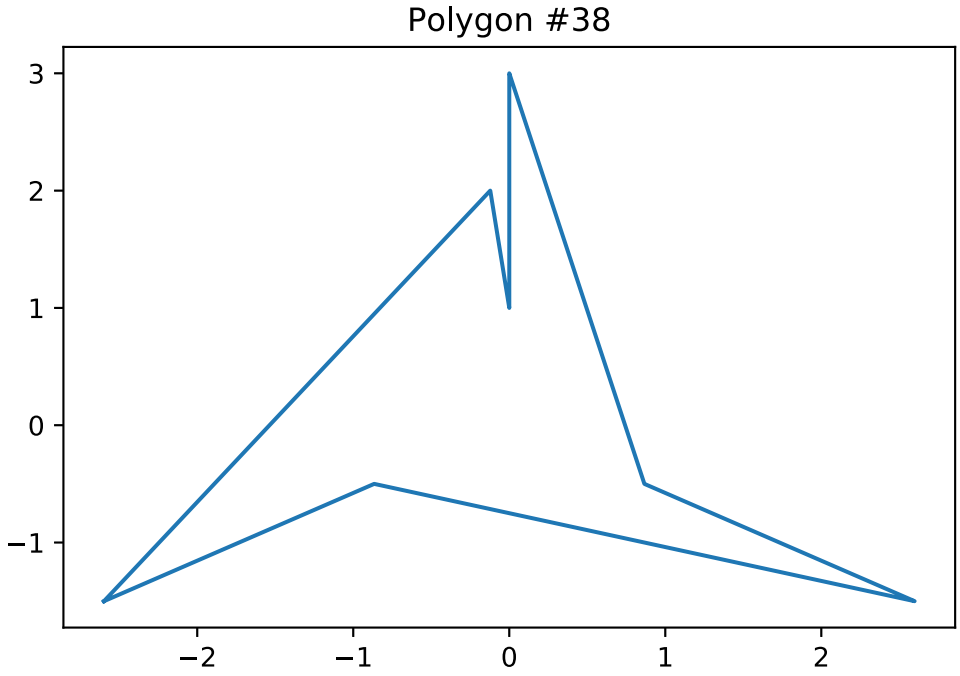

Polygon #33
3
2
1
0
−1
−2
−1
0
1
2

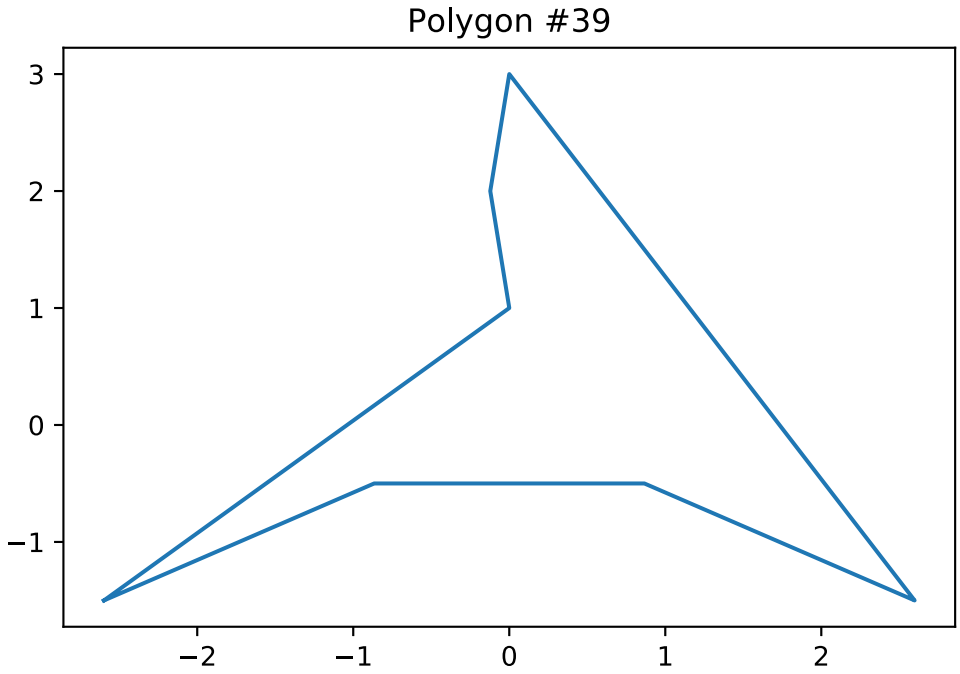

Polygon #34
3
2
1
0
−1
−2
−1
0
1
2

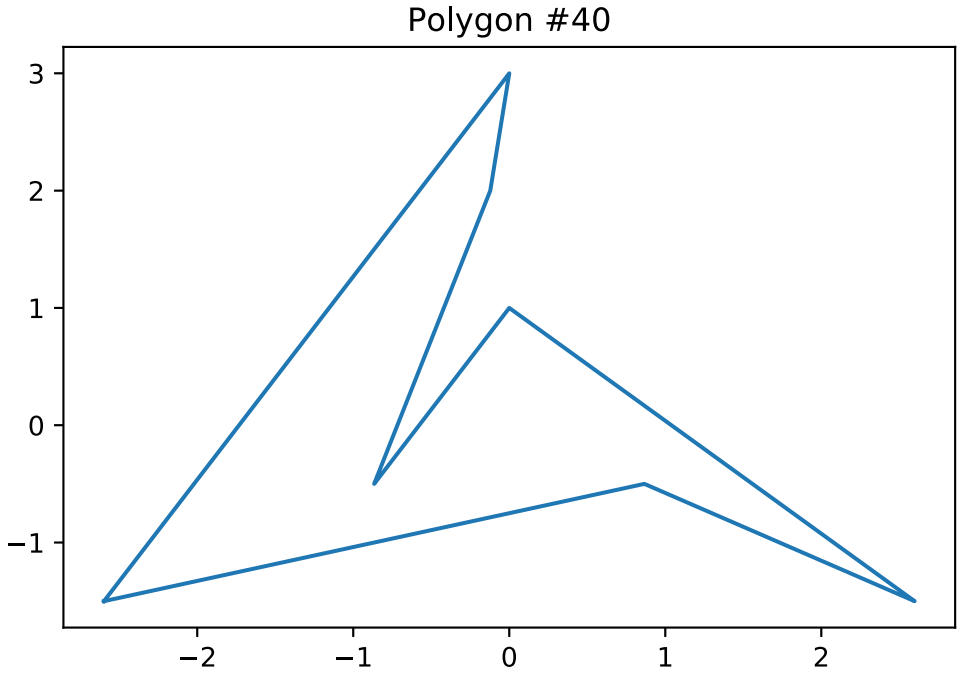

Polygon #35
3
2
1
0
−1
−2
−1
0
1
2

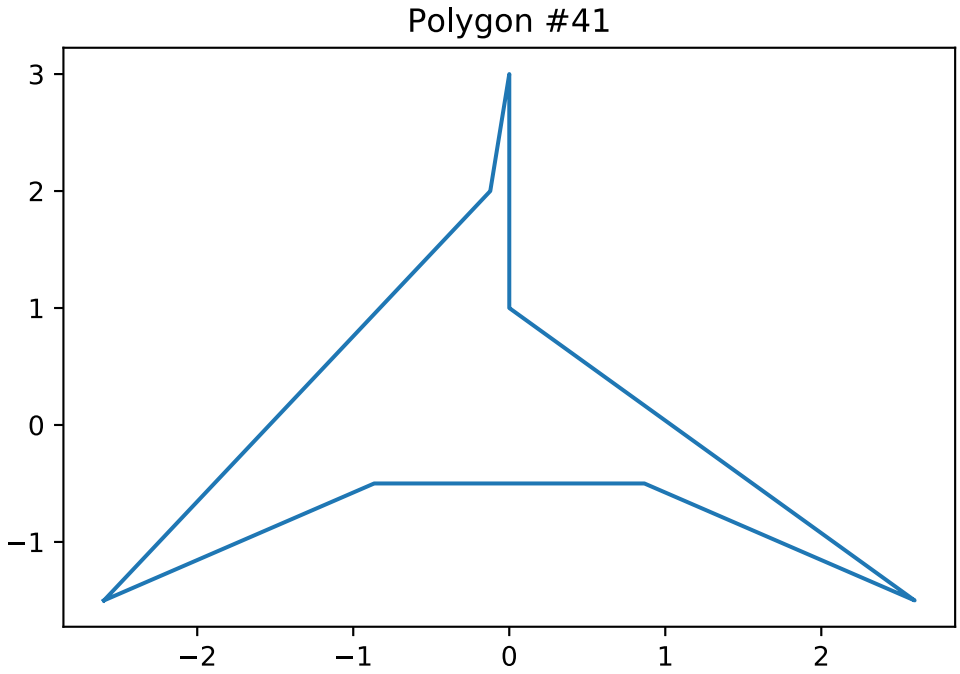

Polygon #36
3
2
1
0
−1
−2
−1
0
1
2

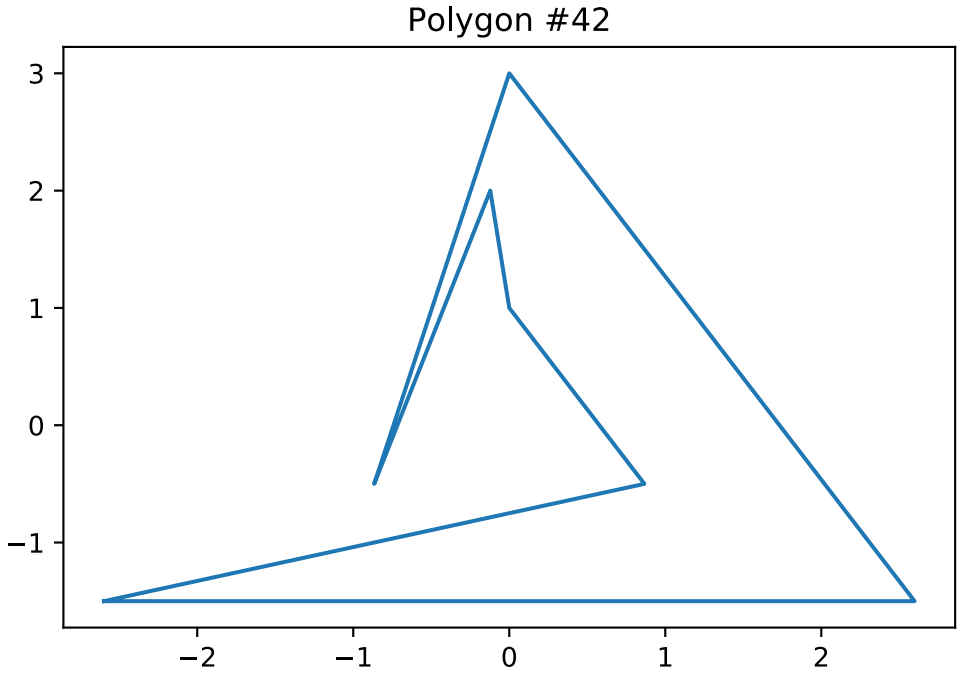

Polygon #37
3
2
1
0
−1
−2
−1
0
1
2

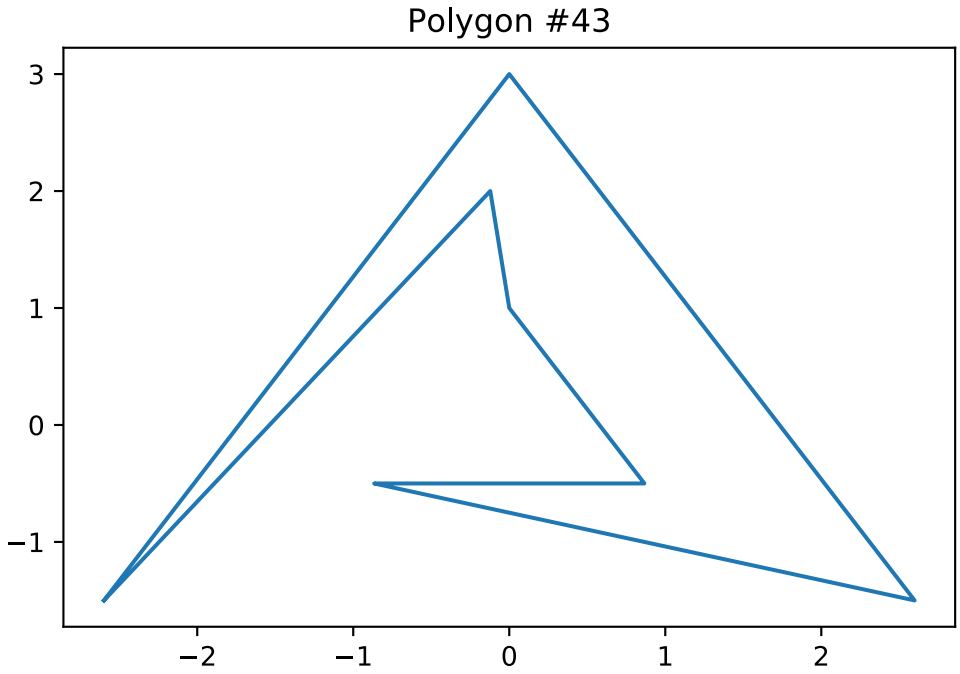

Polygon #38
3
2
1
0
−1
−2
−1
0
1
2

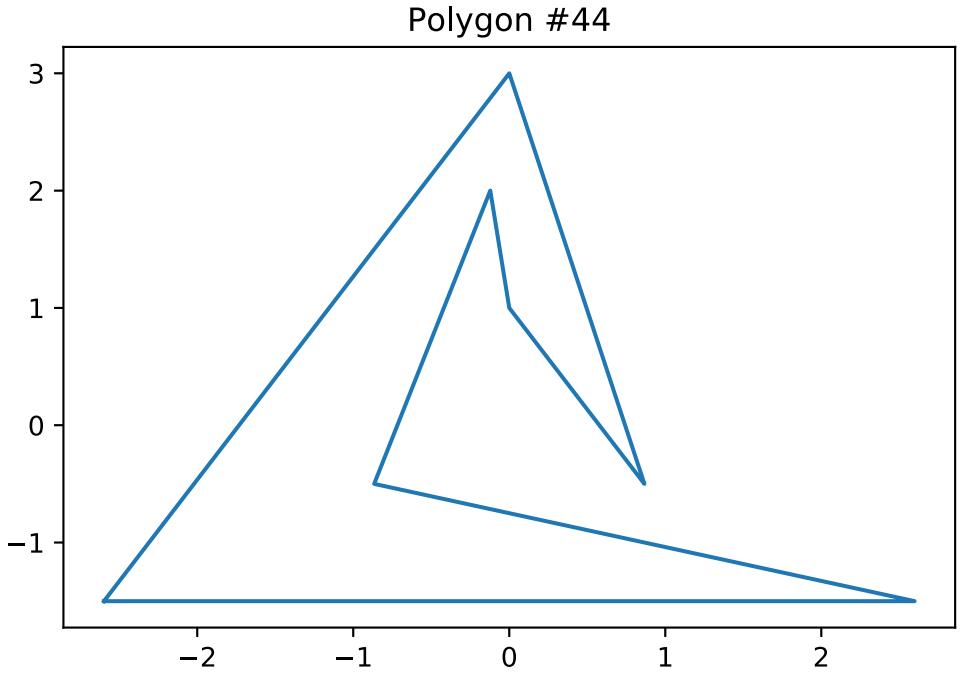

Polygon #39
3
2
1
0
−1
−2
−1
0
1
2

