Module 4. Lesson 2. Turtle. Loops

Confirmation of qualification

To get started with the working tasks, demonstrate your knowledge level.

Prove that you are ready for a "brainstorm"!

Confirmation of qualification

What is a pixel?

What type of graphics does the turtle module work with?

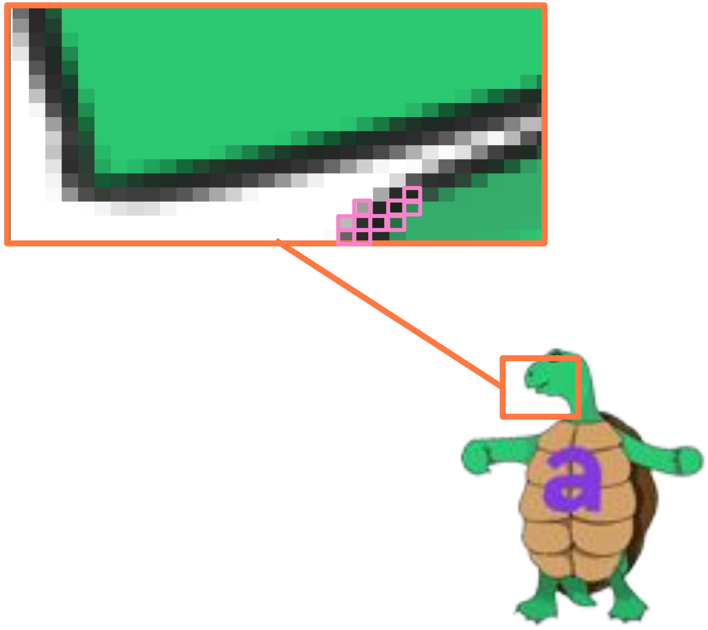
Confirmation of qualification

A pixel is

the smallest (indivisible) part of a graphic image

Bitmap is a collection of pixels.

A bitmap image is a collection of dots (pixels) used to display an image on a computer screen.



The turtle module works with bitmap graphics.

Confirmation of qualification

What is a turtle?

At which point does the turtle appear, and where does it move?

Confirmation of qualification

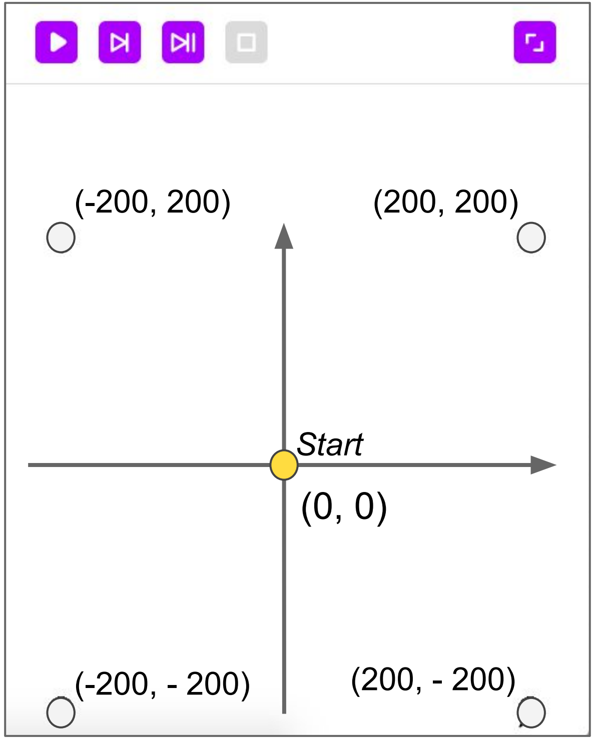
Turtle

is a command executor of the turtle module which draws graphic objects.

Part of the window where the executor is located is called the **coordinate plane**.

The turtle’s position on the plane is determined by two numbers – **coordinates.**

When the program starts, the turtle appears at the starting point (0, 0).



Confirmation of qualification

Which commands match the description?

Move the turtle 100 pixels **forward**	?
Rotate the turtle to the right by 100 degrees	?
Rotate the turtle to the left by 100 degrees	?
Change the thickness of the turtle’s pen by 5 pixels	?
Change the color of the turtle’s pen to orange	?

