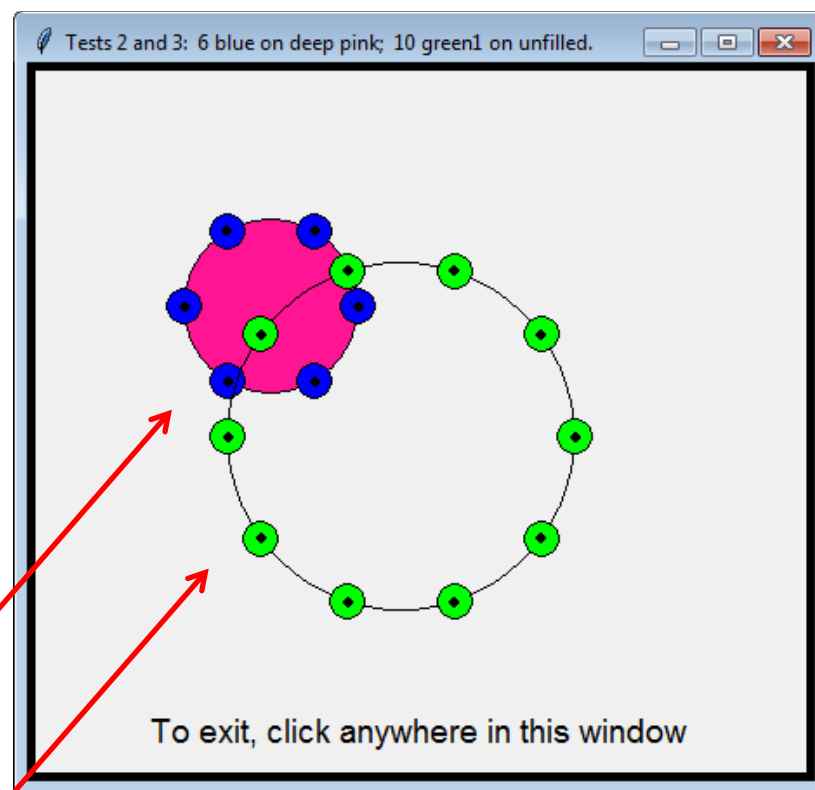
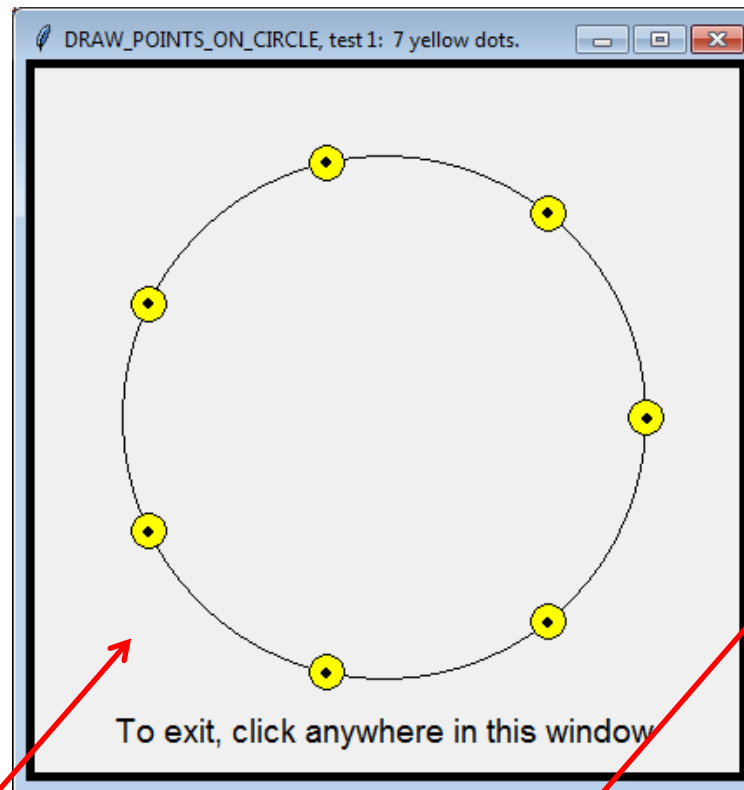


