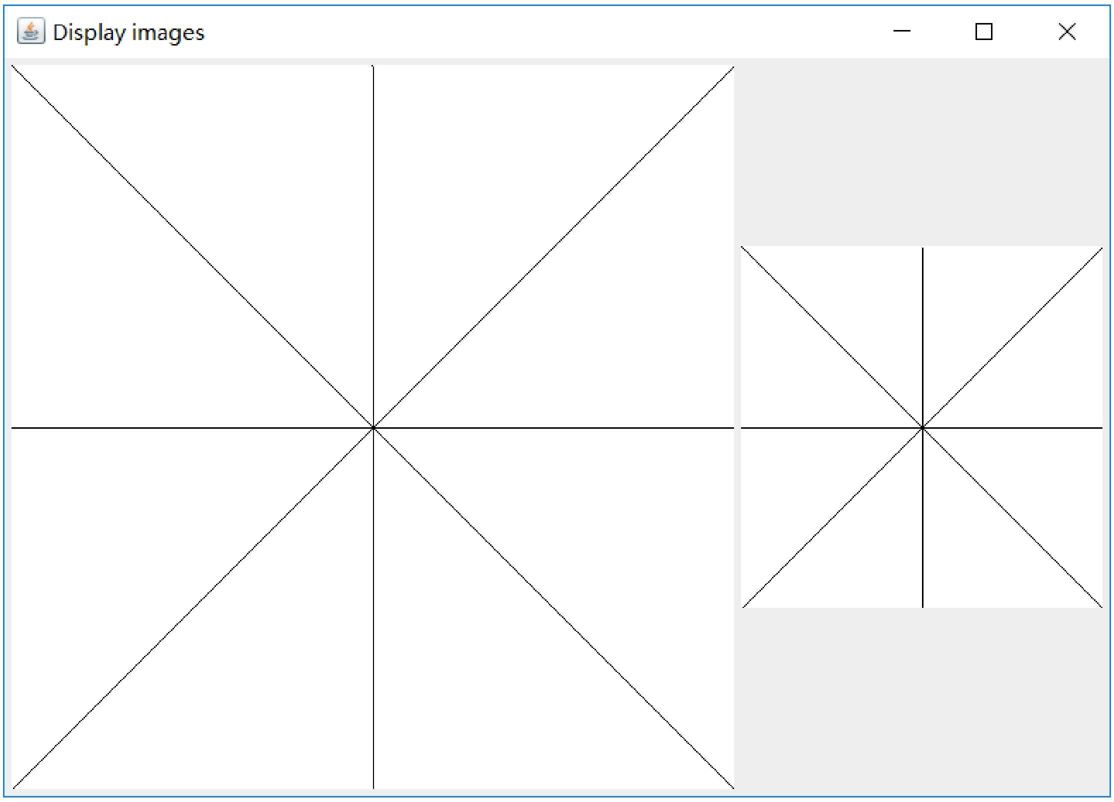
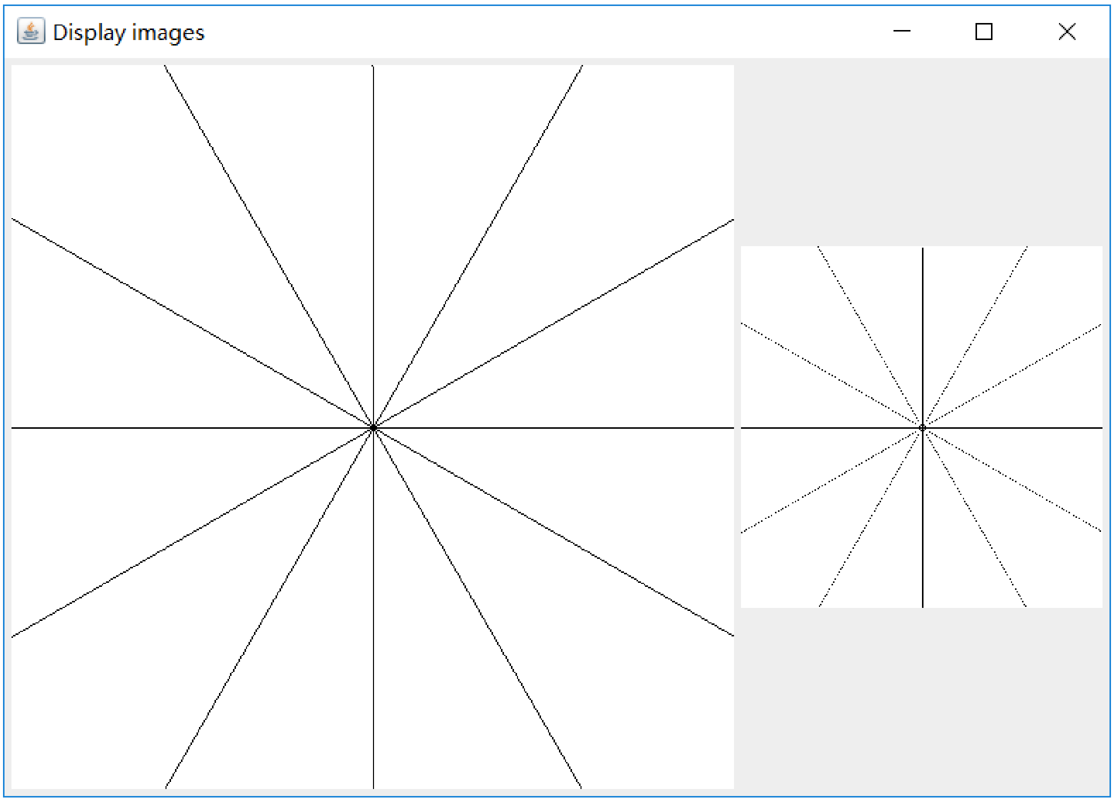
Part 1