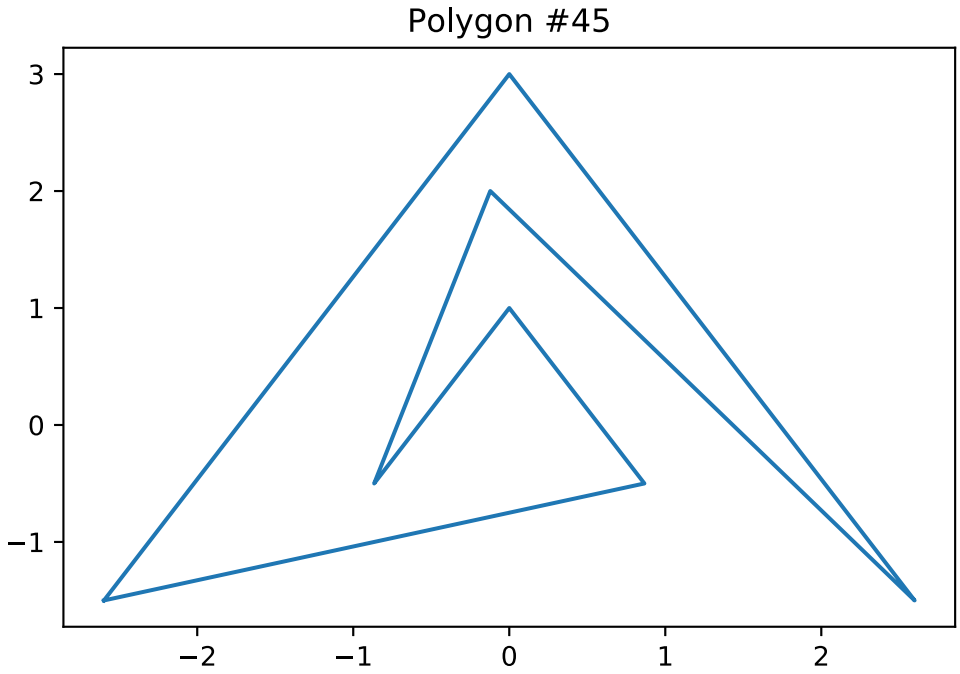

Polygon #40
3
2
1
0
−1
−2
−1
0
1
2

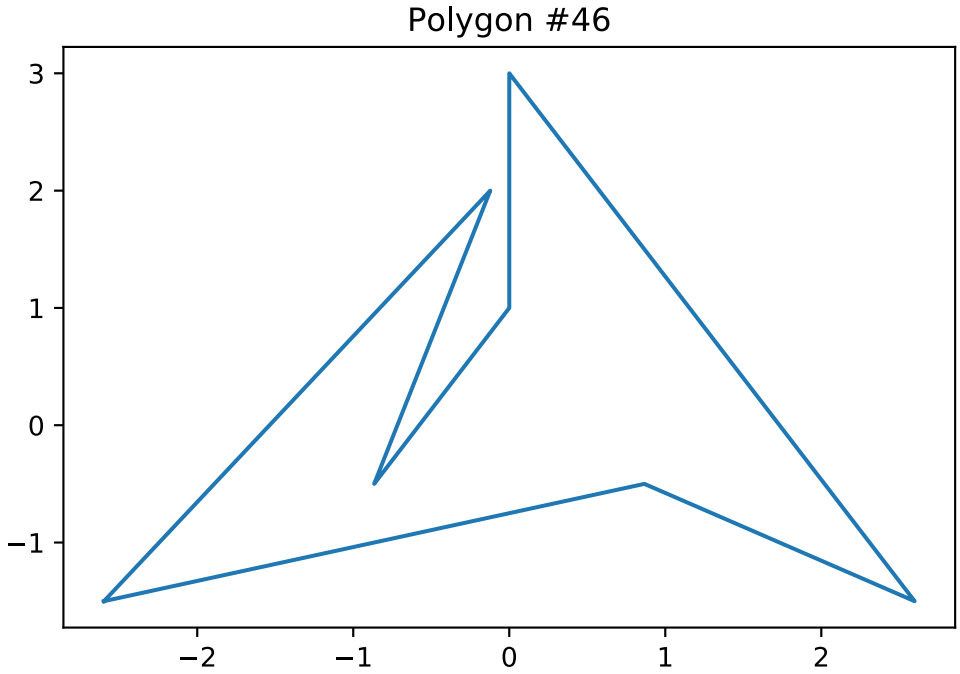

Polygon #41
3
2
1
0
−1
−2
−1
0
1
2

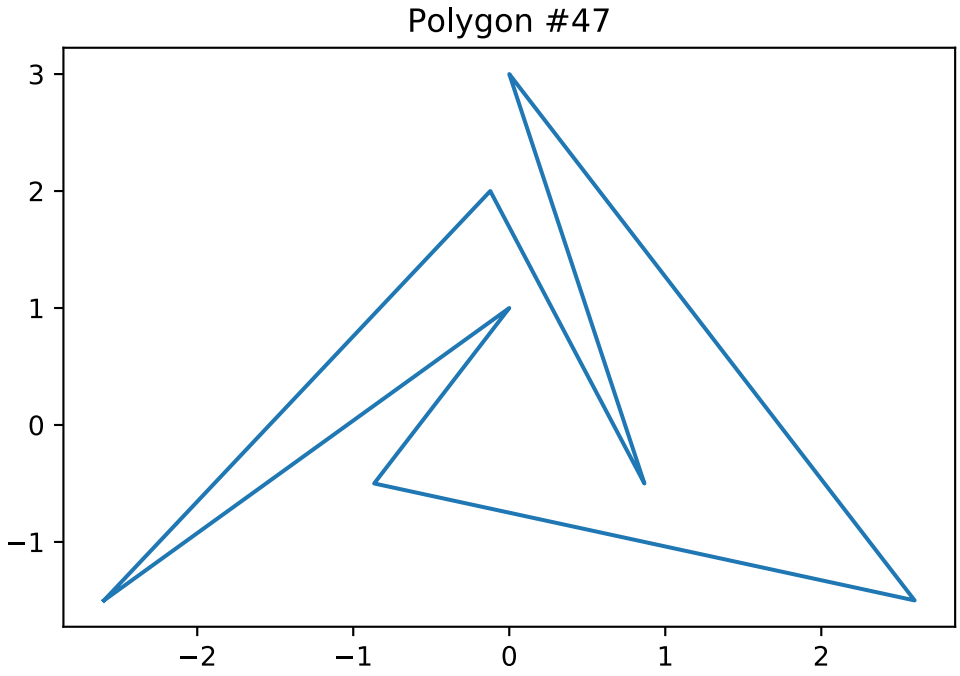

Polygon #42
3
2
1
0
−1
−2
−1
0
1
2

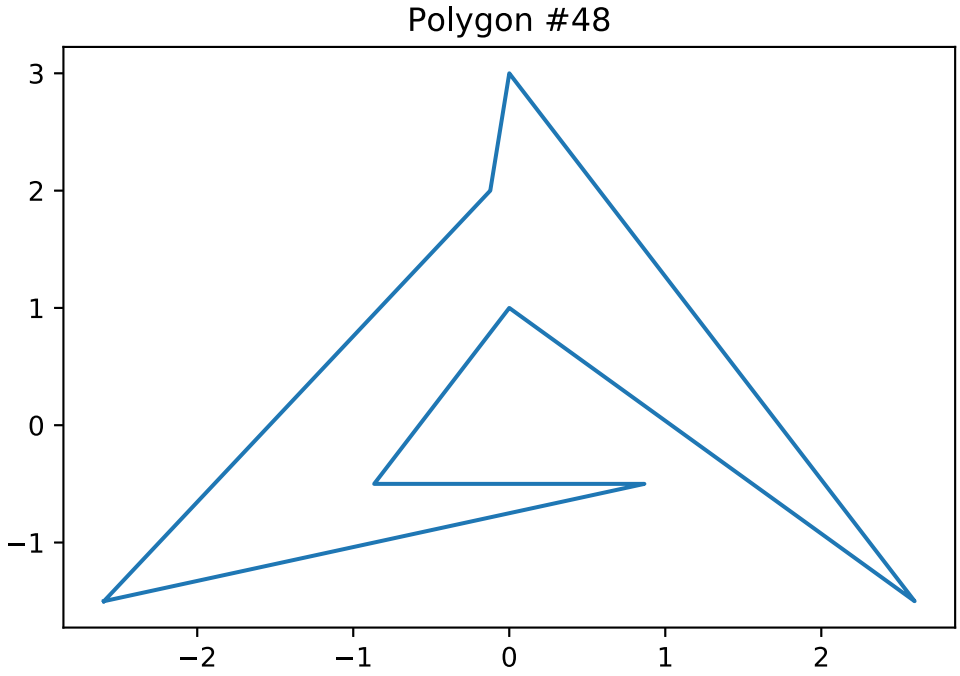

Polygon #43
3
2
1
0
−1
−2
−1
0
1
2

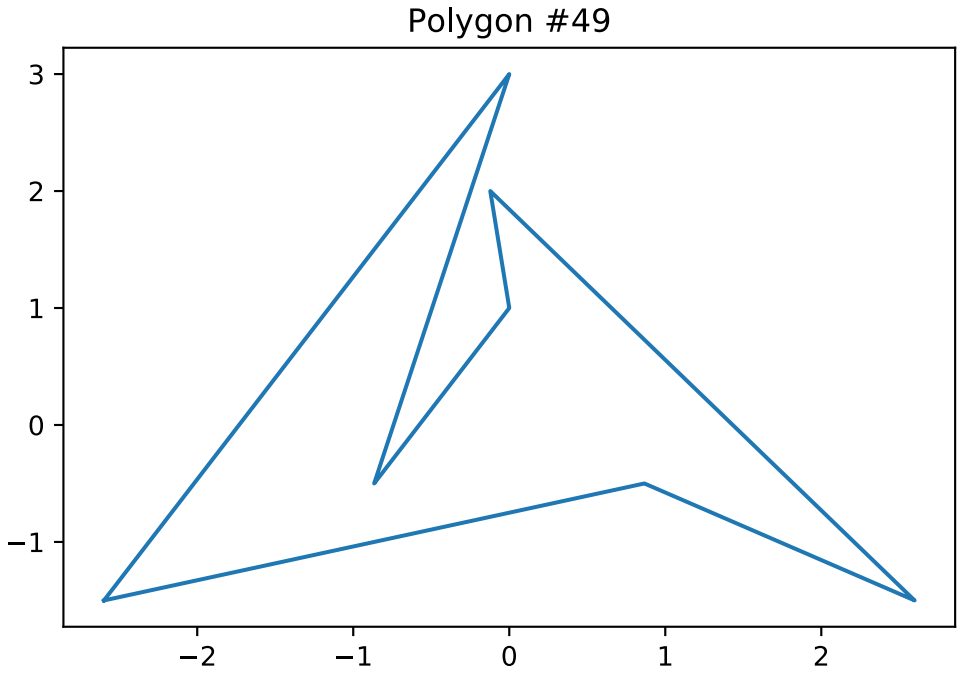

Polygon #44
3
2
1
0
−1
−2
−1
0
1
2

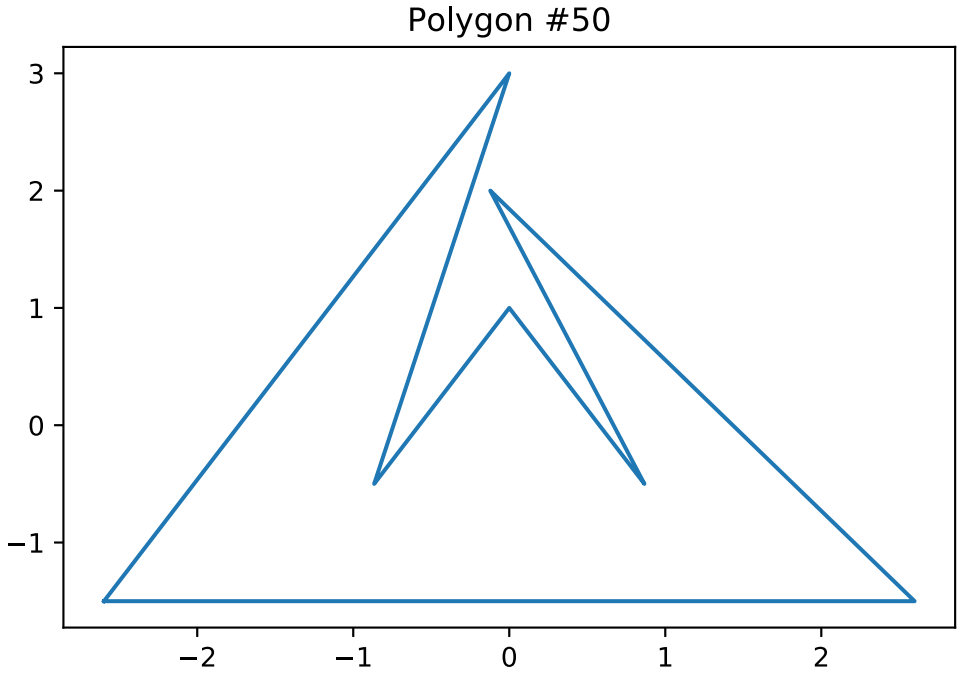

Polygon #45
3
2
1
0
−1
−2
−1
0
1
2

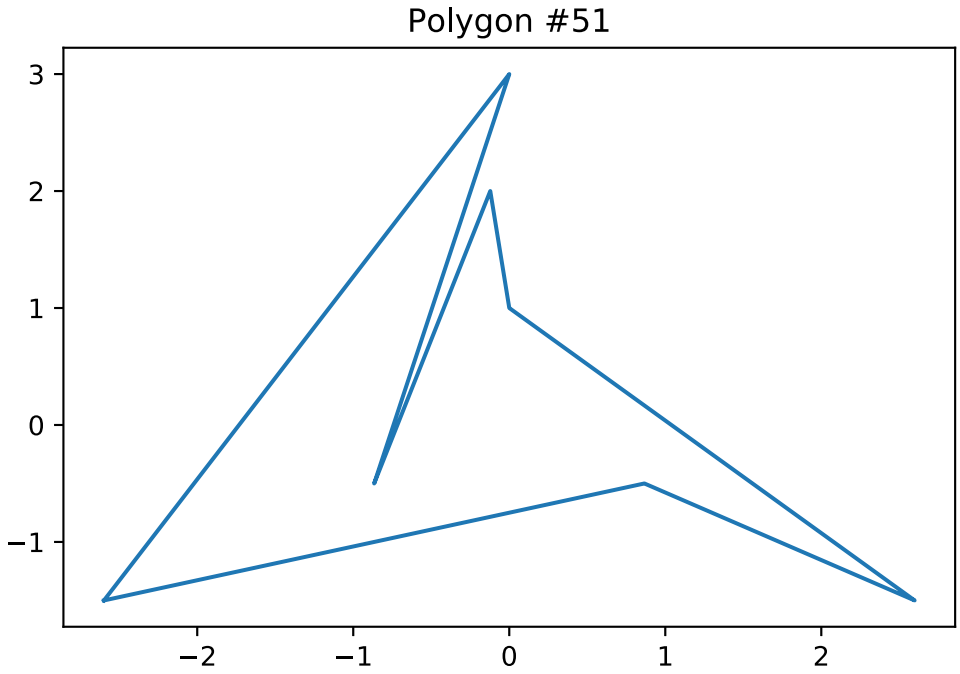

Polygon #46
3
2
1
0
−1
−2
−1
0
1
2

