

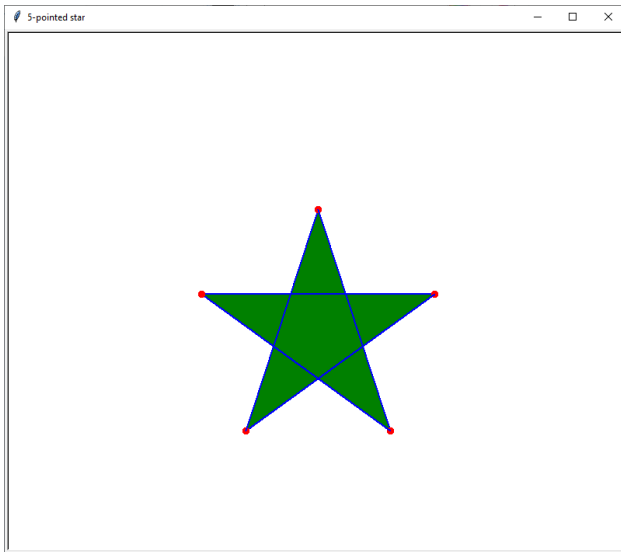
## CSE 20

### Beginning Programming in Python Programming Assignment 2

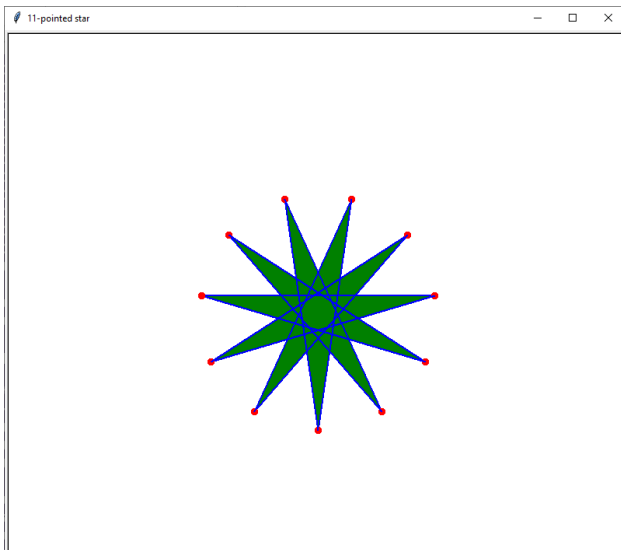
In this project you will write a Python program that uses the turtle module to draw an  $n$  pointed star, where  $n$  is any odd integer greater than or equal to 3, to be obtained from user input. The star will consist of  $n$  line segments of length 300, beginning with the line segment from  $(-150, 0)$  to  $(150, 0)$ . Each line segment will be 2 pixels wide and will be colored blue. Each of the  $n$  points of the star will be a red dot of diameter 10. The interior of the star will be filled green.

Save your program in a file called `Star.py`. Several sample runs of your program are given below with user input and turtle graphics window output.

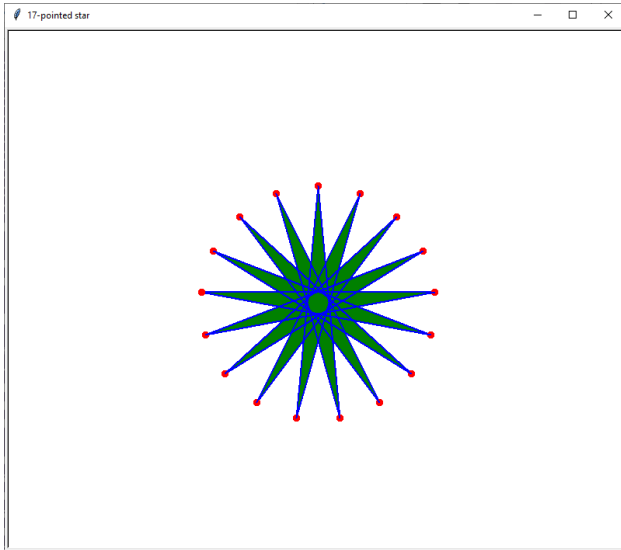
```
$ python Star.py
Enter an odd integer greater than or equal to 3: 5
```



```
$ python Star.py
Enter an odd integer greater than or equal to 3: 11
```



```
$ python Star.py  
Enter an odd integer greater than or equal to 3: 17
```