1.Let's keep s=2.0.

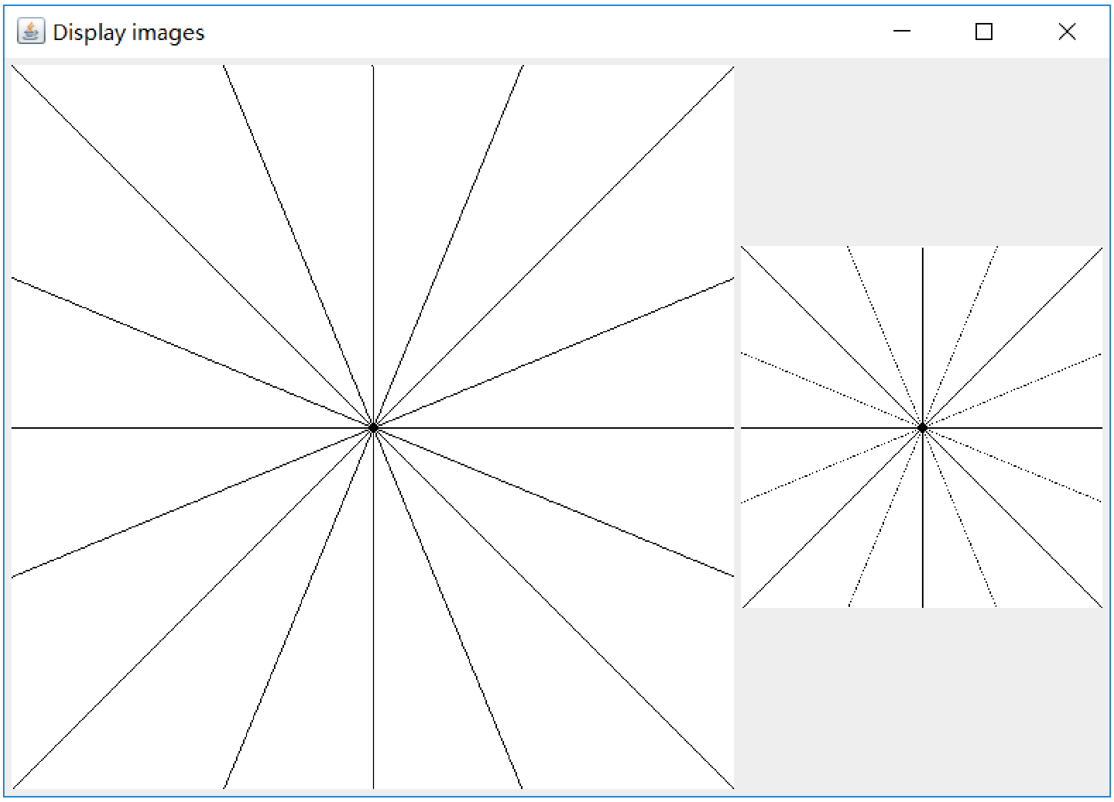
n=8:



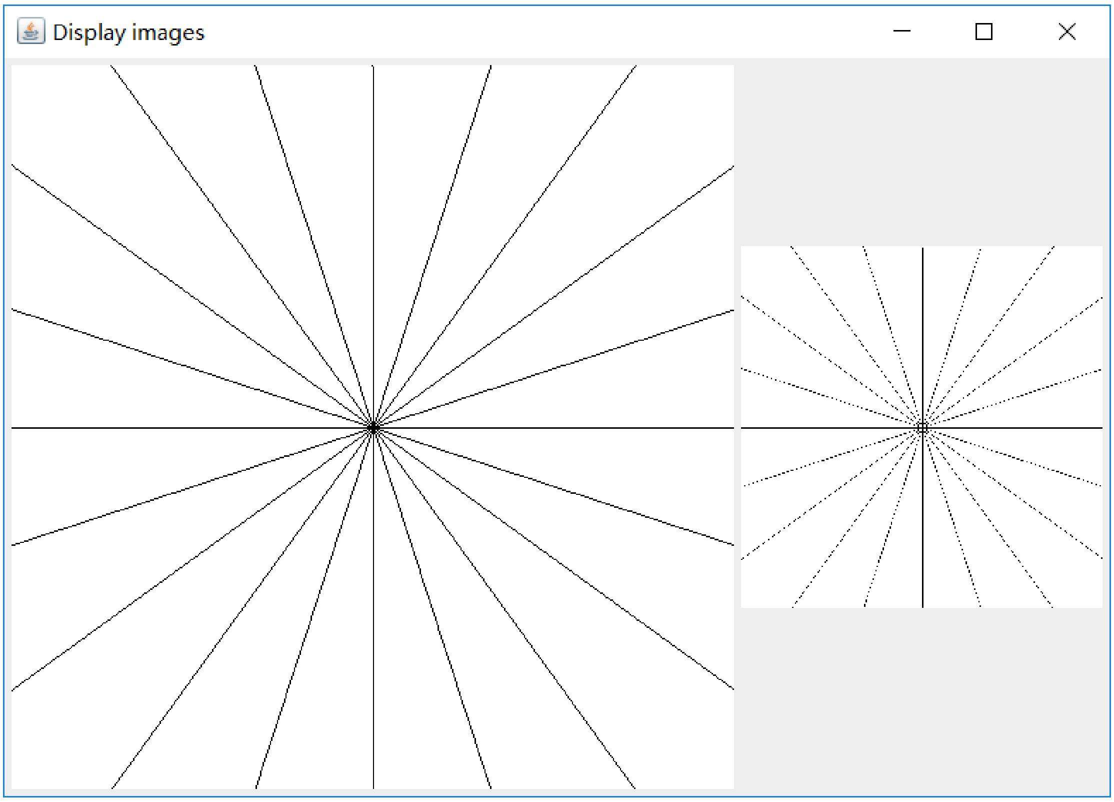
n=12:



n=16:



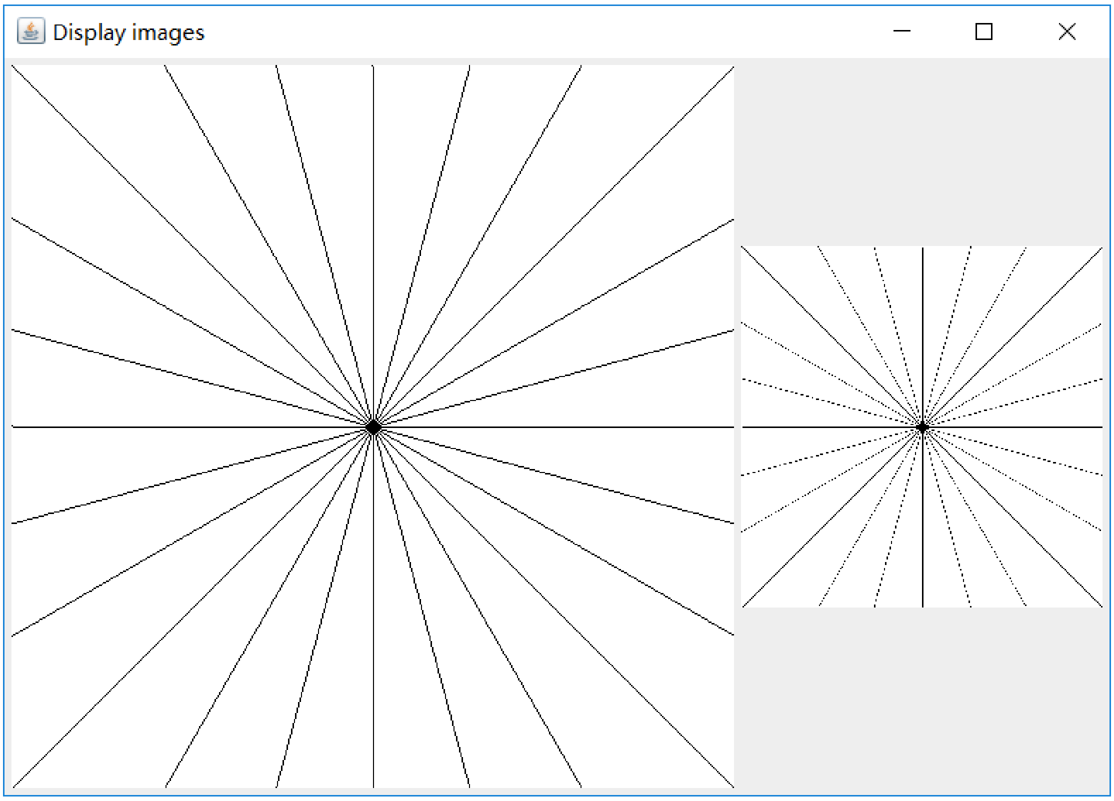
n=20:



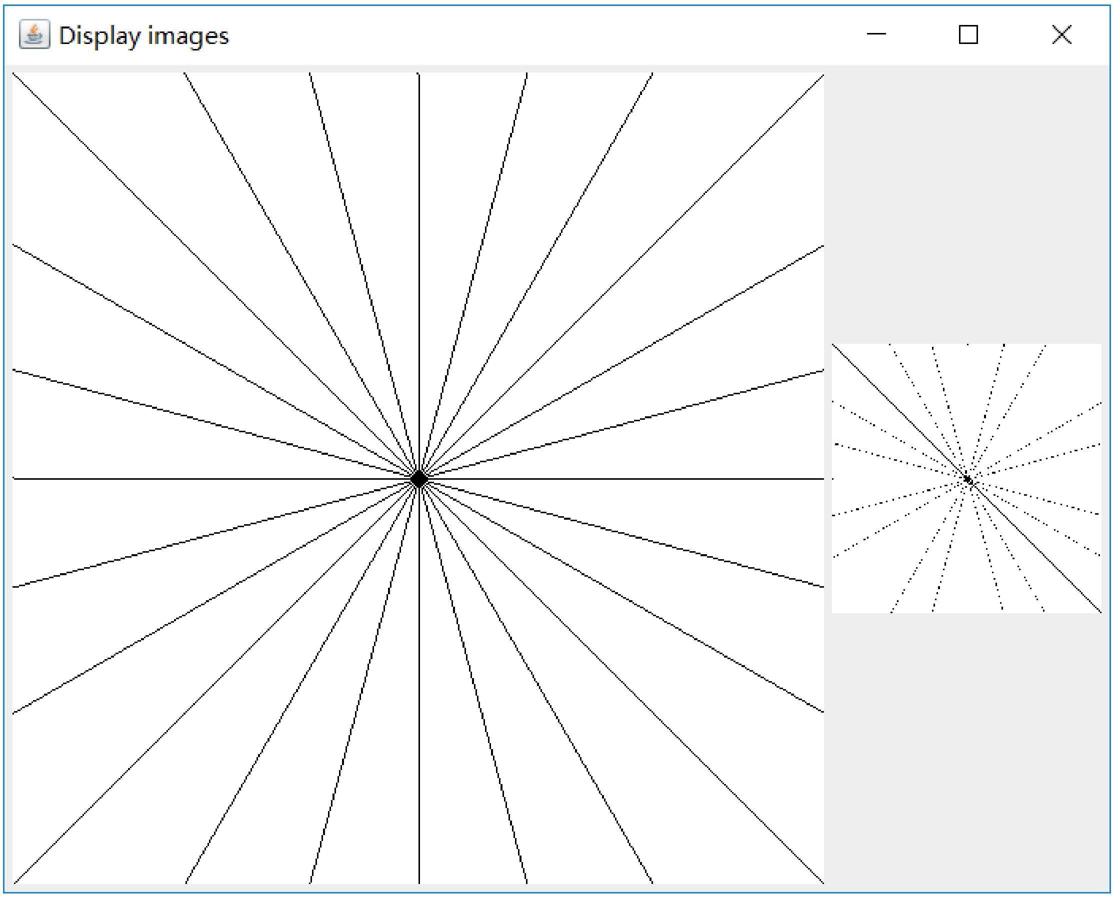
According to the results above, we can see that the effect of aliasing becomes more obvious while n increases (especially to lines which are none of horizontal, vertical or diagonal).