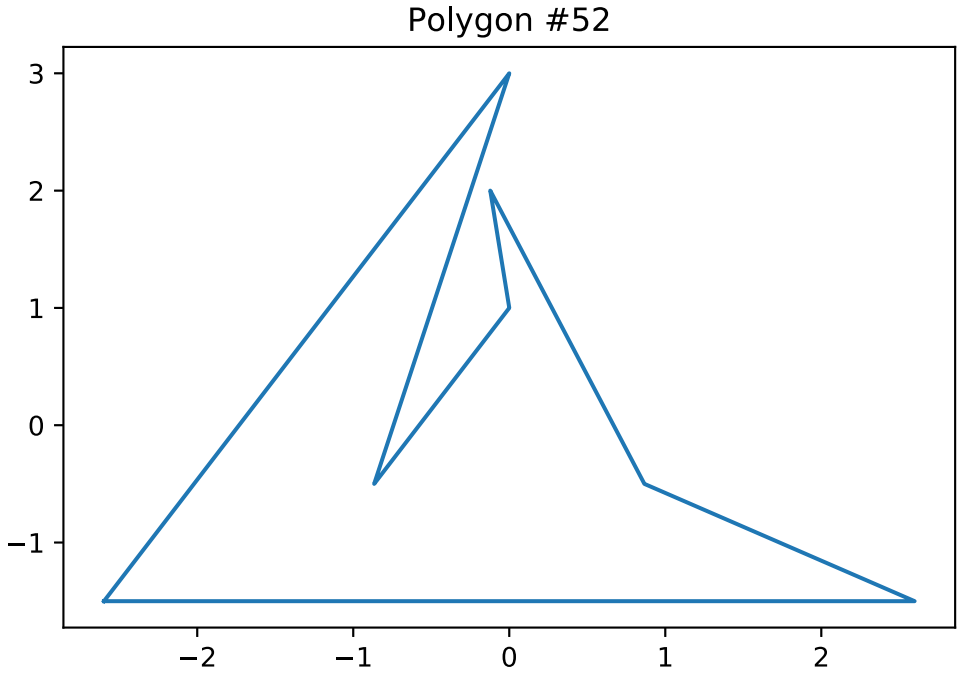

Polygon #47
3
2
1
0
−1
−2
−1
0
1
2

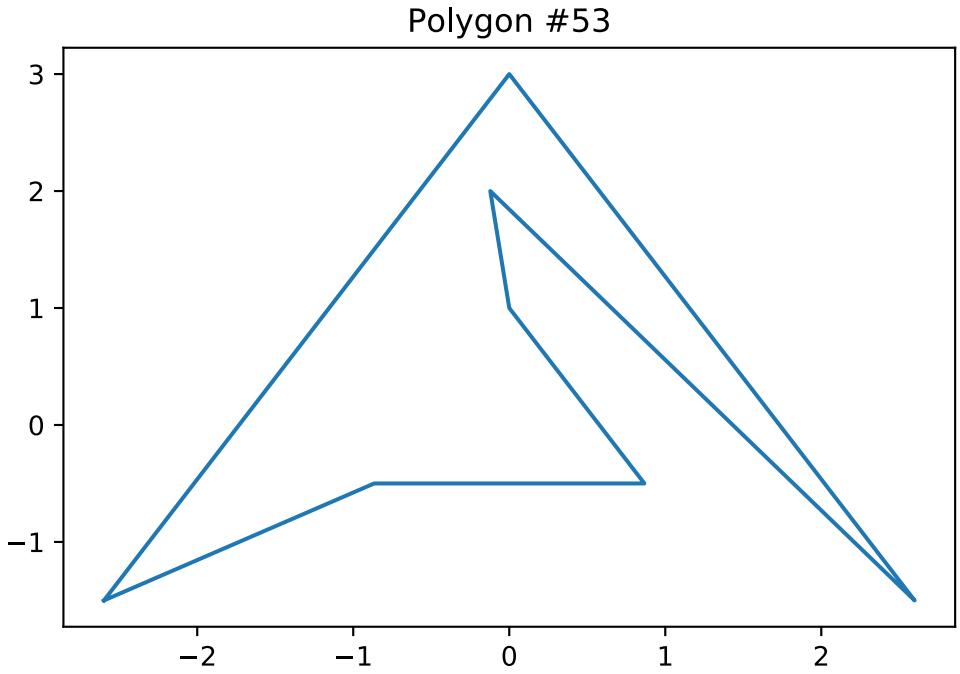

Polygon #48
3
2
1
0
−1
−2
−1
0
1
2

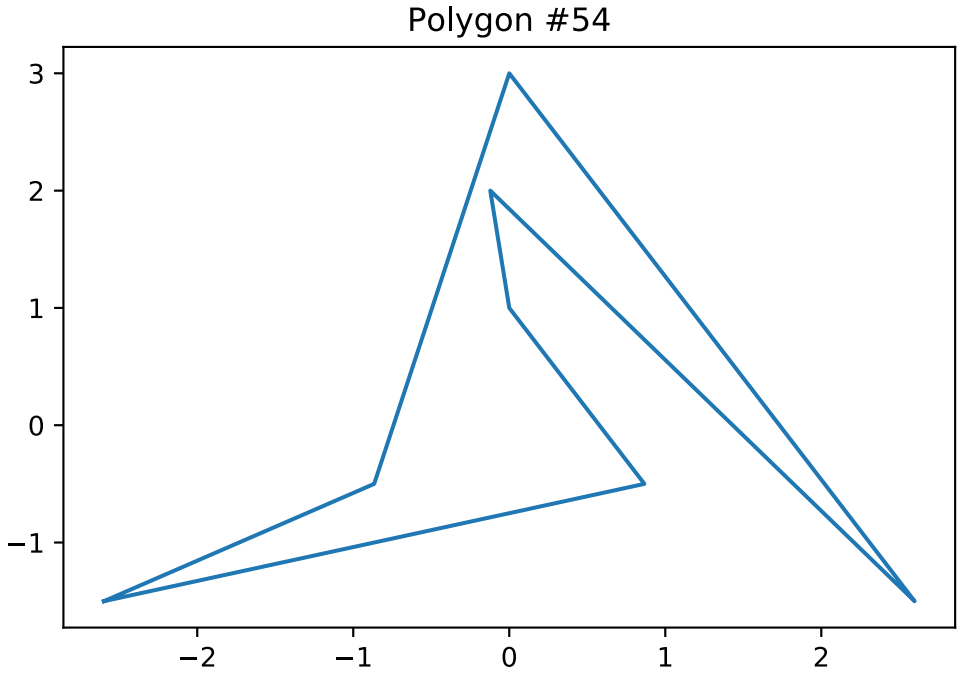

Polygon #49
3
2
1
0
−1
−2
−1
0
1
2

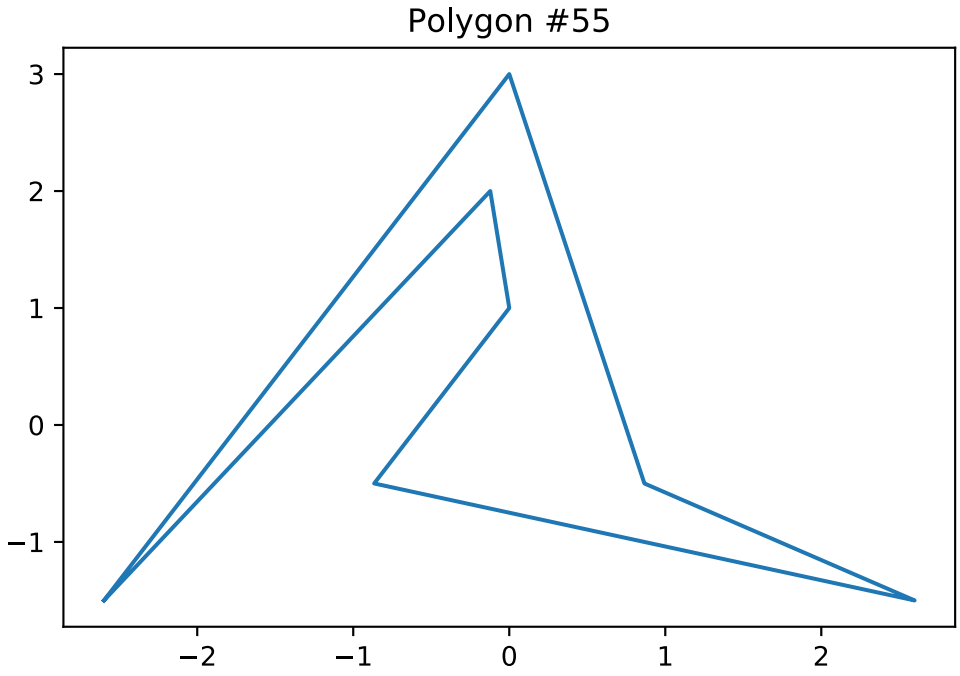

Polygon #50
3
2
1
0
−1
−2
−1
0
1
2

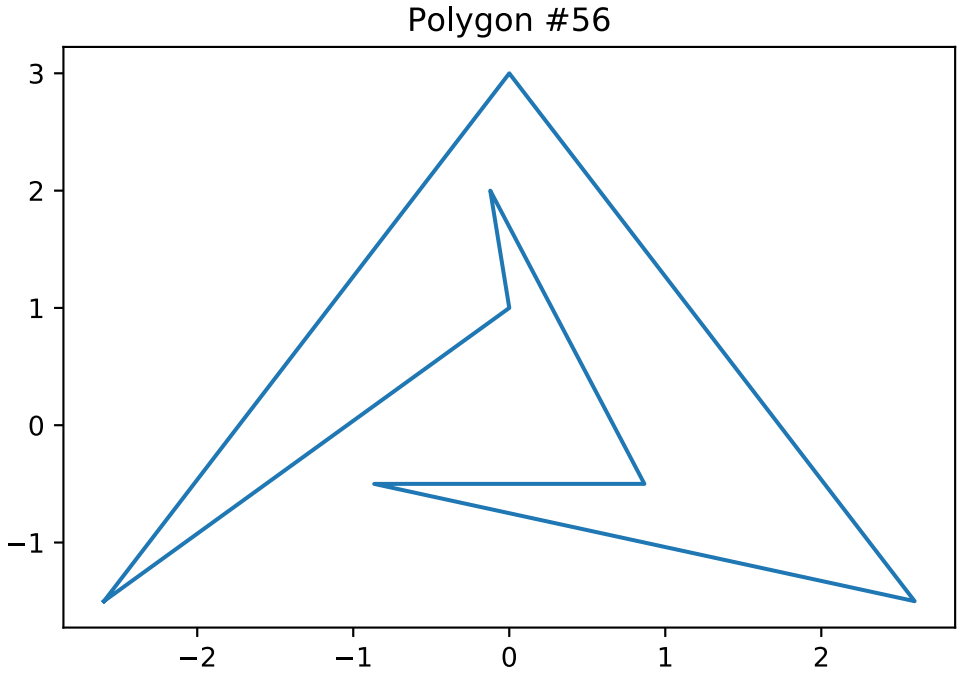

Polygon #51
3
2
1
0
−1
−2
−1
0
1
2

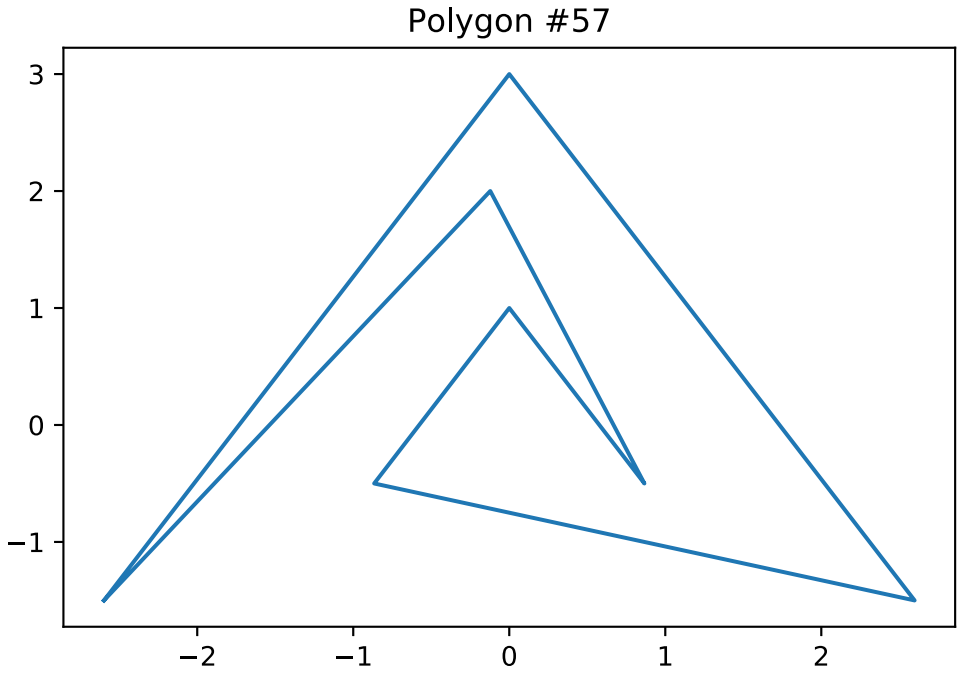

Polygon #52
3
2
1
0
−1
−2
−1
0
1
2

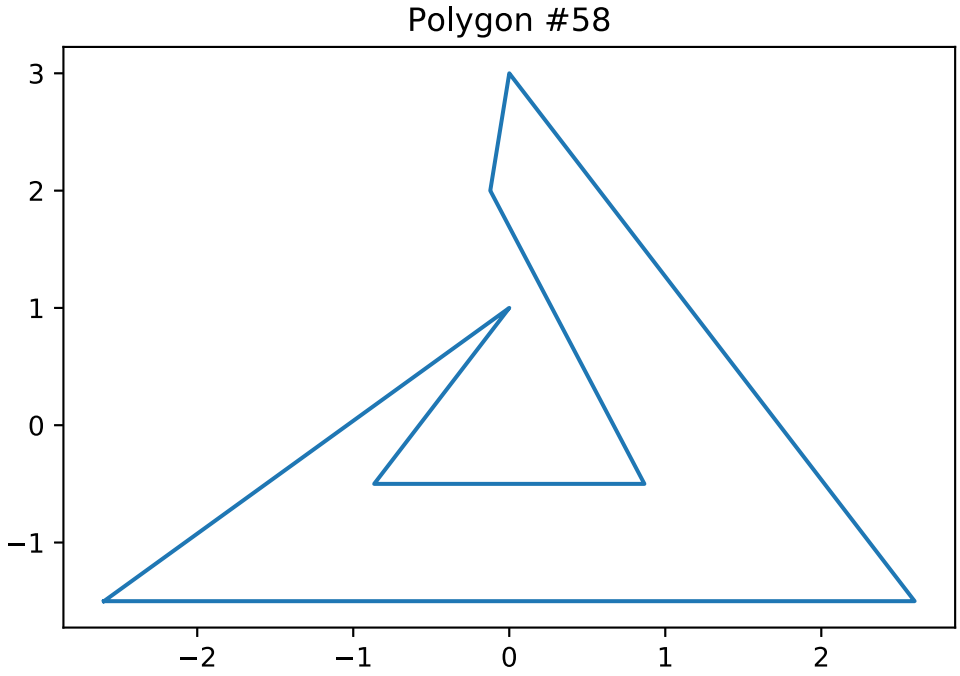

Polygon #53
3
2
1
0
−1
−2
−1
0
1
2