Your tests can draw whatever you decide are good tests. **These are just examples.**

Examples of the figures that your

draw_points_on_circle

function might draw:

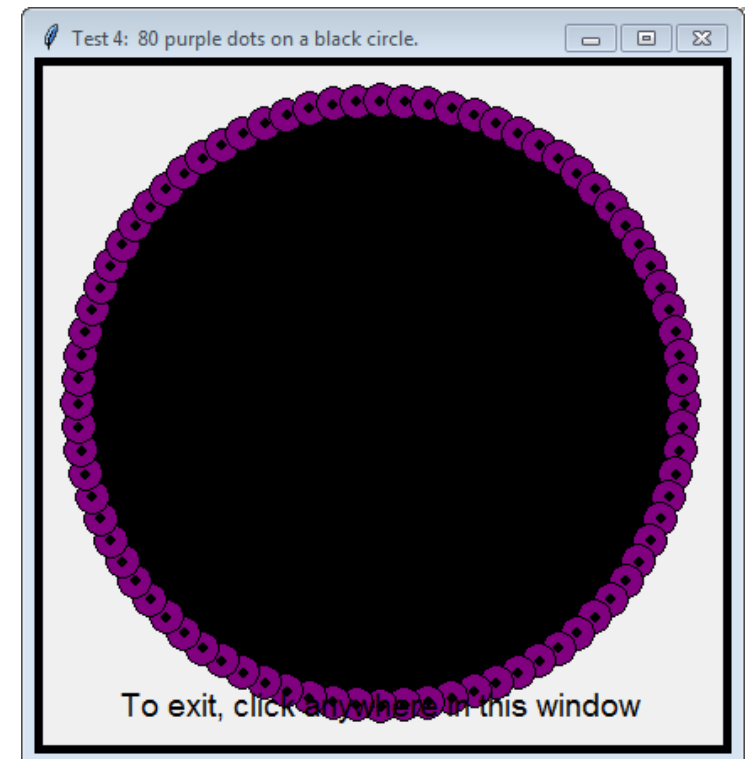


- On one window: **7 points (yellow dots)**
- Then on another window:
 - **6 points (blue dots) on a deep pink circle and**
 - **10 points (green1 dots)**

Note the tiny black dots in the center of the colored dots; those are the points on the circumference of the circle, generated by the ***generate_points_on_circle*** function that you are given and must **use (i.e., call)**.