2. Let’s keep n=24.

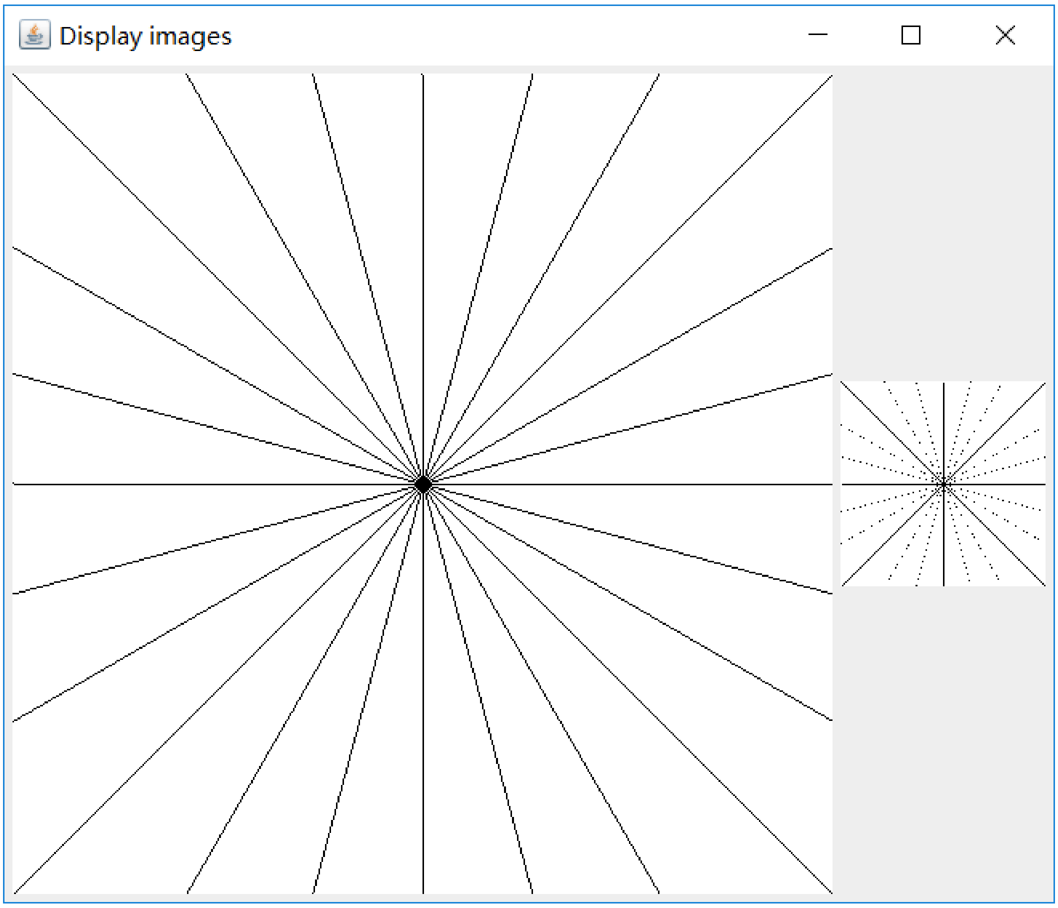
s=2.0:



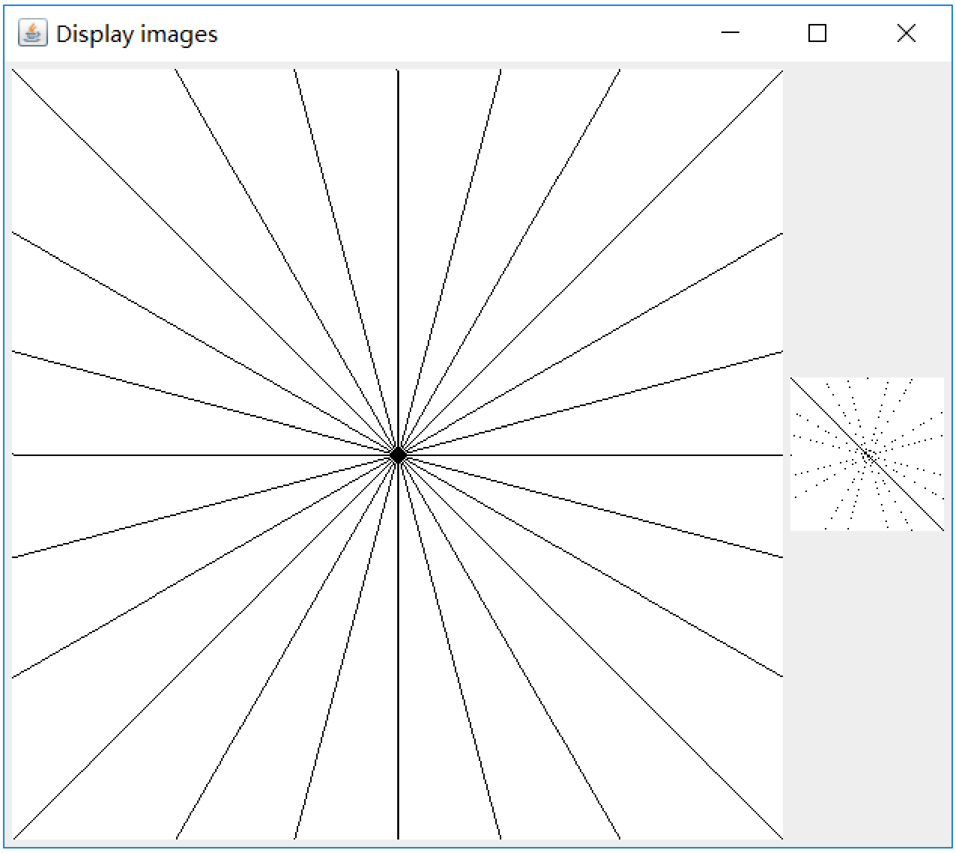
s=3.0:



s=4.0:



s=5.0:



According to the results above, we find that the aliasing effect becomes more obvious while s increases.

Part 2

In Part 2 n is set to 64 in all cases.

1. Theoretically the formula relating s (rotations per second), fps and os (rotations per second) should be:

$$os = \frac{f(\theta) * fps}{360}$$

where

$$\theta = \left(\frac{s * 360}{fps}\right)\%360, \qquad f(\theta) = \begin{cases} \theta, & if \ \theta \leq 180^{o} \\ 360^{o} - \theta, & else \end{cases}$$

if θ is less than 180^{o}, the rotational direction of output wheel should be as same as the input wheel. Otherwise, there is a temporal aliasing.

2. Theoretically, os=10 r/s. The program result shows that the output wheel rotate in the same direction as the input wheel.
3. Theoretically, os=6 r/s. The program result shows that the output wheel rotate in the reverse direction of the input wheel.
4. Theoretically, os=0. The program result shows that the output wheel does not rotate or only moves a little.
5. Theoretically, os=2 r/s. The program result shows that the output wheel rotate in the same direction as the input wheel.