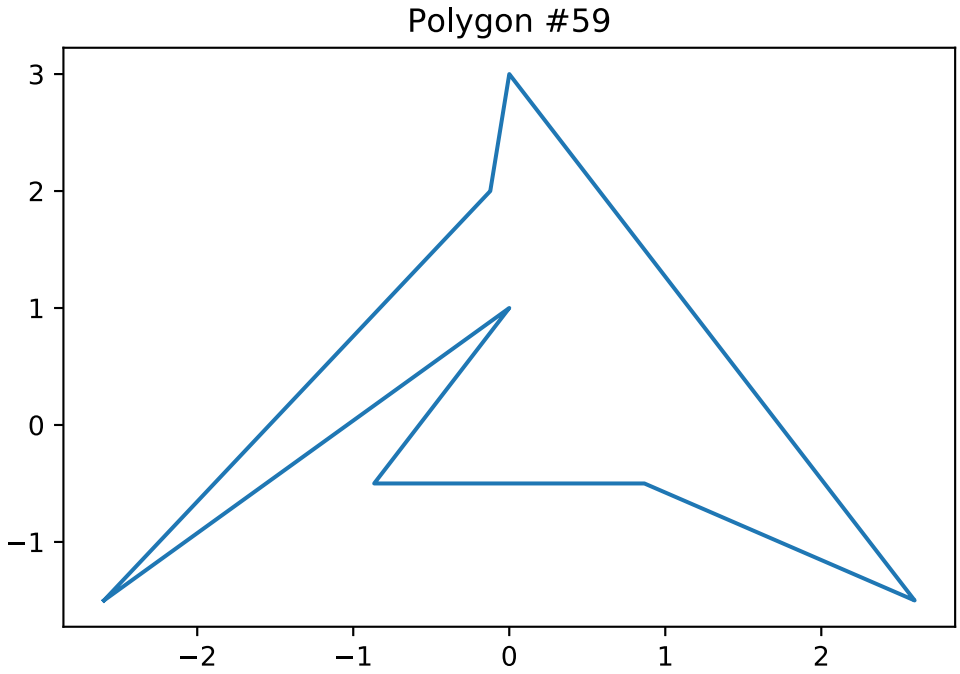

Polygon #54
3
2
1
0
−1
−2
−1
0
1
2

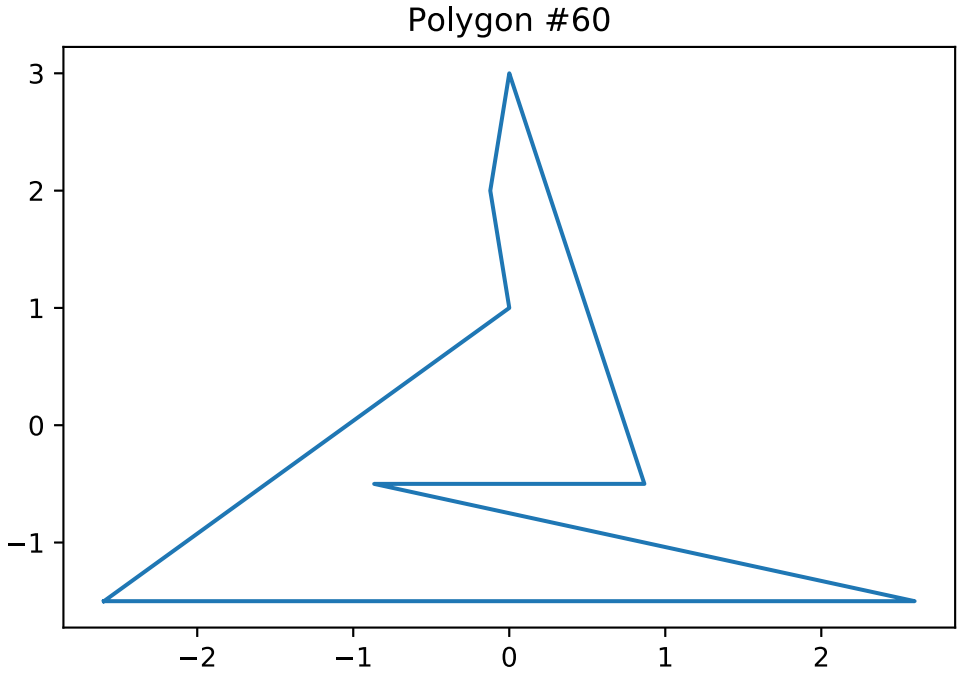

Polygon #55
3
2
1
0
−1
−2
−1
0
1
2

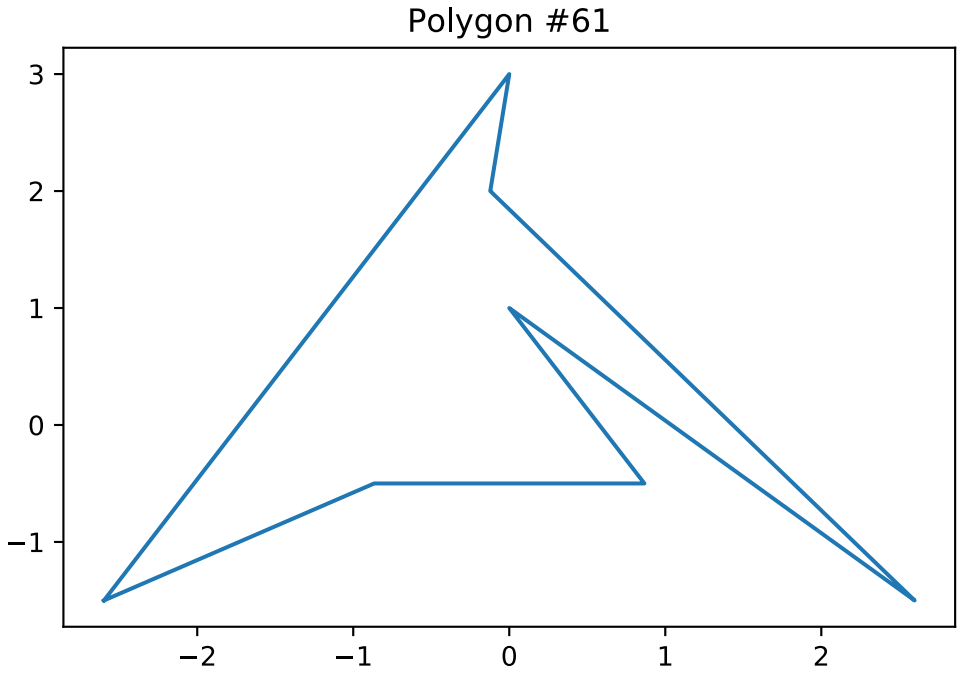

Polygon #56
3
2
1
0
−1
−2
−1
0
1
2

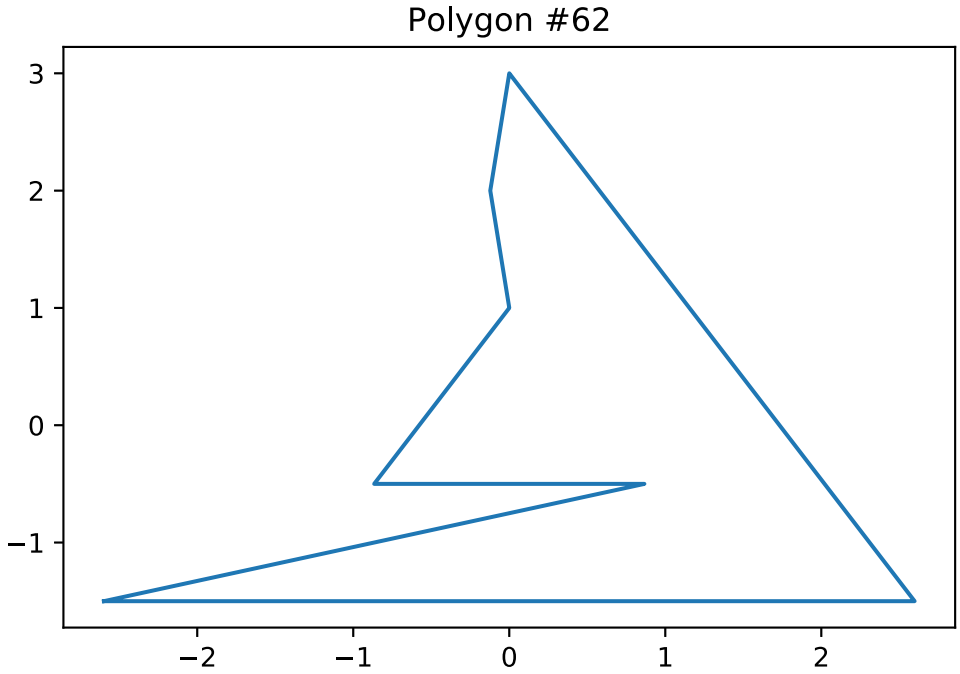

Polygon #57
3
2
1
0
−1
−2
−1
0
1
2

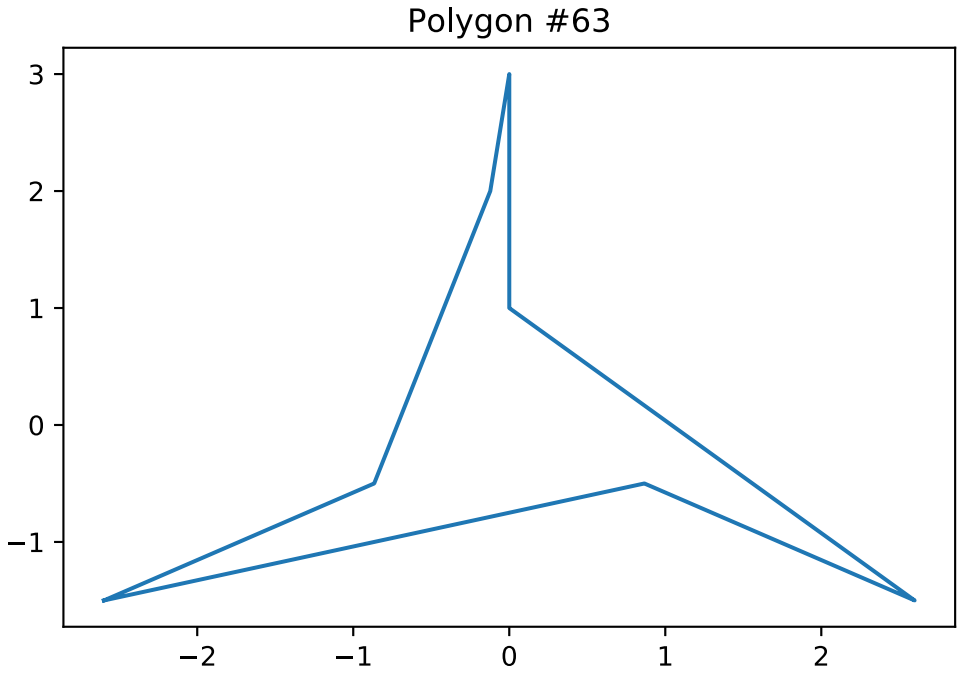

Polygon #58
3
2
1
0
−1
−2
−1
0
1
2

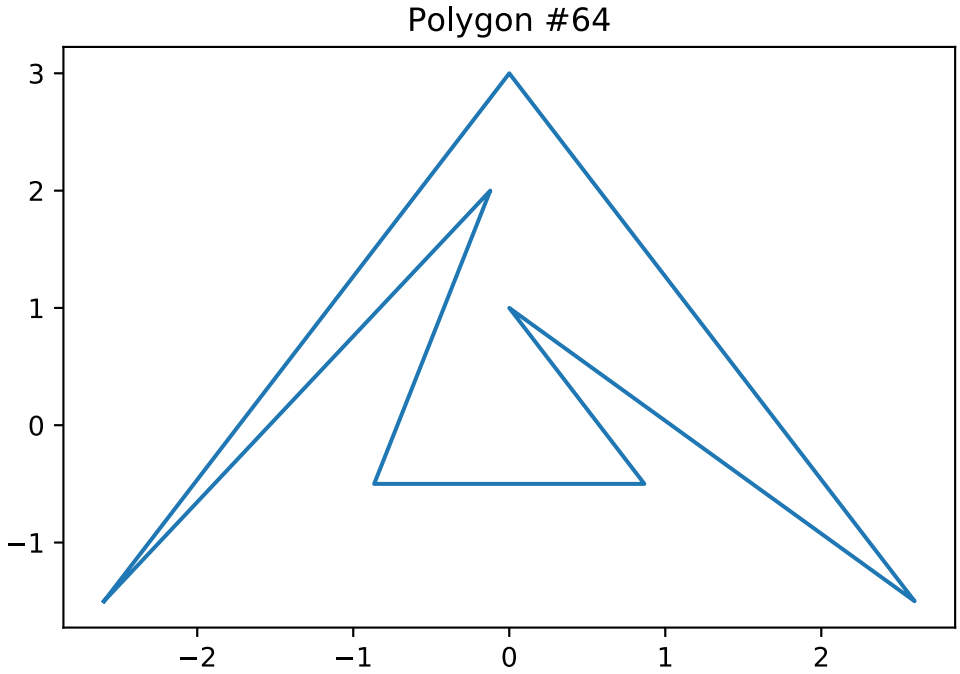

Polygon #59
3
2
1
0
−1
−2
−1
0
1
2

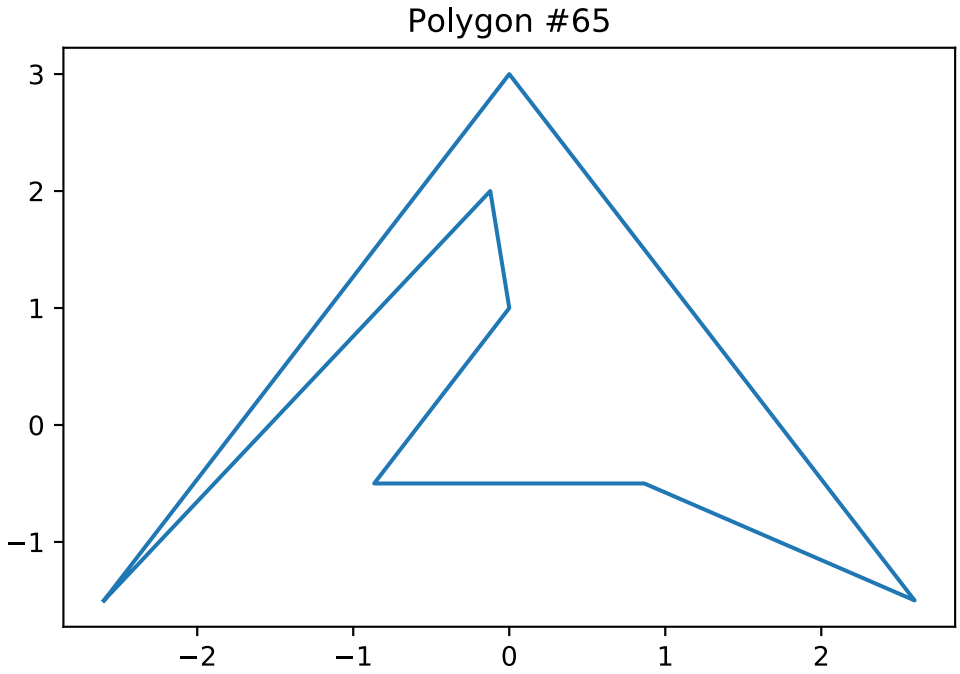

Polygon #60
3
2
1
0
−1
−2
−1
0
1
2