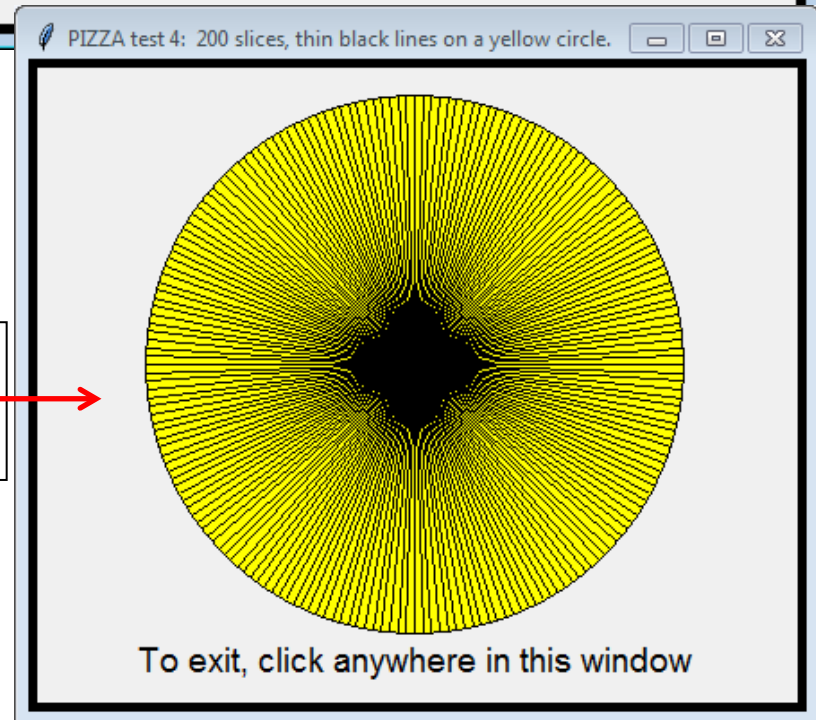
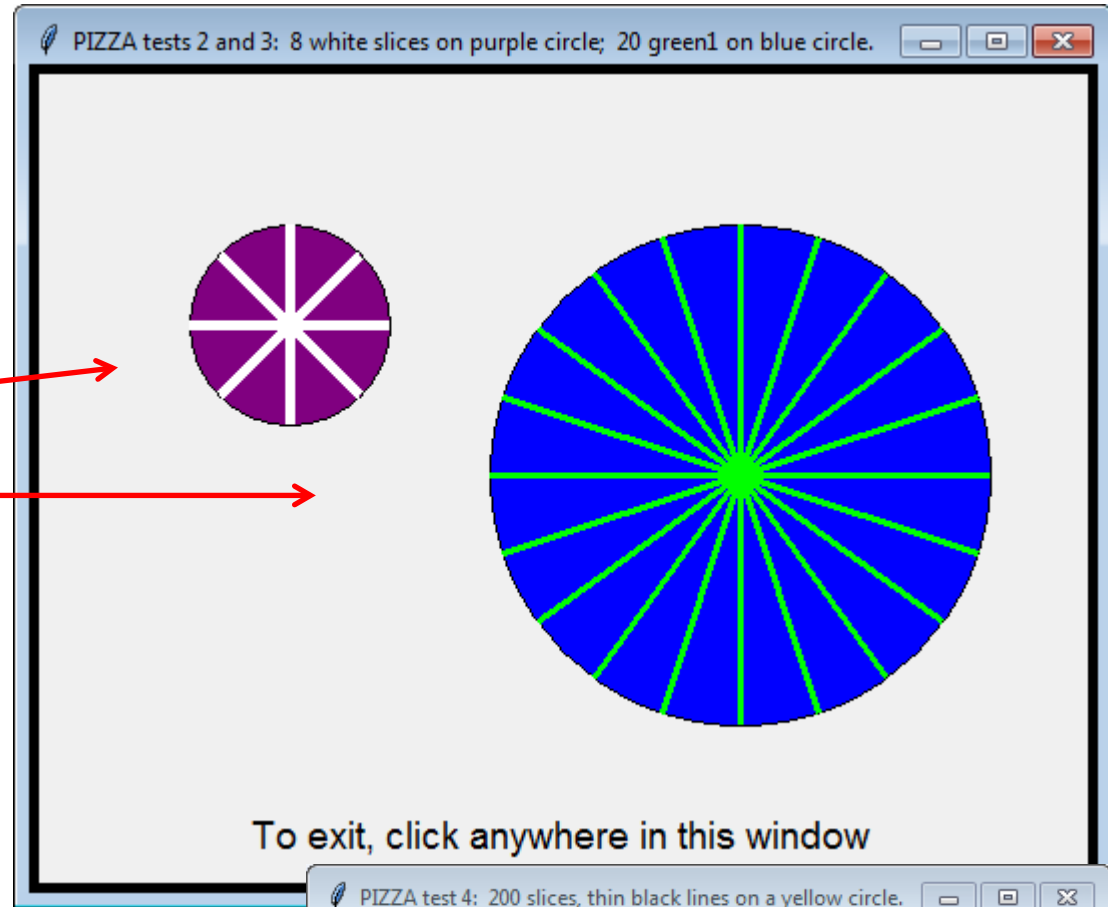
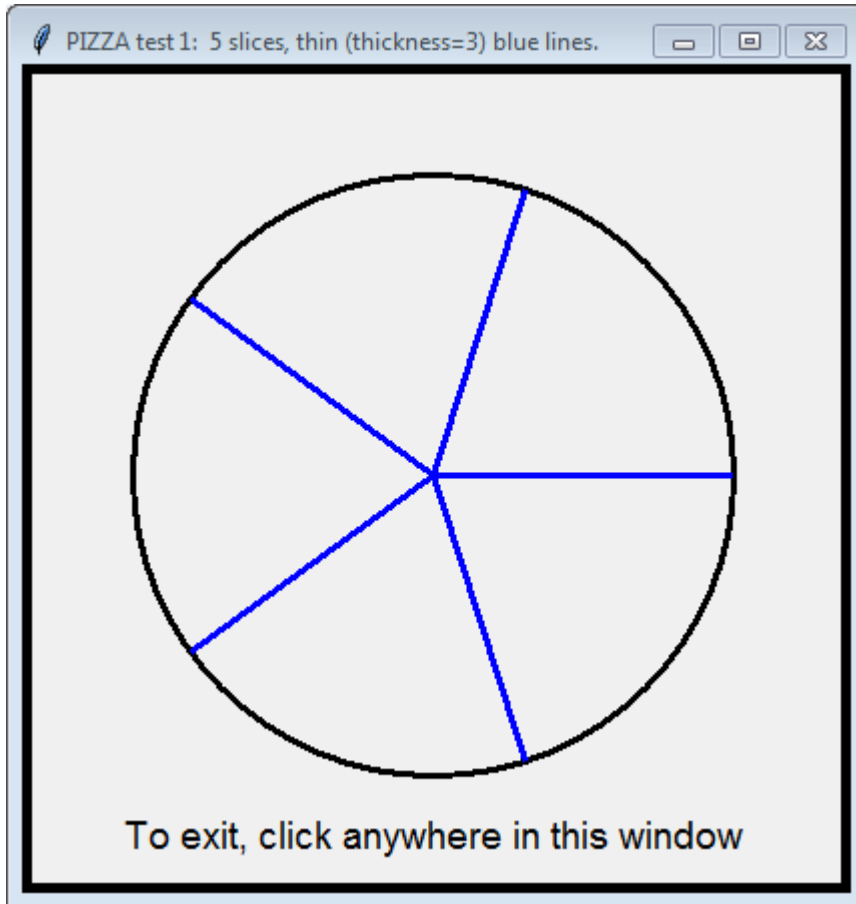
IMPORTANT: You must **CALL** function ***generate_points_on_circle***, but you must **NOT copy** any of its code. This is an exercise about **using** functions that already exist, usually in a library.