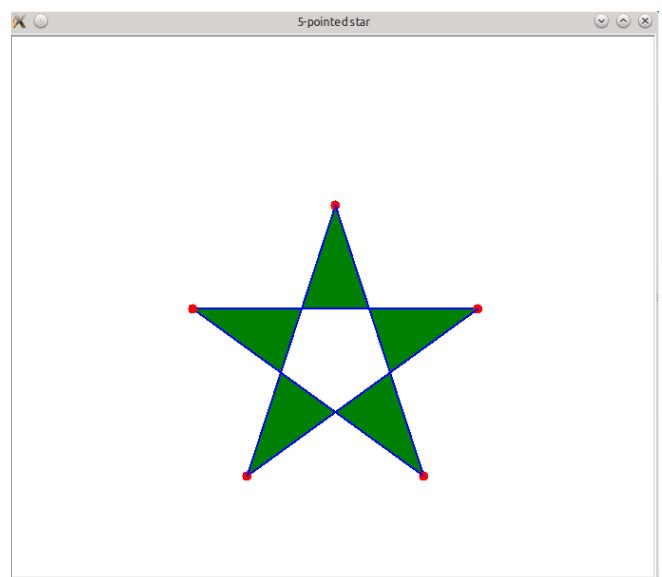
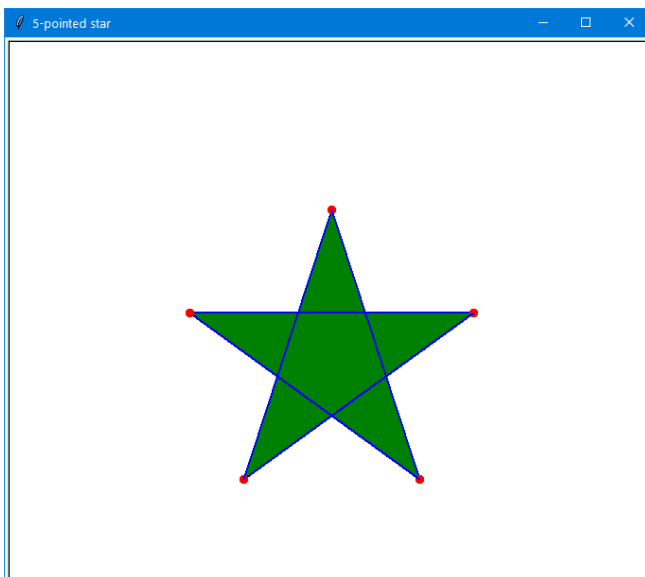
Notice that the graphics window displays the title " $n$ -pointed star" where  $n$  is the value entered by the user. Your program will use a for loop to draw the star, following the example `TurtleLoop3.py` posted on the webpage. The animation that draws the figure should be as fast as possible, and the final picture will have no turtle in it. All the turtle operations needed to satisfy these requirements can be found in the turtle module index at:

<https://docs.python.org/3/library/turtle.html?highlight=turtle#module-turtle>

Submit the file `Star.py` to Gradescope under the assignment name pa2.

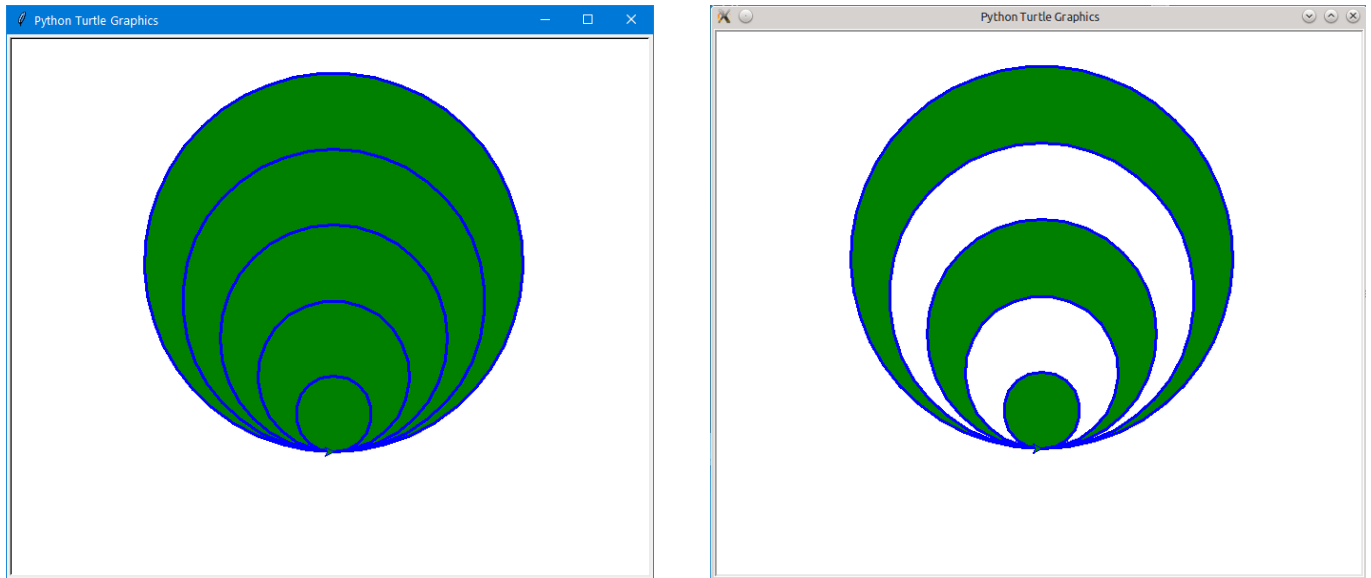
### Important Note

Be aware that the images produced by your program when run on Windows (pictured above) will look different when run on Macintosh (or Linux.) For instance, the 5-pointed star below left is from a Windows computer, while the one on the right will be produced by a Mac.



In other words, the same program, when run on two different platforms will produce two different pictures. The difference is due to alternate ways in which Windows and Mac render images. As long as your program is producing the right picture for your platform, your program is correct. Don't try to get your program to produce the picture for the other operating system. If you are on Windows, emulate the picture on the left, if on Mac, the right.

Another example of this phenomenon is illustrated by the program [Winding.py](#), posted in the examples section of the webpage. This one program produces the following two pictures on the two platforms, again Windows on the left, Mac on the right.



As a (no credit) exercise, look at the code for `Winding.py` and try to see what distinguishes the white pixels from the green pixels in the Mac picture. Hint: count the number of times a pixel has been encircled by the curve before it was filled. What do all the white pixels have in common, and what do all the green pixels have in common?