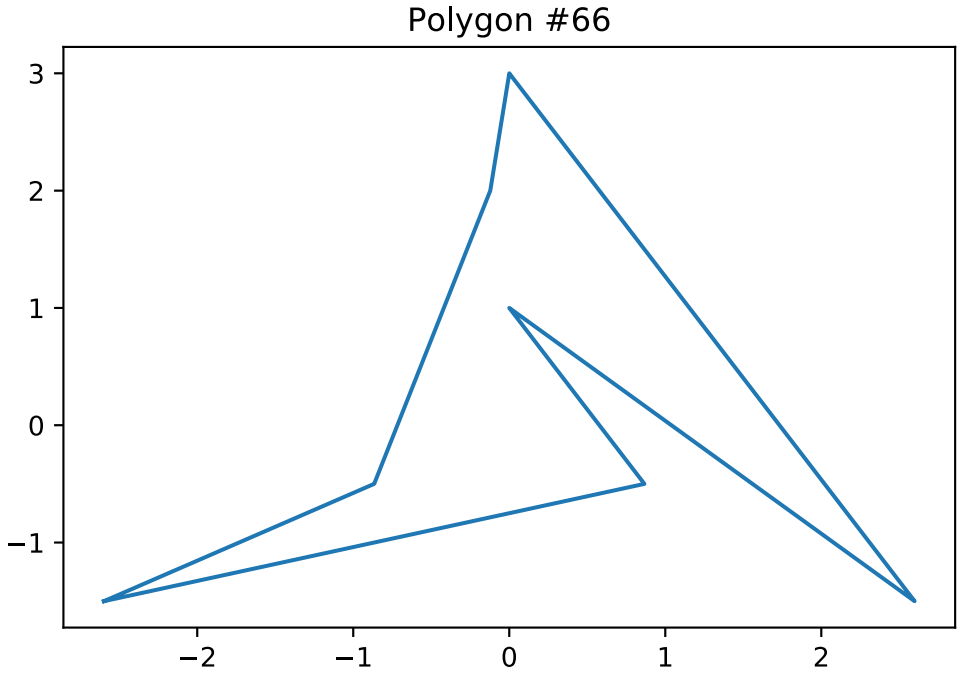

Polygon #61
3
2
1
0
−1
−2
−1
0
1
2

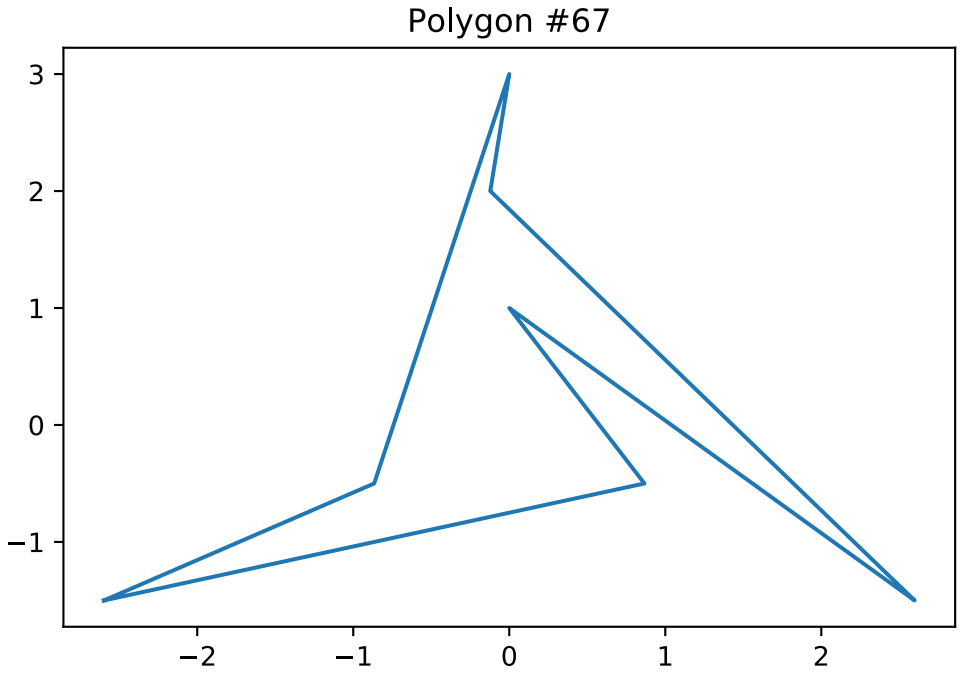

Polygon #62
3
2
1
0
−1
−2
−1
0
1
2

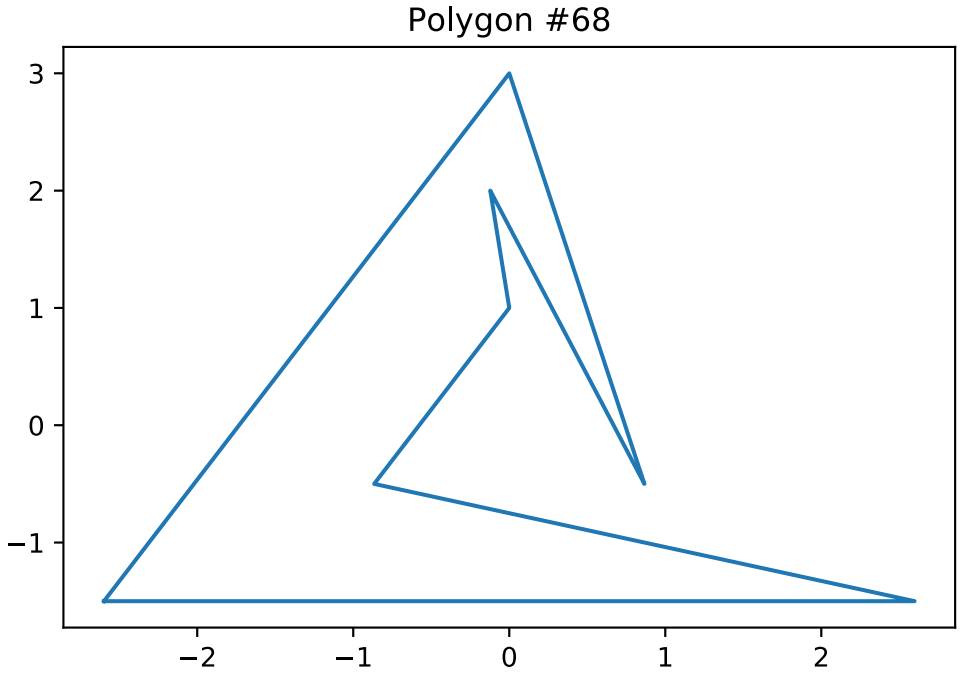

Polygon #63
3
2
1
0
−1
−2
−1
0
1
2

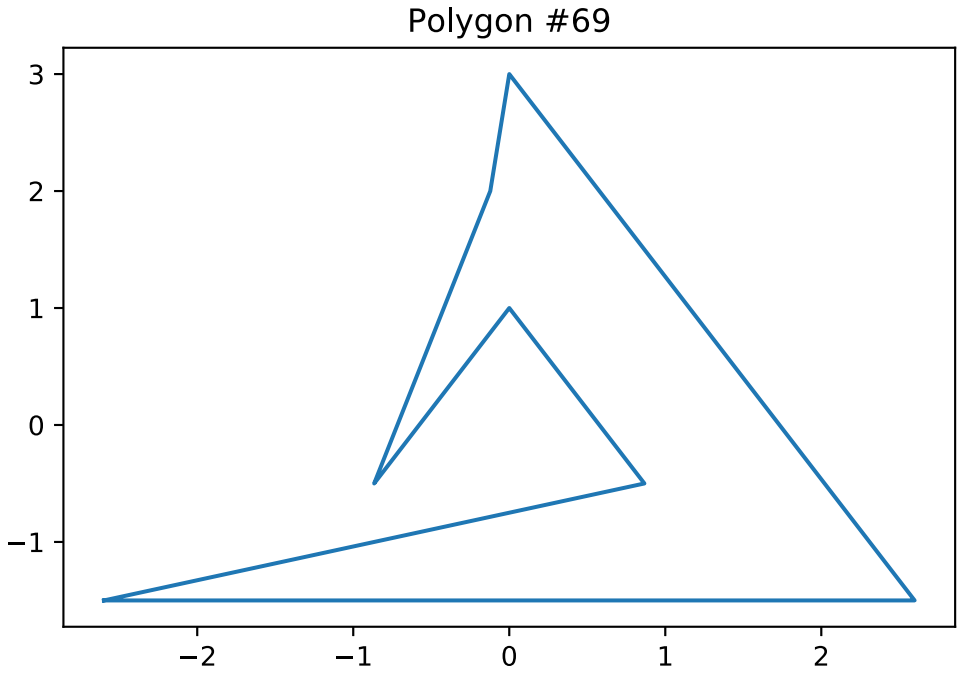

Polygon #64
3
2
1
0
−1
−2
−1
0
1
2

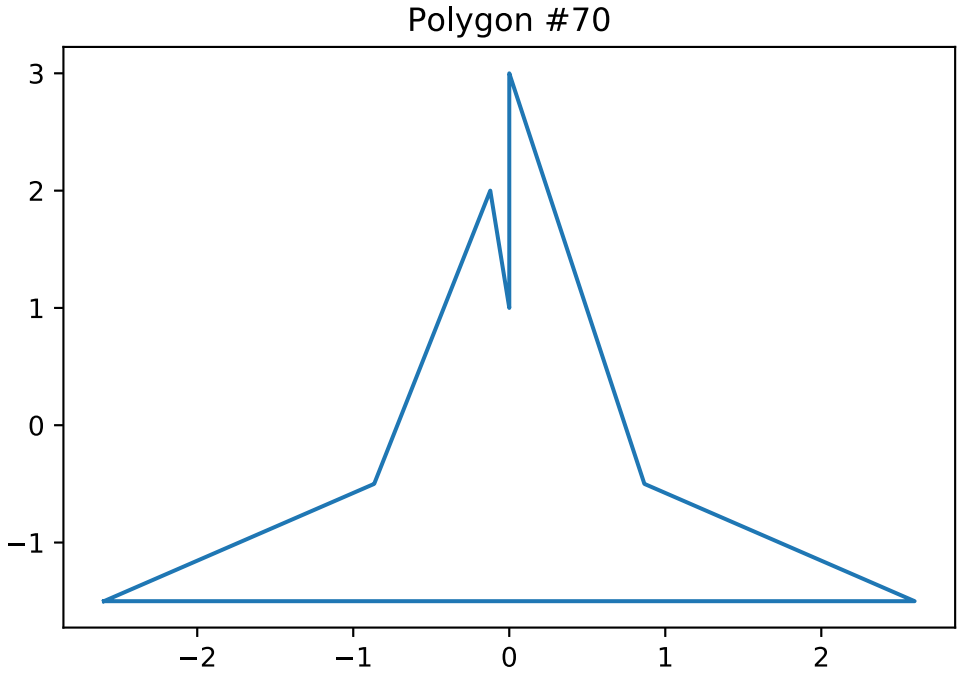

Polygon #65
3
2
1
0
−1
−2
−1
0
1
2

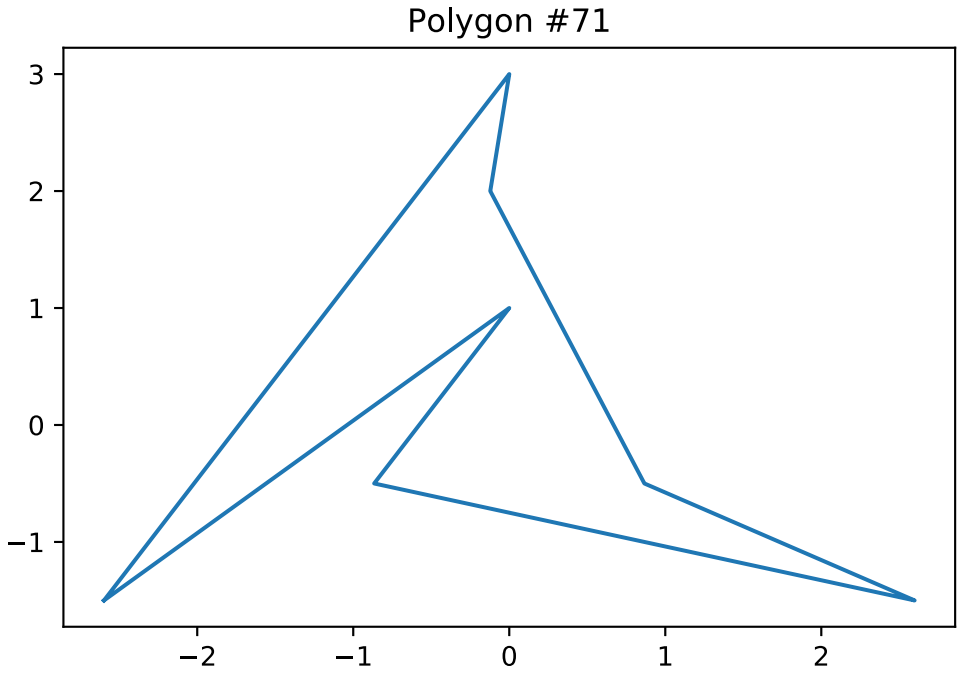

Polygon #66
3
2
1
0
−1
−2
−1
0
1
2

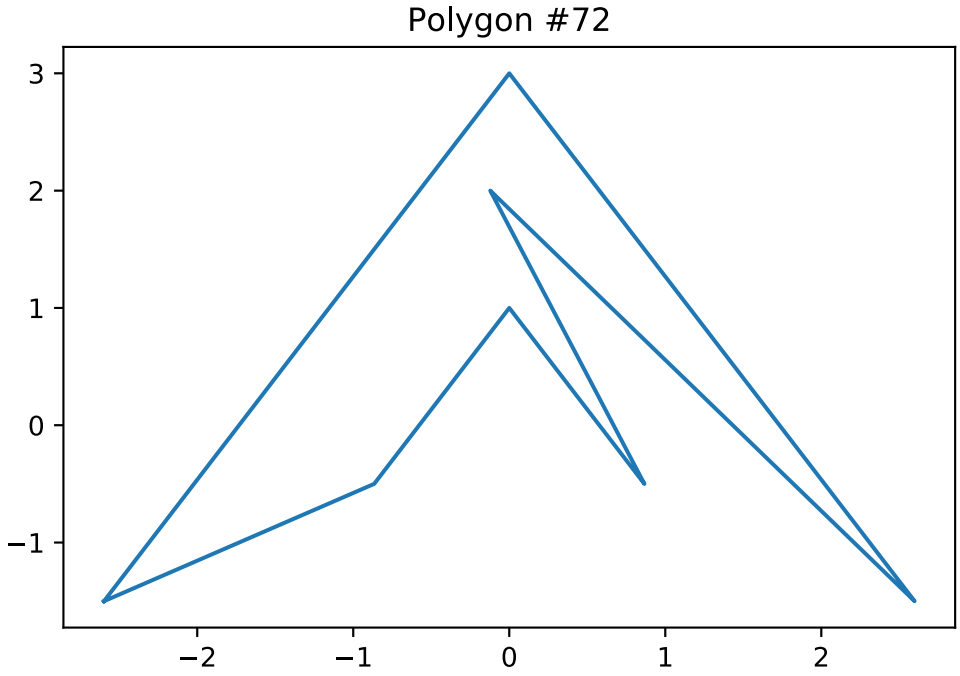

Polygon #67
3
2
1
0
−1
−2
−1
0
1
2