Challenge: Can you make a cool picture like this?



Examples of the figures that your *pizza* function might draw:

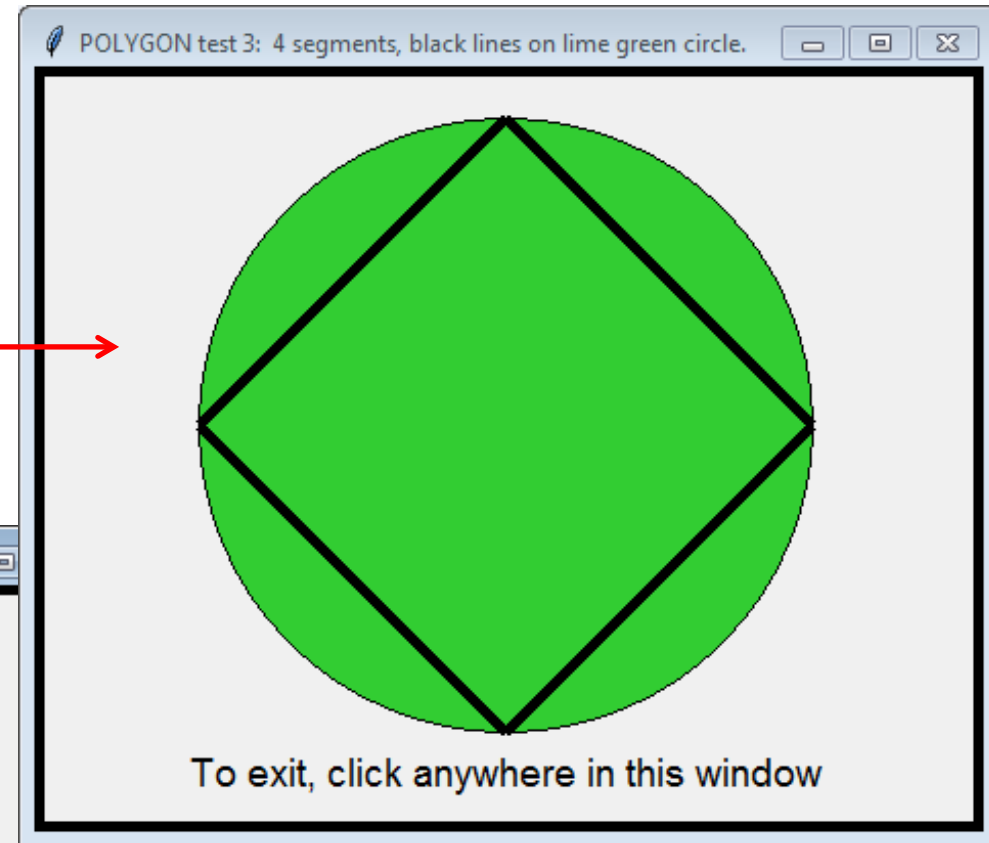
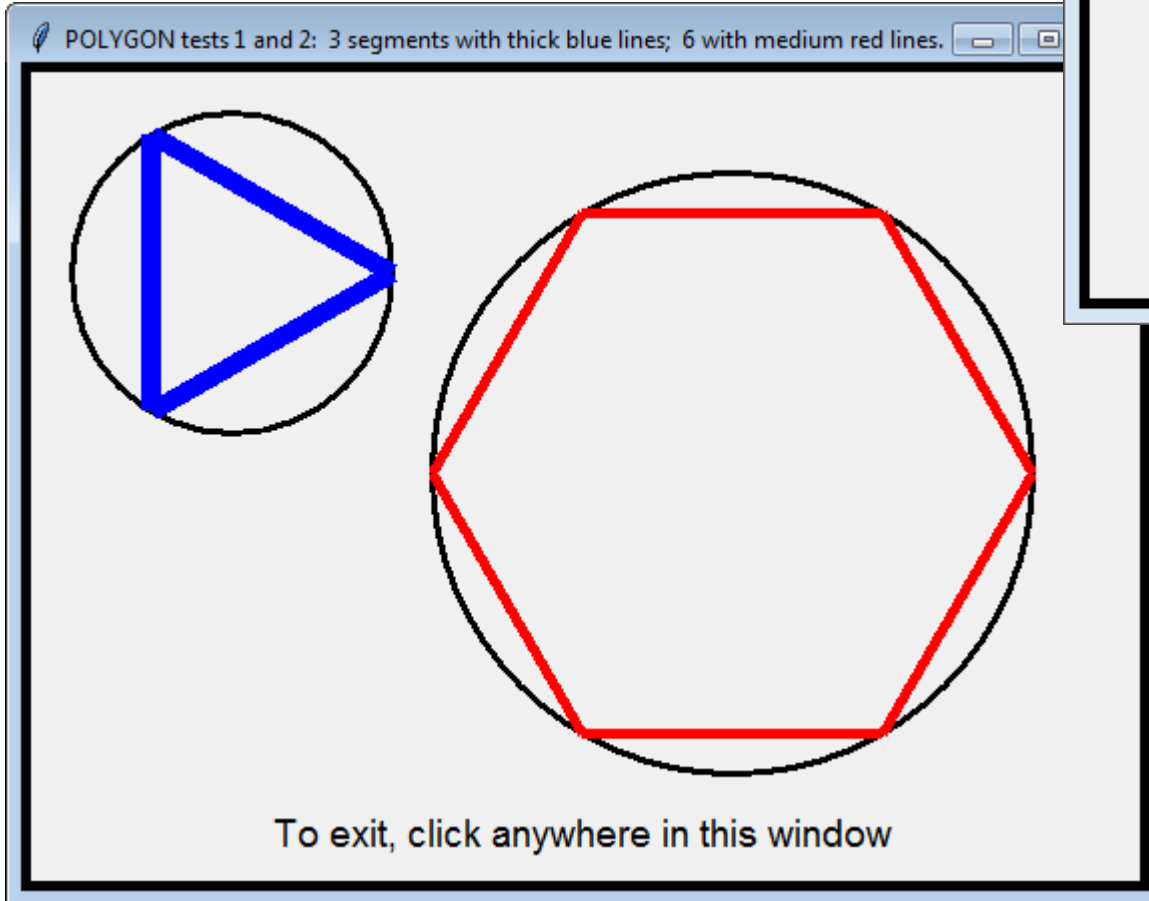
- On one window: 5 slices (**blue** lines, thickness=3)
- Then on another window:
 - 8 slices (**white** lines, thickness=5) on a **purple** circle and
 - 20 slices (**green1** lines, thickness=3) on a **blue** circle



Challenge: Can you make a cool picture like this?