Confirmation of qualification

Which commands match the description?

Move the turtle 100 pixels **forward**	forward(100)
Rotate the turtle to the right by 100 degrees	right(100)
Rotate the turtle to the left by 100 degrees	left(100)
Change the thickness of the turtle's pen by 5 pixels	pensize(5)
Change the color of the turtle's pen to orange	color("orange")

Confirmation of qualification

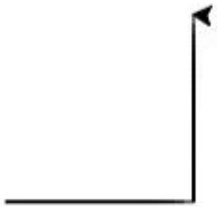
By which angle shall the turtle rotate to draw:

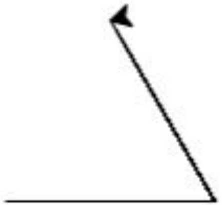
a) a square;
b) a triangle;
c) a hexagon?

Confirmation of qualification

By which angle shall the turtle rotate to draw:

a) a square;
b) a triangle;
c) a hexagon?

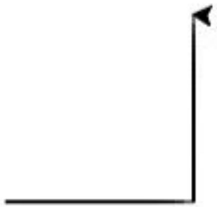


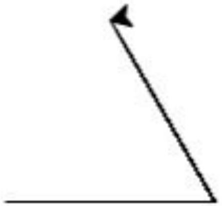


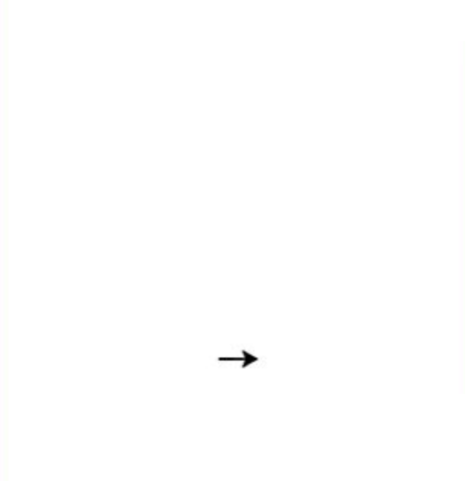
Confirmation of qualification

Turtle rotation angle:

a) square: 90 degrees;
b) triangle: 120 degrees;
c) hexagon: 60 degrees.







Confirmation of qualification

Which commands match the description?

Start filling with the current color	?
Stop filling with the current color	?
Pen up	?
Pen down	?
Move the turtle to the point with coordinates (-100, 40)	?

Confirmation of qualification

Which commands match the description?

Start filling with the current color	begin_fill()
Stop filling with the current color	end_fill()
Pen up	penup()
Pen down	pendown()
Move the turtle to the point with coordinates (-100, 40)	goto(-100, 40)

Confirmation of qualification

Qualification is confirmed!

Great, you are ready to brainstorm and complete your work task!

Confirmation of qualification

Module 4. Lesson 2. Turtle. Loops

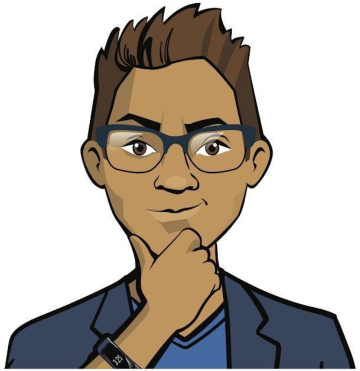
“Brainstorm”:

Drawing shapes with loops

Drawing shapes with loops

Consider a few work tasks.

Each one can be solved in some ways.
Still, we will try to find the optimal one.



“Brainstorm”

A couple of useful commands

Command	*Value*
speed(<number in 0 - 10 range>)	Change the turtle speed
hideturtle()	Hide the executor turtle from the screen (only the picture remains)

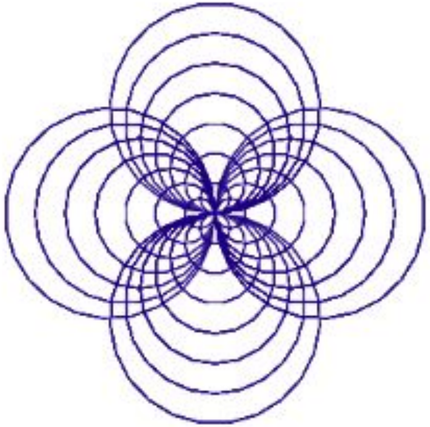
Speed 3

Speed 10

“Brainstorm”