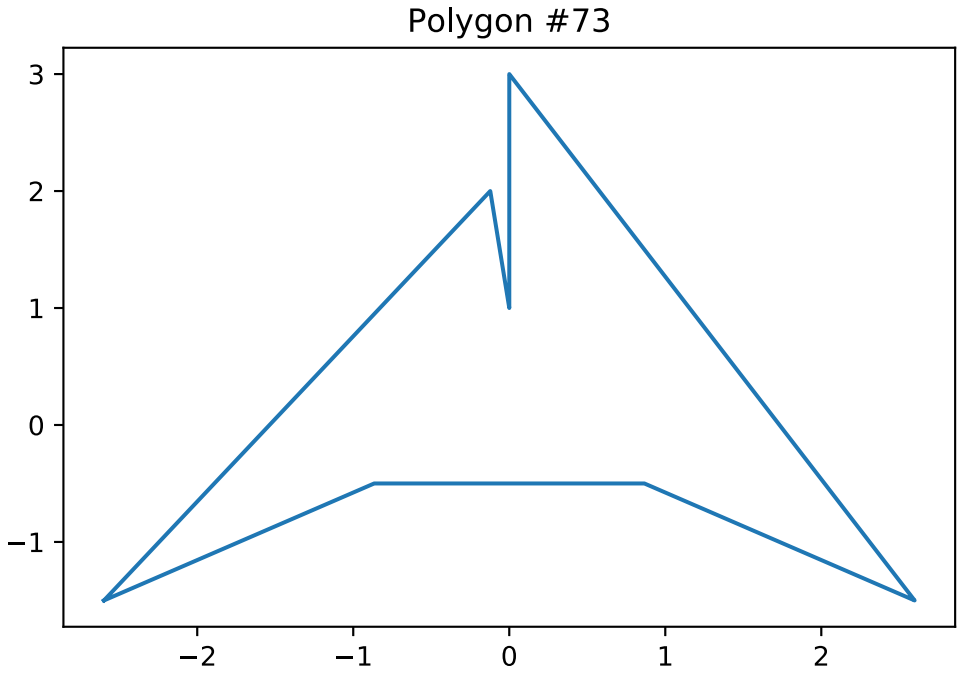

Polygon #68
3
2
1
0
−1
−2
−1
0
1
2

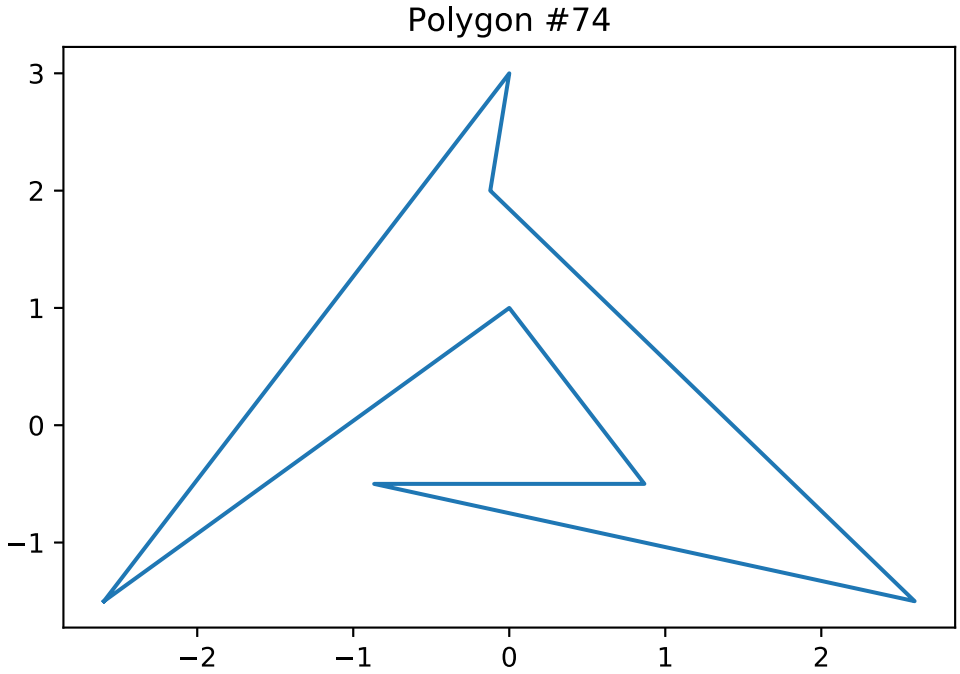

Polygon #69
3
2
1
0
−1
−2
−1
0
1
2

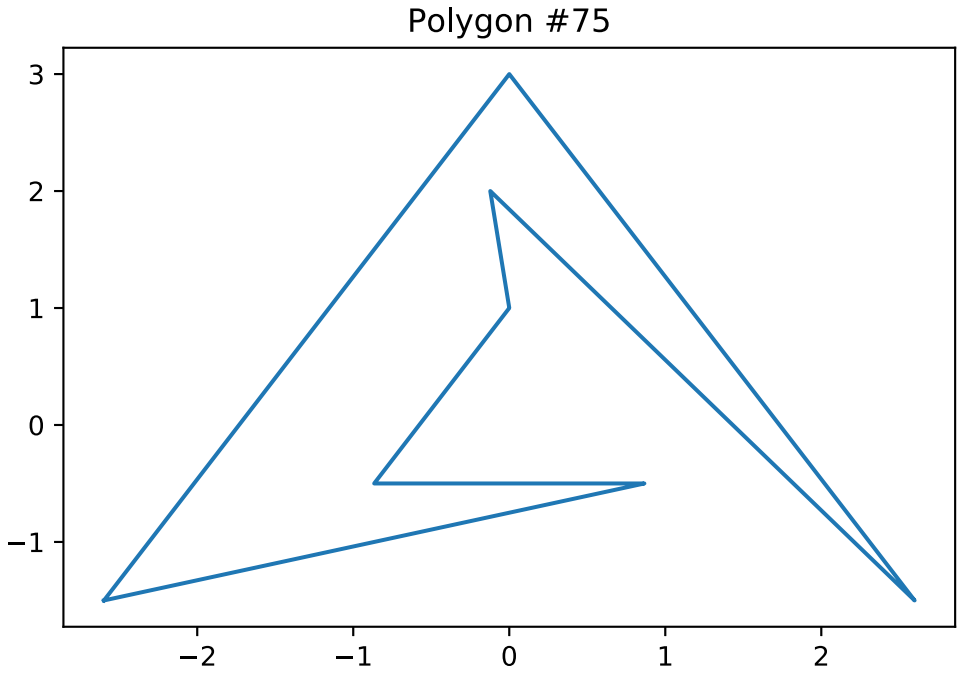

Polygon #70
3
2
1
0
−1
−2
−1
0
1
2

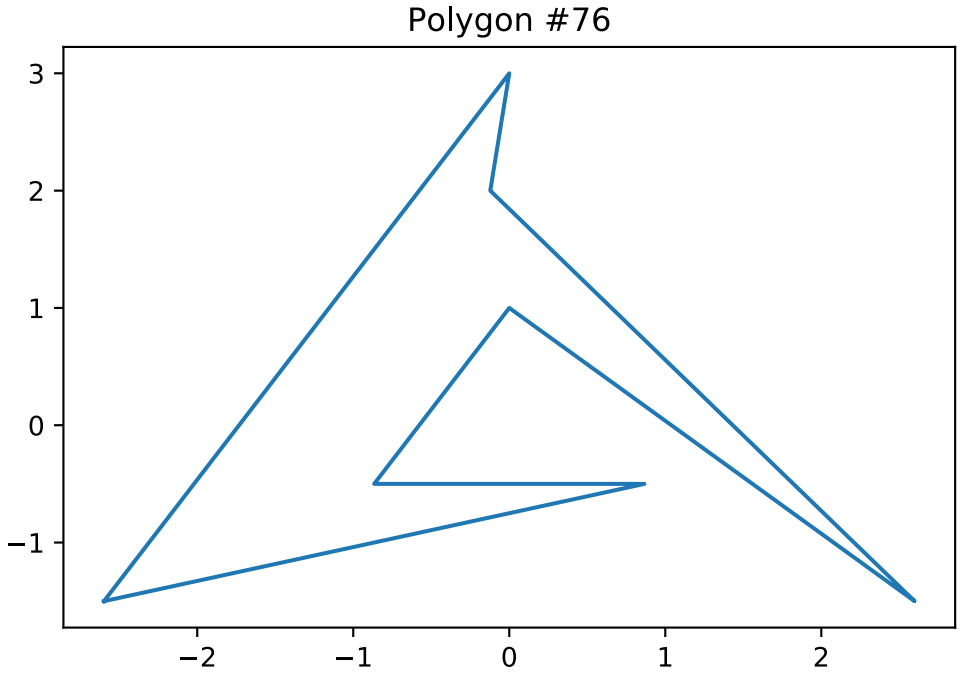

Polygon #71
3
2
1
0
−1
−2
−1
0
1
2

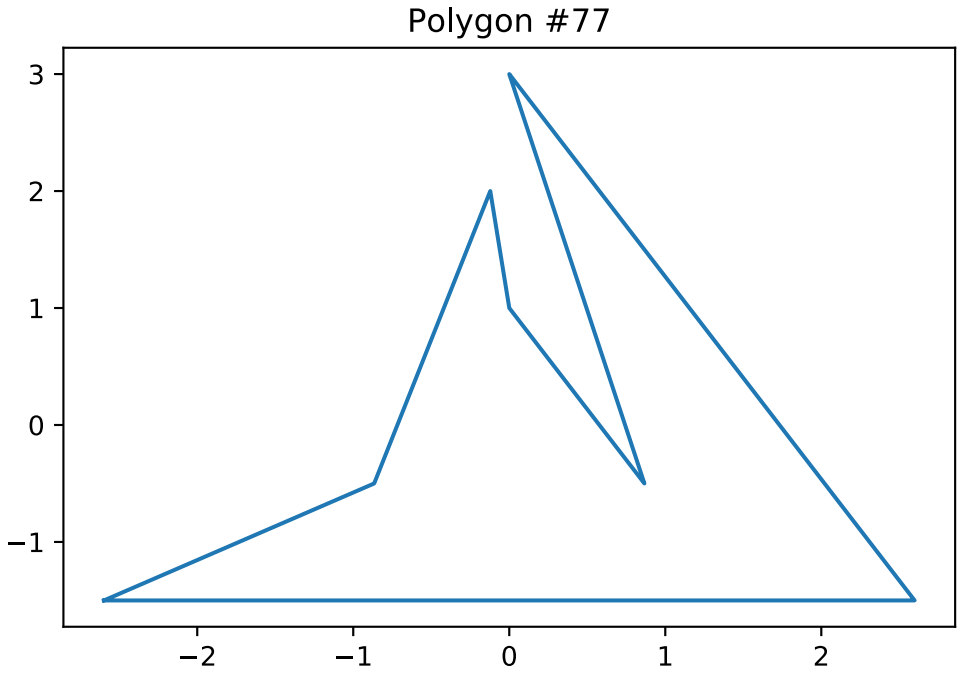

Polygon #72
3
2
1
0
−1
−2
−1
0
1
2

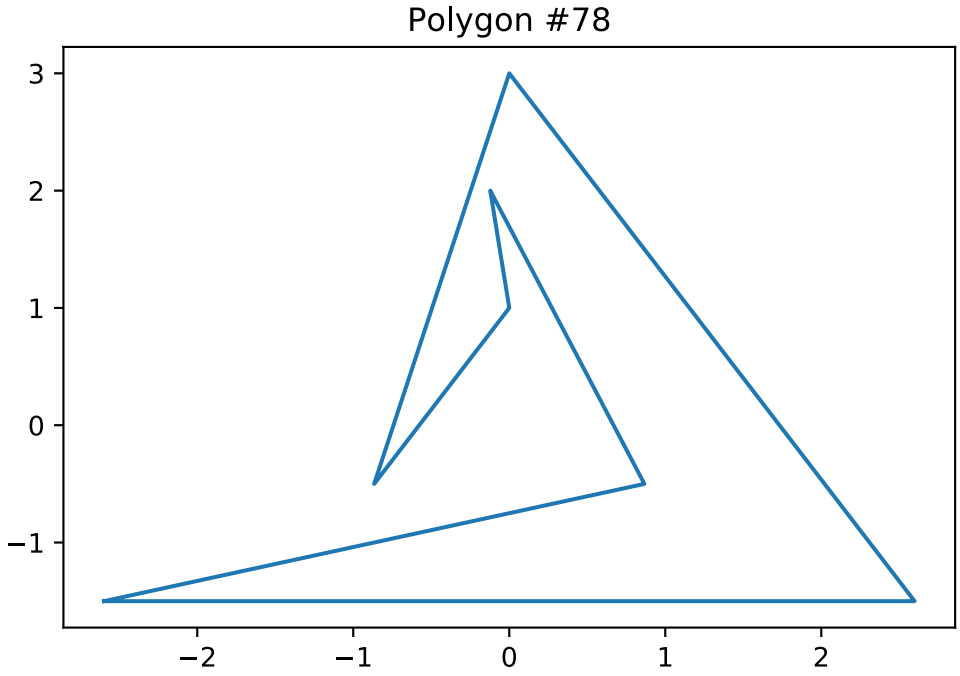

Polygon #73
3
2
1
0
−1
−2
−1
0
1
2

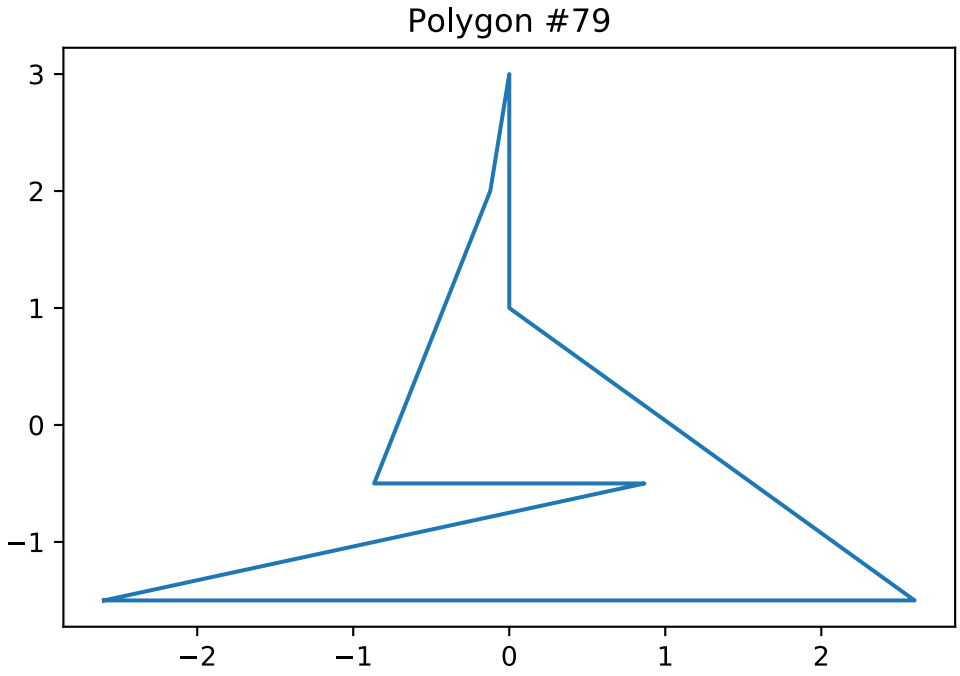

Polygon #74
3
2
1
0
−1
−2
−1
0
1
2