Examples of the figures that your *polygon* function might draw:

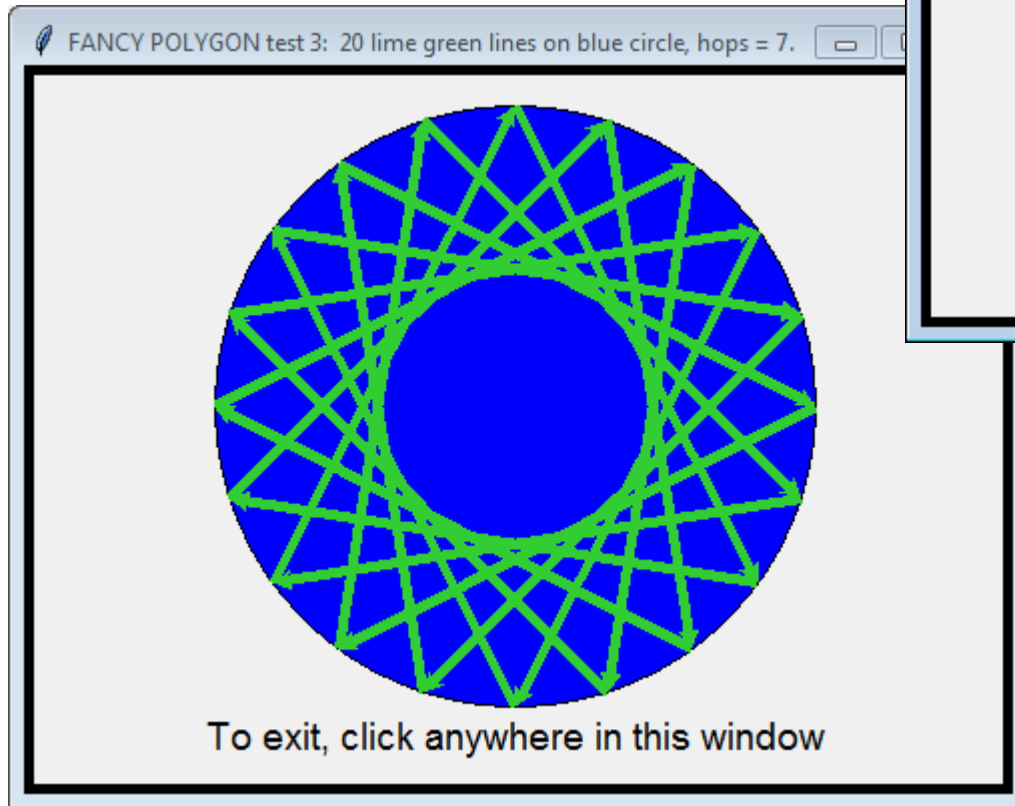
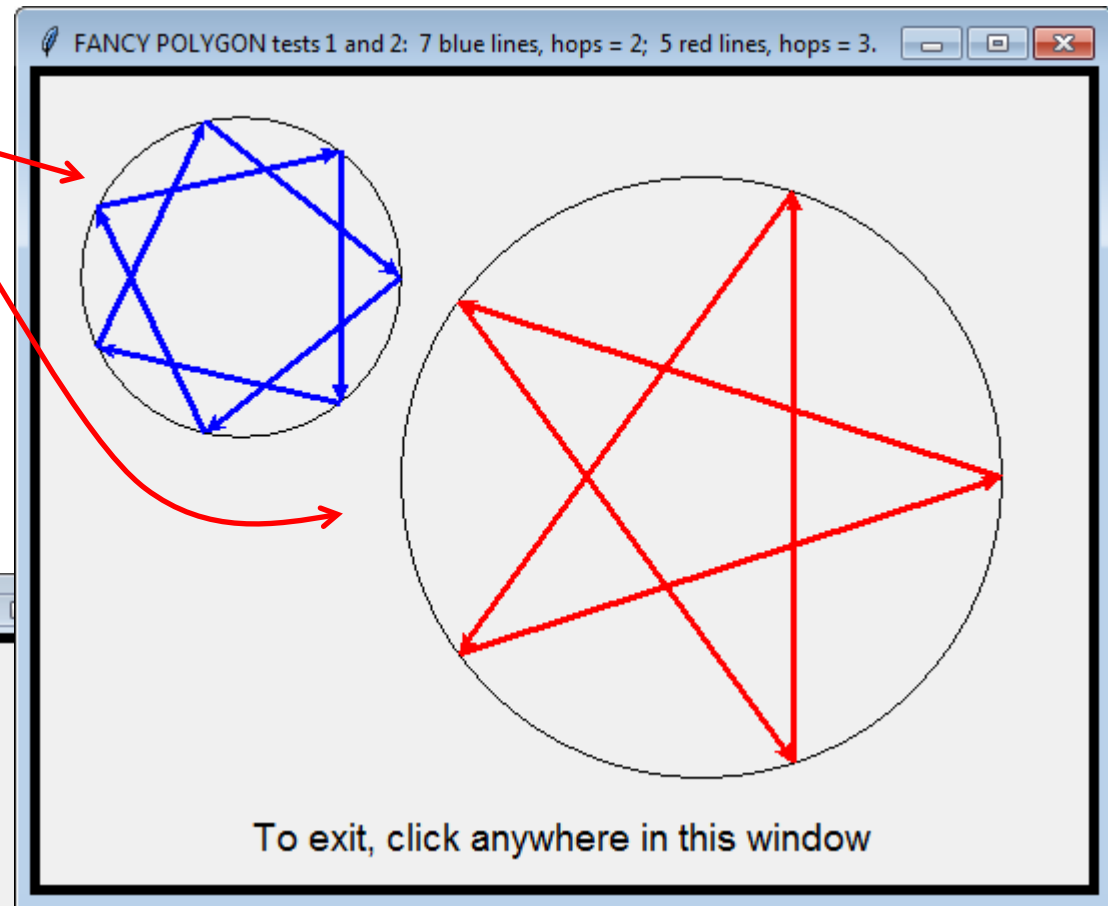
- On one window:
 - 3 segments (**blue** lines, thickness=10) and
 - 6 segments (**red** lines, thickness=5)
- Then on another window: 4 segments (**black** lines, thickness=5) on a **lime green** circle