Consider the task

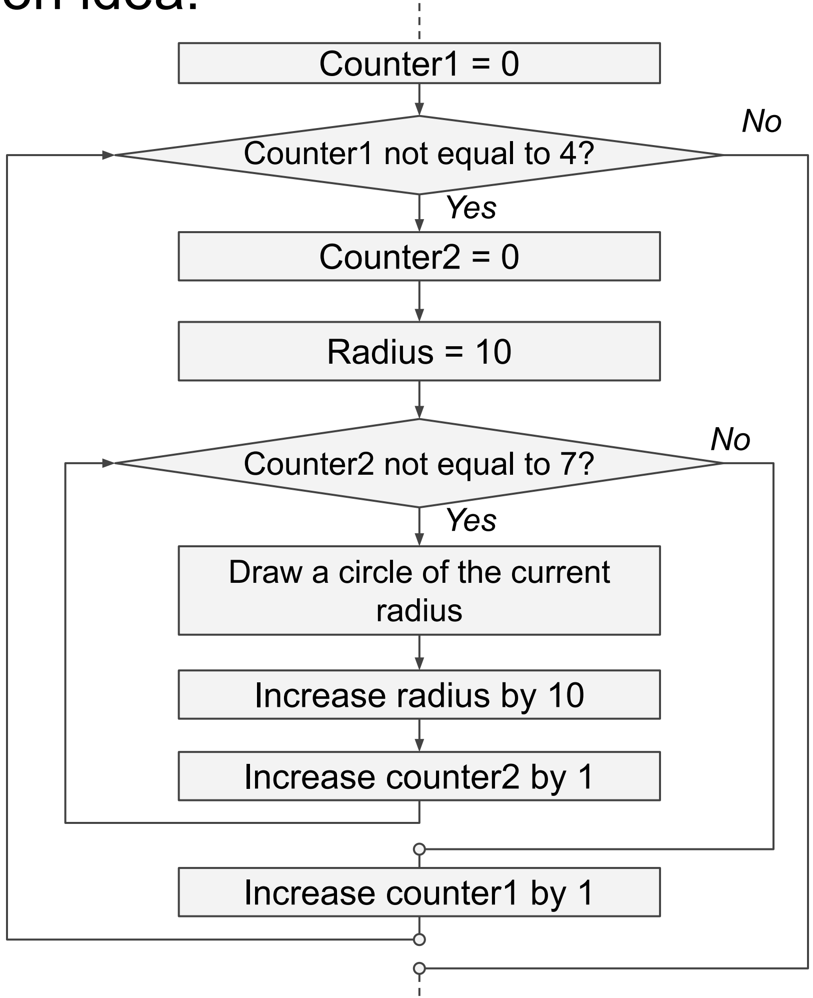
Task. Program an executor so that it draws the Illusion Museum logo. The shape consists of four parts, each containing seven circles. The radius of the first circle is 10. The radius of each subsequent circle is greater by 10. Hide the executor when the image drawing is complete.



What commands should we use? Which solution will be optimal?

“Brainstorm”

Solution idea:

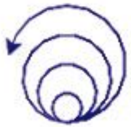


"Brainstorm"

Sample code:

```
from turtle import *
def draw_set():
   size = 10
   for i in range(7):
       circle(size)
       size += 10
```

Drawing a set of seven circles



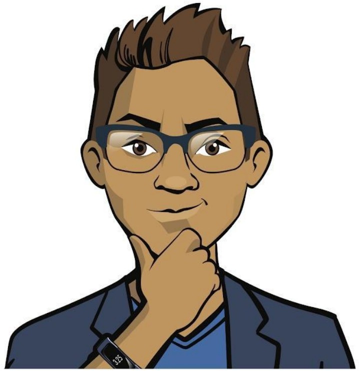
```
speed(10)
color("navy")

for i in range(4):
   draw_set()
   left(90)
hideturtle()
exitonclick()
```

Drawing four sets with a 90-degree rotation

“Brainstorm”

Before we continue:

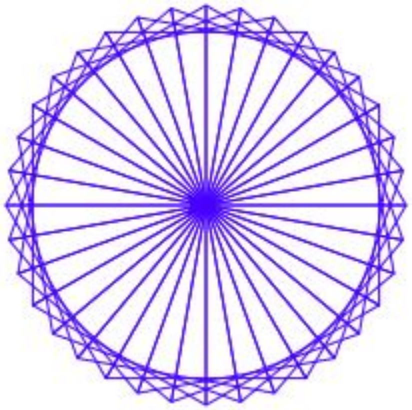
1. **How can we change the program** to have 5 petals in the flower on the logo instead of 4 petals?
2. The customer wants to place a nice animated rendering of the logo on the website of Illusion Museum. **How can we change the program** to make the logo draw slower?



"Brainstorm"

Consider the task

Task. “Dream” amusement park also intends to print an ad containing its panoramic-wheel logo. Wheel arm length is 100. The angle between the arms is 10 degrees. The image color is blue. Hide the executor when the image drawing is complete.



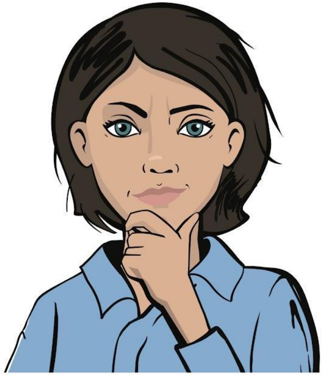
Looks nice, but complicated... How can we code such a pattern?

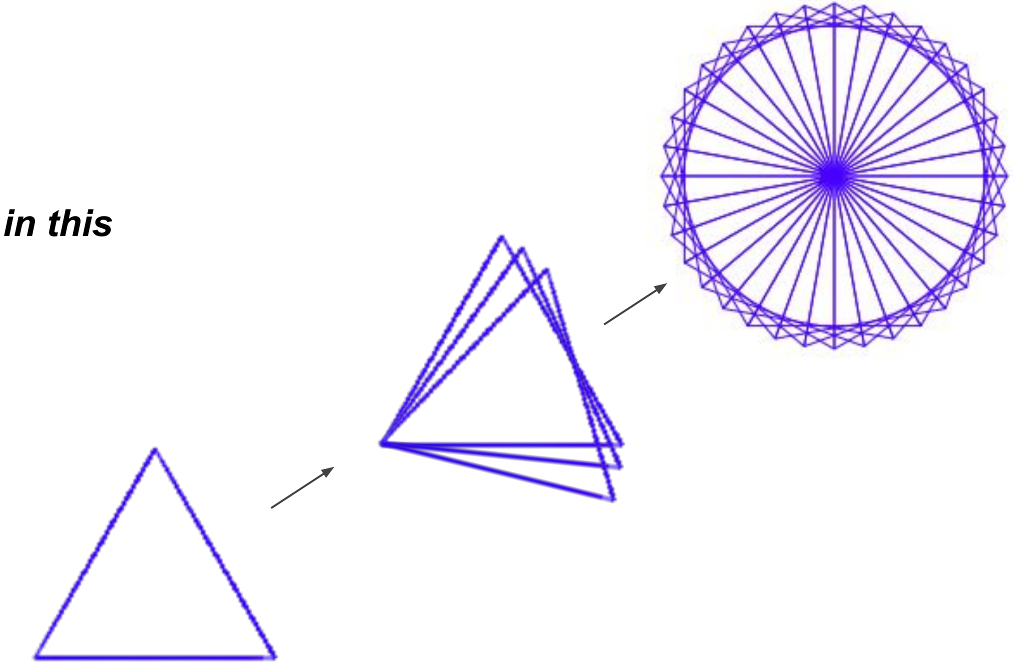
“Brainstorm”

Consider the task

Task. “Dream” amusement park also intends to print an ad containing its panoramic-wheel logo. Wheel arm length is 100. The angle between the arms is 10 degrees. The image color is blue. Hide the executor when the image drawing is complete.