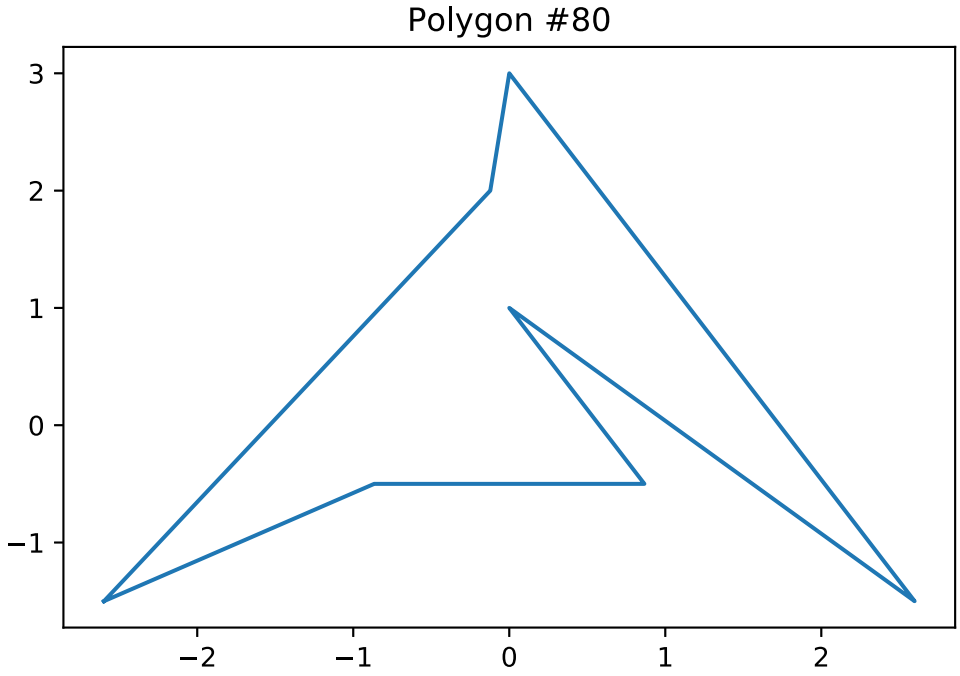

Polygon #75
3
2
1
0
−1
−2
−1
0
1
2

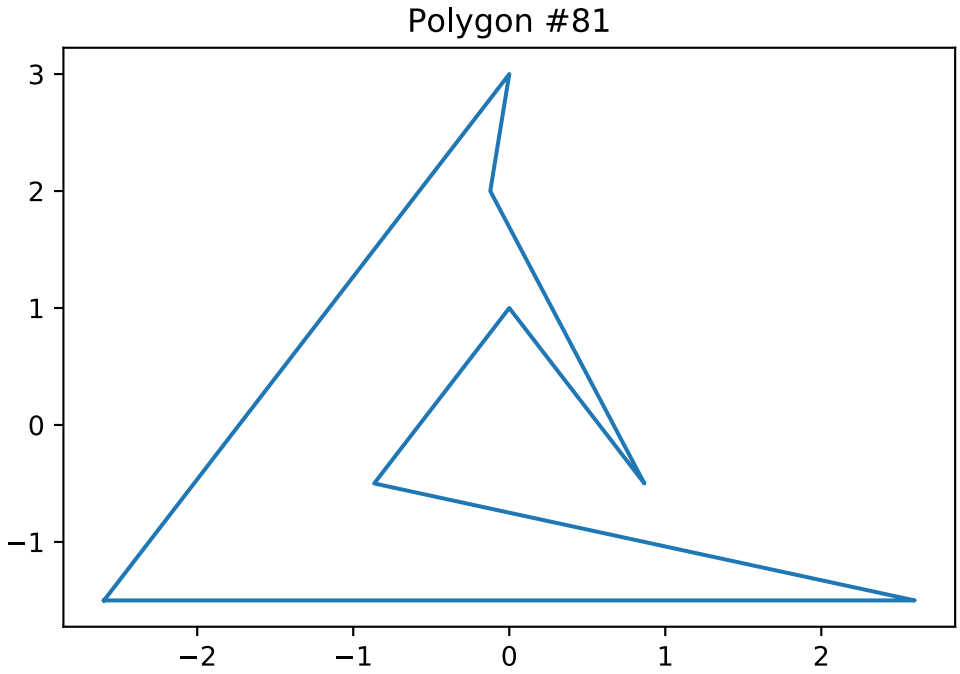

Polygon #76
3
2
1
0
−1
−2
−1
0
1
2

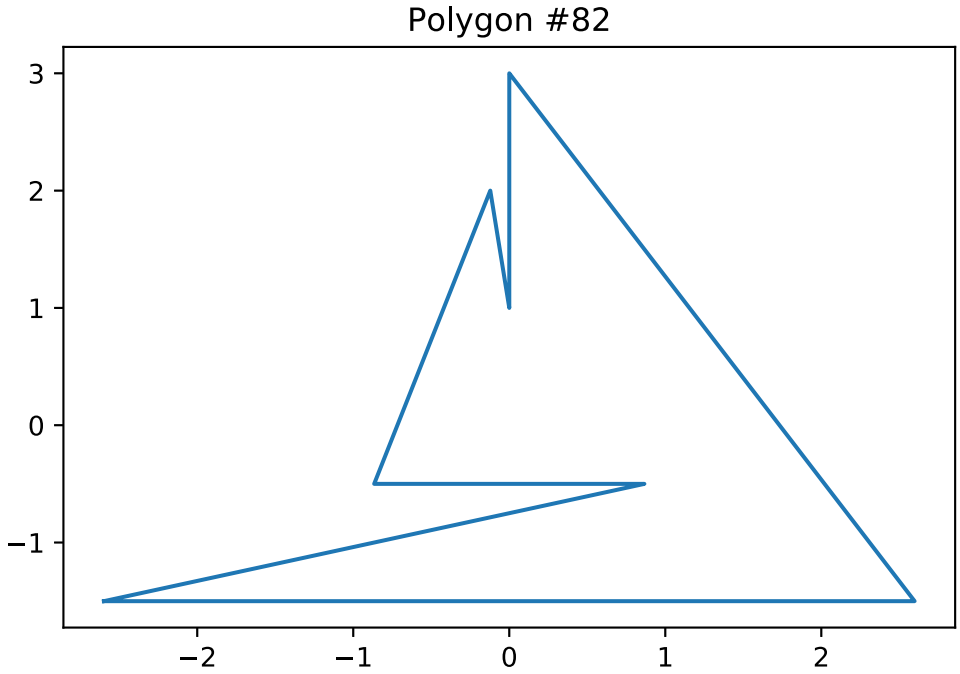

Polygon #77
3
2
1
0
−1
−2
−1
0
1
2

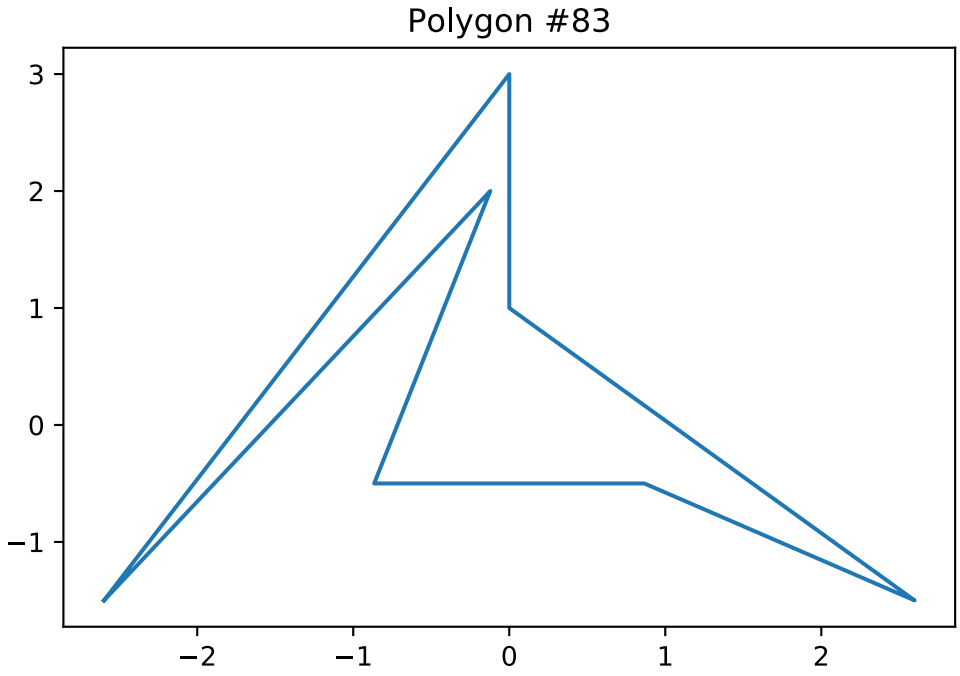

Polygon #78
3
2
1
0
−1
−2
−1
0
1
2

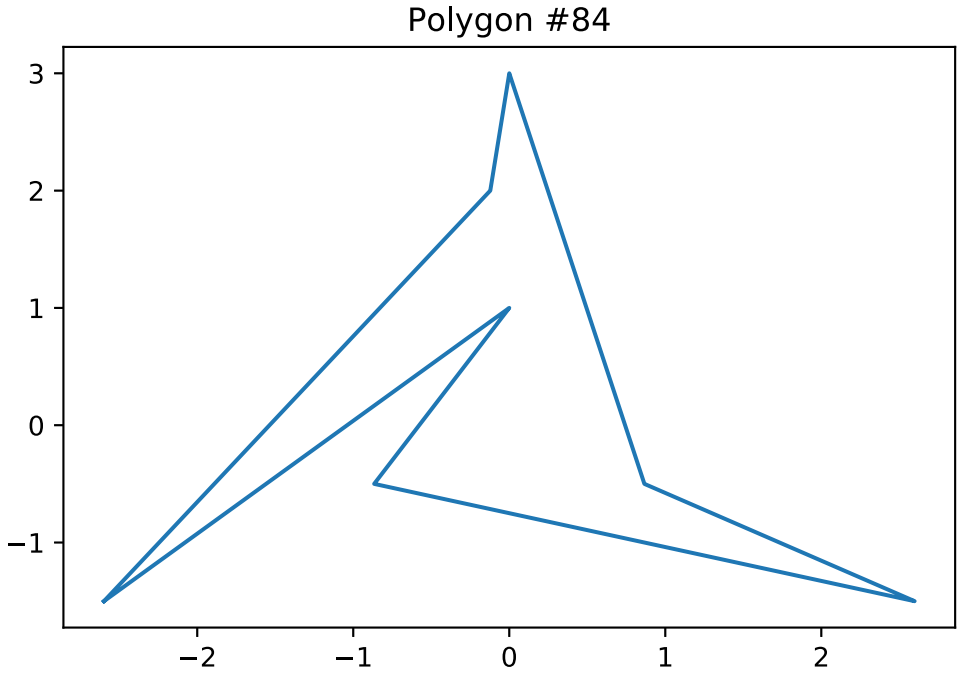

Polygon #79
3
2
1
0
−1
−2
−1
0
1
2

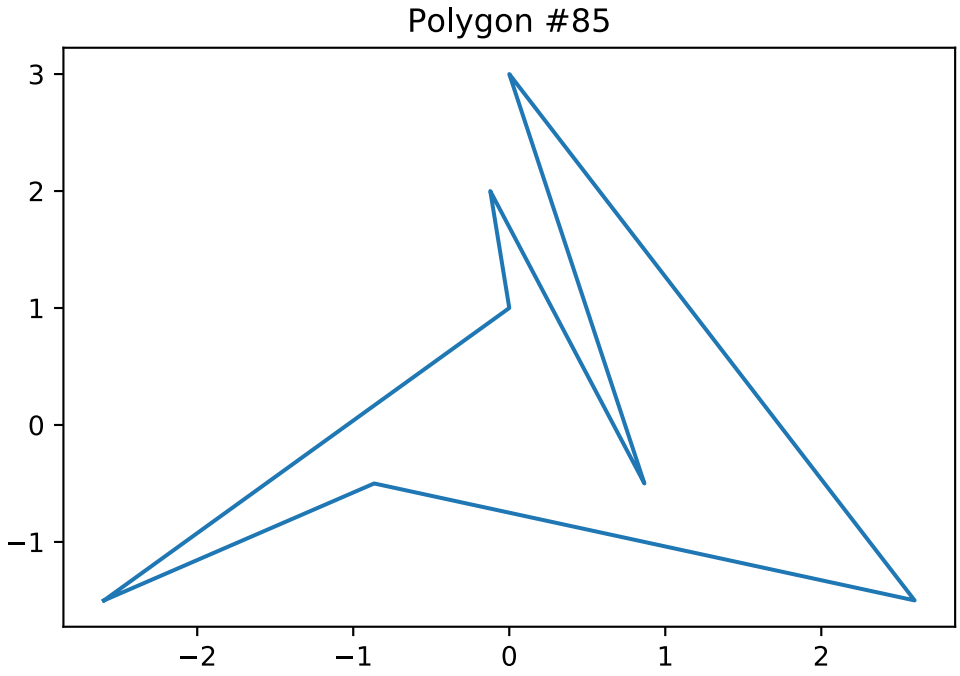

Polygon #80
3
2
1
0
−1
−2
−1
0
1
2

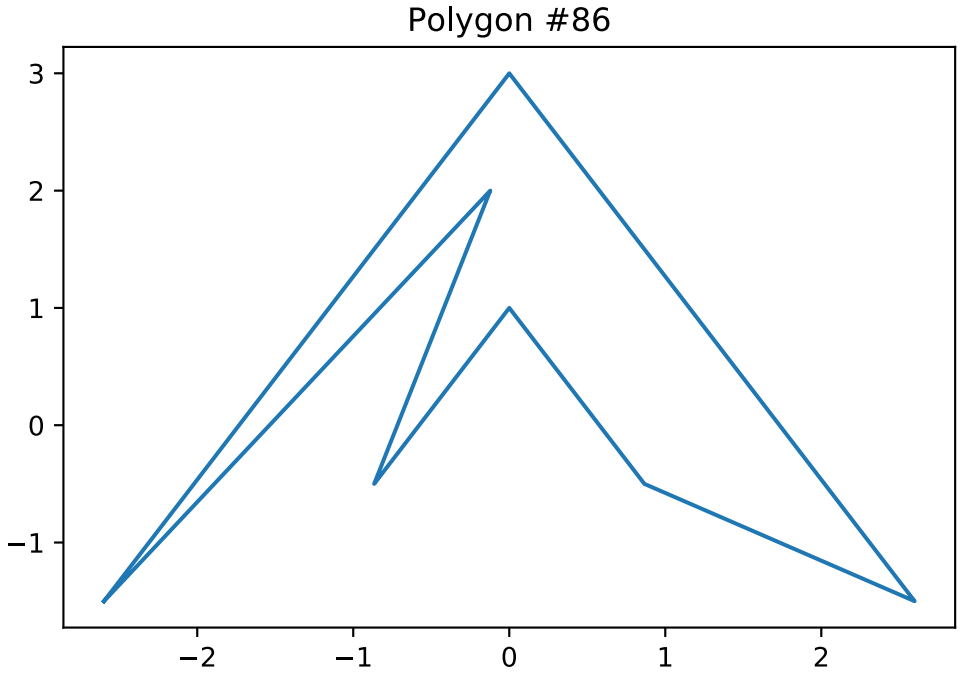

Polygon #81
3
2
1
0
−1
−2
−1
0
1
2