Examples of the figures that your

fancy_polygon function might draw:

- 7 segments with `hops_to_next_point = 2` (blue lines, thickness=3)
- 5 segments with `hops_to_next_point = 3` (red lines, thickness=3)
- 20 segments with `hops_to_next_point = 7` (lime green lines, thickness=5, on a blue-filled circle)

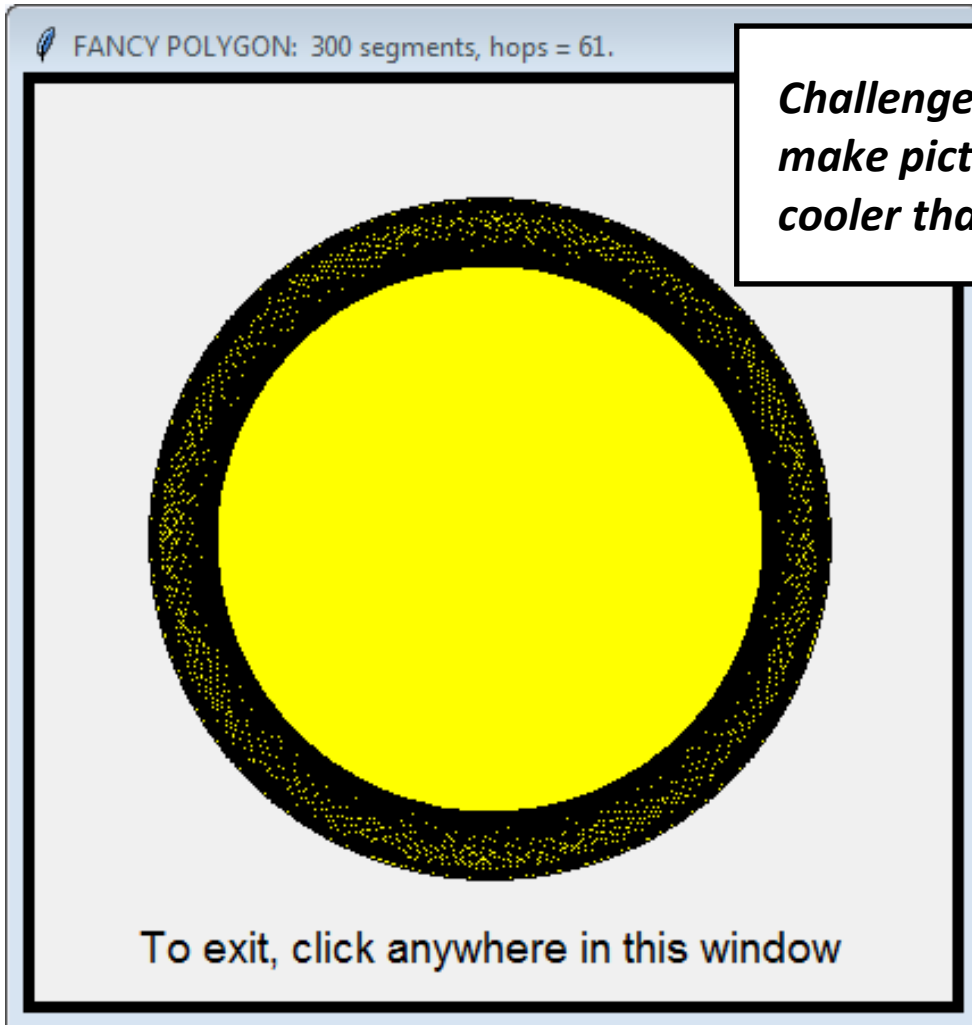
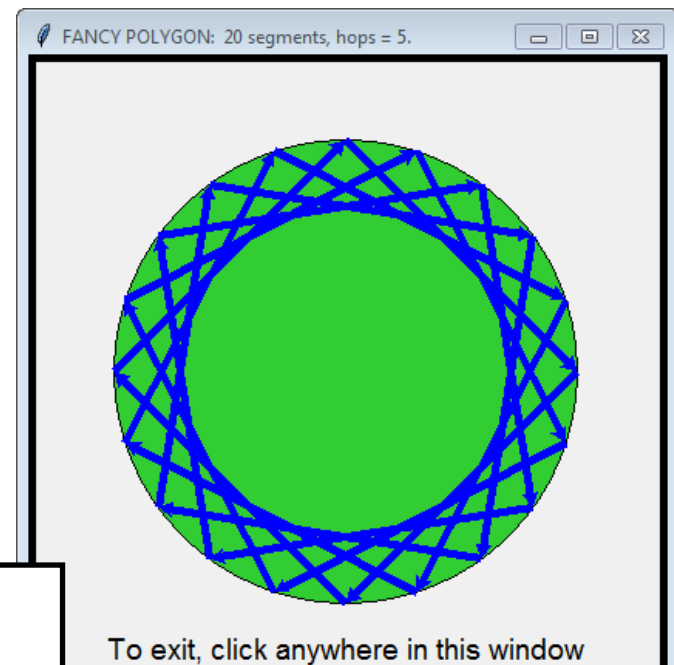


Challenge: What cool pictures can YOU make with *fancy_polygon*?

See the next page for ideas!

More examples of the figures that your *fancy_polygon* function might draw:

- **20** segments, **hops_to_next_point** = 5
(**blue** lines, thickness=5) on a **limegreen-filled** circle
- **51** segments with **hops_to_next_point** = 25
(**red** lines, thickness=1) on a **white-filled** circle
- **300** segments with **hops_to_next_point** = 61
(**black** lines, thickness=1) on a **yellow-filled** circle



Challenge: Can you make pictures even cooler than these?