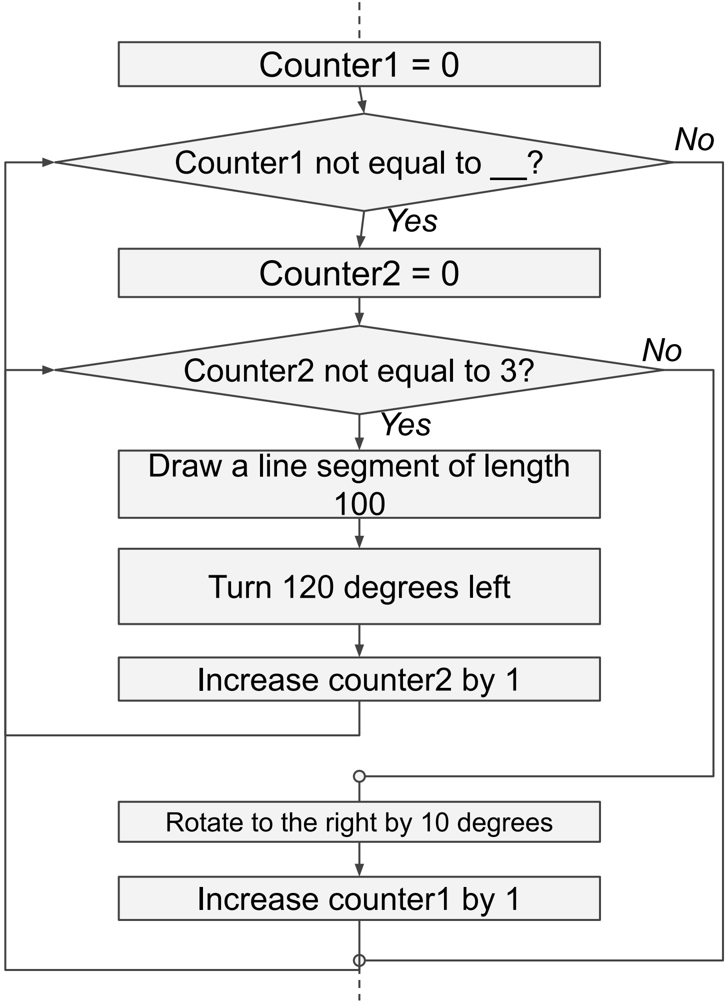
I think I can see triangles in this picture...





“Brainstorm”

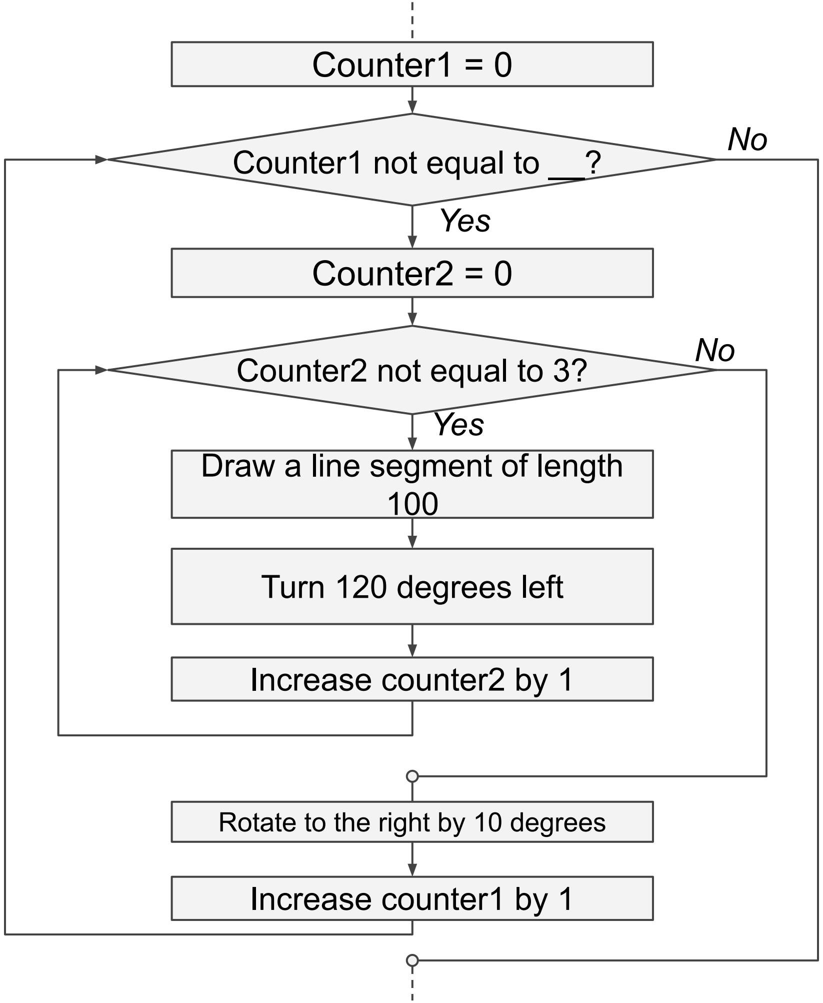
Solution idea:



How many times shall the outer loop execute?

“Brainstorm”

Solution idea:



Counter1 = 0

Counter1 not equal to __?

No

Yes

Counter2 = 0

Counter2 not equal to 3?

No

Yes

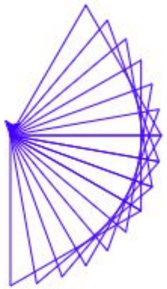
Draw a line segment of length 100

Turn 120 degrees left

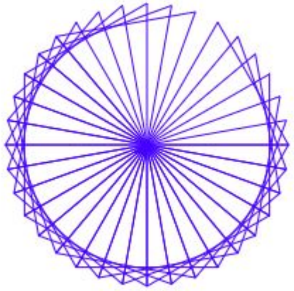
Increase counter2 by 1

Rotate to the right by 10 degrees

Increase counter1 by 1



10 times



30 times

Let’s try to guess...

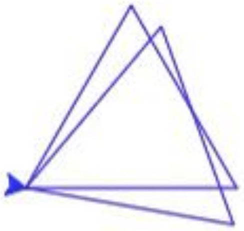
“Brainstorm”

Sample code:

```
from turtle import *


def draw_triangle():
    color("blue")
    for i in range(3):
        forward(100)
        left(120)
speed(10) #increasing the speed
for i in range(36):
    draw_triangle()
    right(10)
hideturtle()
exitonclick()
```

Drawing a triangle

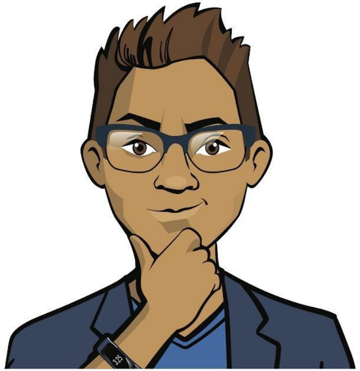
Drawing **36** triangles with a 10-degree rotation



“Brainstorm”

Before we continue:

1. **How can we change the program** to have more arms in the panoramic wheel than now? And how can we get less?
2. The customer wants to have a more friendly logo. To do this, we need to paint the panoramic wheel two colors: red and orange. **How can we do this?**



“Brainstorm”

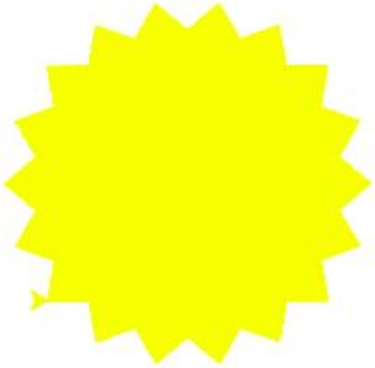
Module 4. Lesson 2. Turtle. Loops

“Brainstorm”:

Drawing shapes in loops

Let’s do another task related to a star

Task. Sun is the only star in the Solar system. Write a program that draws it like in the picture. The length of the section at the baseline of the Sun is 150. The number of rays is 18. The color is yellow. Hide the executor when the work is done.



We have already solved a similar task with a pentagonal star.

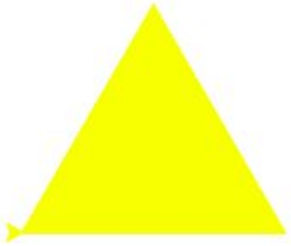
How can we code the turtle now? <u>What will the rotation angle be</u>?

“Brainstorm”

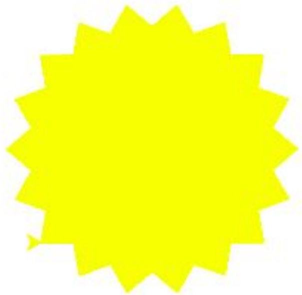
Let’s do another task related to a star

Task. Sun is the only star in the Solar system. Write a program that draws it like in the picture. The length of the section at the baseline of the Sun is 150. The number of rays is 18. The color is yellow. Hide the executor when the work is done.

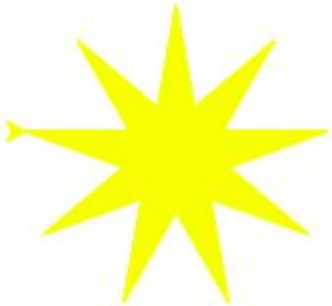
Similar to the pentagonal star, let’s reduce the task to drawing a section and rotation by the same angle. Let’s figure out this angle.



150-pixel section,
<u>120-degree</u> angle



150-pixel section,
<u>100-degree</u> angle



150-pixel section,
<u>200-degree</u> angle

“Brainstorm”

Sample code:

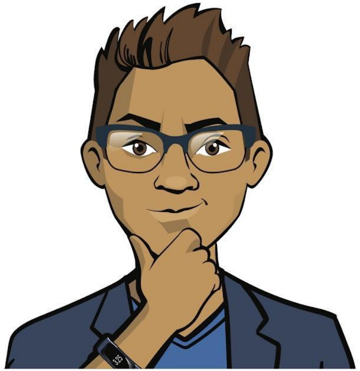
```
from turtle import *
def sun ():
    begin_fill()
    for i in range(18):
        forward(150)
        left(100)
    end_fill()


color("yellow")
speed(10)
sun()


exitonclick()
```

Drawing the Sun. 18
vertices ->
18 sections

“Brainstorm”

Before we continue:

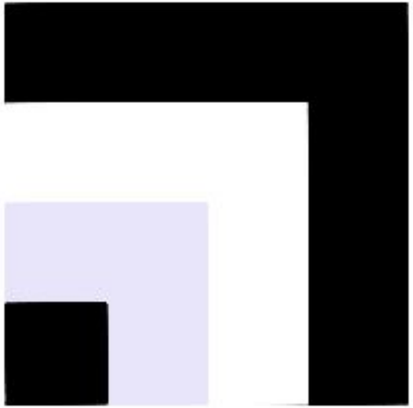
1. **Is it possible** to code a symmetric star similarly by repeating the actions in a loop 11 times? 7 times? Why?
2. The customer wants to place a few yellow stars on the banner. **How do we complete the program**?



"Brainstorm"

Let’s do one more task

Task. A black-and-white movie festival will take place in the city very soon. Code a pattern for the walls of a movie hall where the festival will be held. Start drawing at (-50, -50). Square colors are black, lavender, white, black. Hide the executor when the work is done.

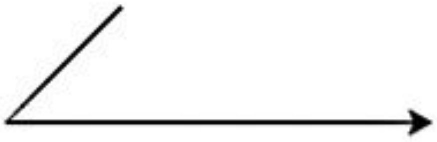


How to write such a program?

“Brainstorm”

Sample code:

```
from turtle import *
goto(-50, -50)
def draw_square(length, cur_color):
    color(cur_color)
    begin_fill()
    for i in range(4):
        forward(length)
        left(90)
    end_fill()
draw_square(200, 'black')
draw_square(150, 'white')
draw_square(100, 'lavender')
draw_square(50, 'black')
hideturtle()
exitonclick()
```

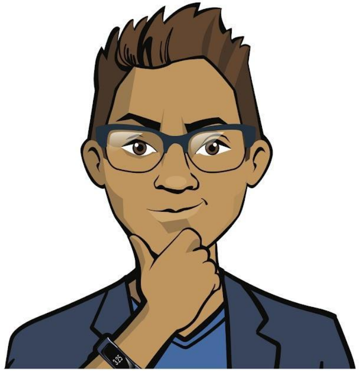


“Brainstorm”

Before we continue:

1. The customer has asked to add another square with grey, 250-pixel side.
   How do we complete the program?

1. **How do we add** a violet, 5-pixel thick outer frame to the combination of squares?



“Brainstorm”