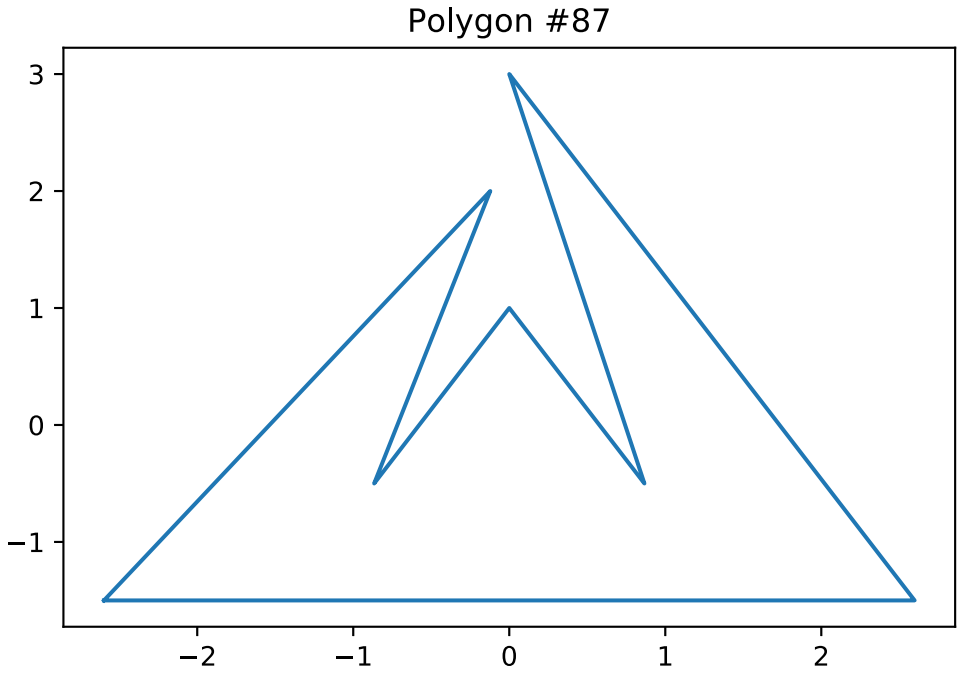

Polygon #82
3
2
1
0
−1
−2
−1
0
1
2

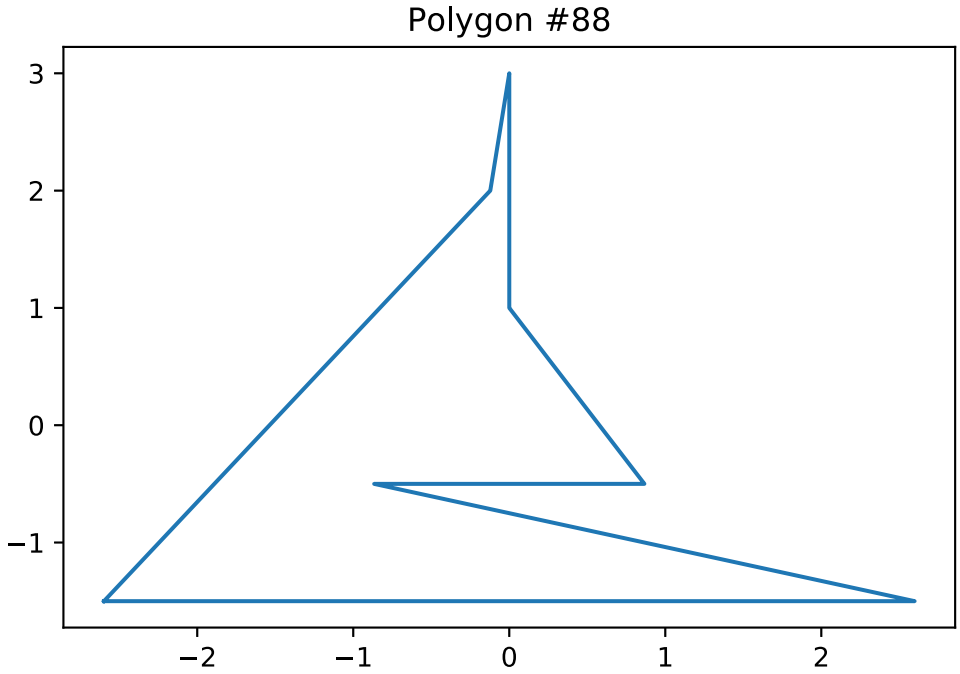

Polygon #83
3
2
1
0
−1
−2
−1
0
1
2

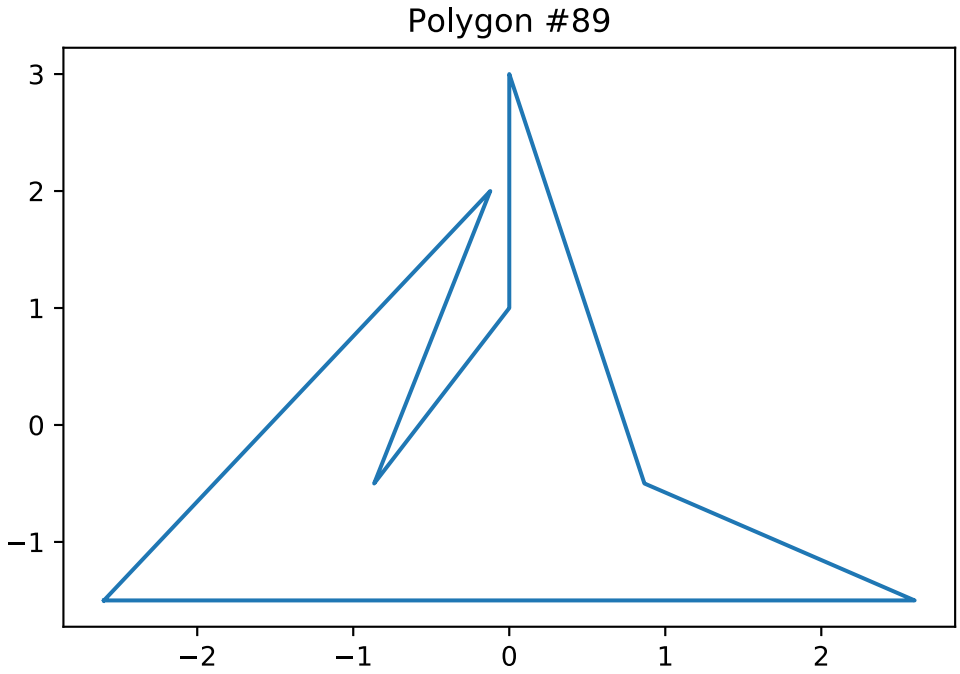

Polygon #84
3
2
1
0
−1
−2
−1
0
1
2

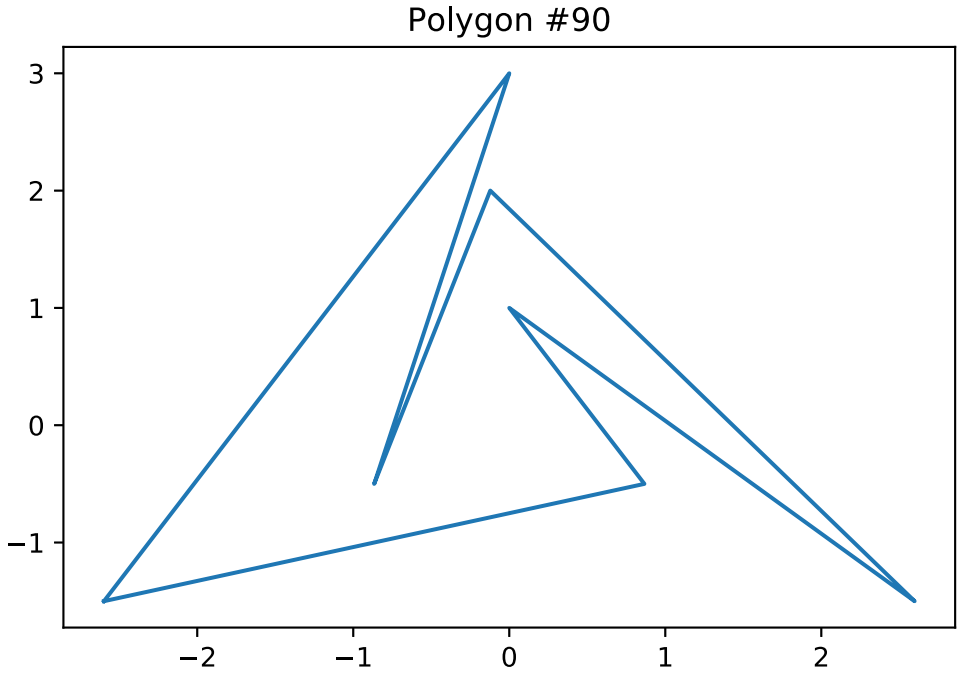

Polygon #85
3
2
1
0
−1
−2
−1
0
1
2

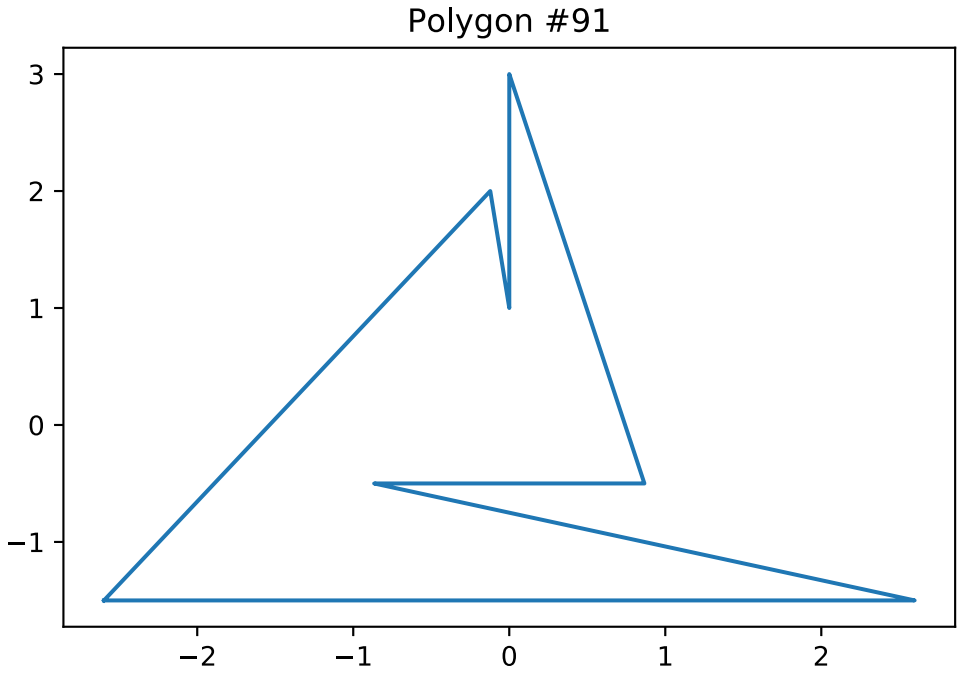

Polygon #86
3
2
1
0
−1
−2
−1
0
1
2


Polygon #87
3
2
1
0
−1
−2
−1
0
1
2


Polygon #88
3
2
1
0
−1
−2
−1
0
1
2


Polygon #89
3
2
1
0
−1
−2
−1
0
1
2


Polygon #90
3
2
1
0
−1
−2
−1
0
1
2



In [ ]: