CS 170: Introduction to Computer Science I – Spring 2023
Homework Assignment #1
Due by Tuesday, January 31, 2023 at 4:00 pm

Submission instructions

Submit your assignment through the QTest system, using the exam ID: **CS170.hw1**.
No email submissions are accepted. No late submissions are accepted.

General instructions and hints

In those problems asking you to write a method, always call the method several times to test that it works properly with a variety of different values for the parameters. Test it on more examples than the ones shown in this handout. Even if the problem asks you for just one method, you can always write additional helper methods to simplify and organize your code. Make sure you write comments to explain your code.

Comment requirements: Always comment the top of each method (what the method does, the meaning of the input parameters, the meaning of the output value). Write comments within the methods to explain the strategy you are using to solve the problem, and to clarify blocks of code that may be difficult to understand.

Problem 1: Beginning of the semester survey (2 points).

Go to the following website and complete the survey.

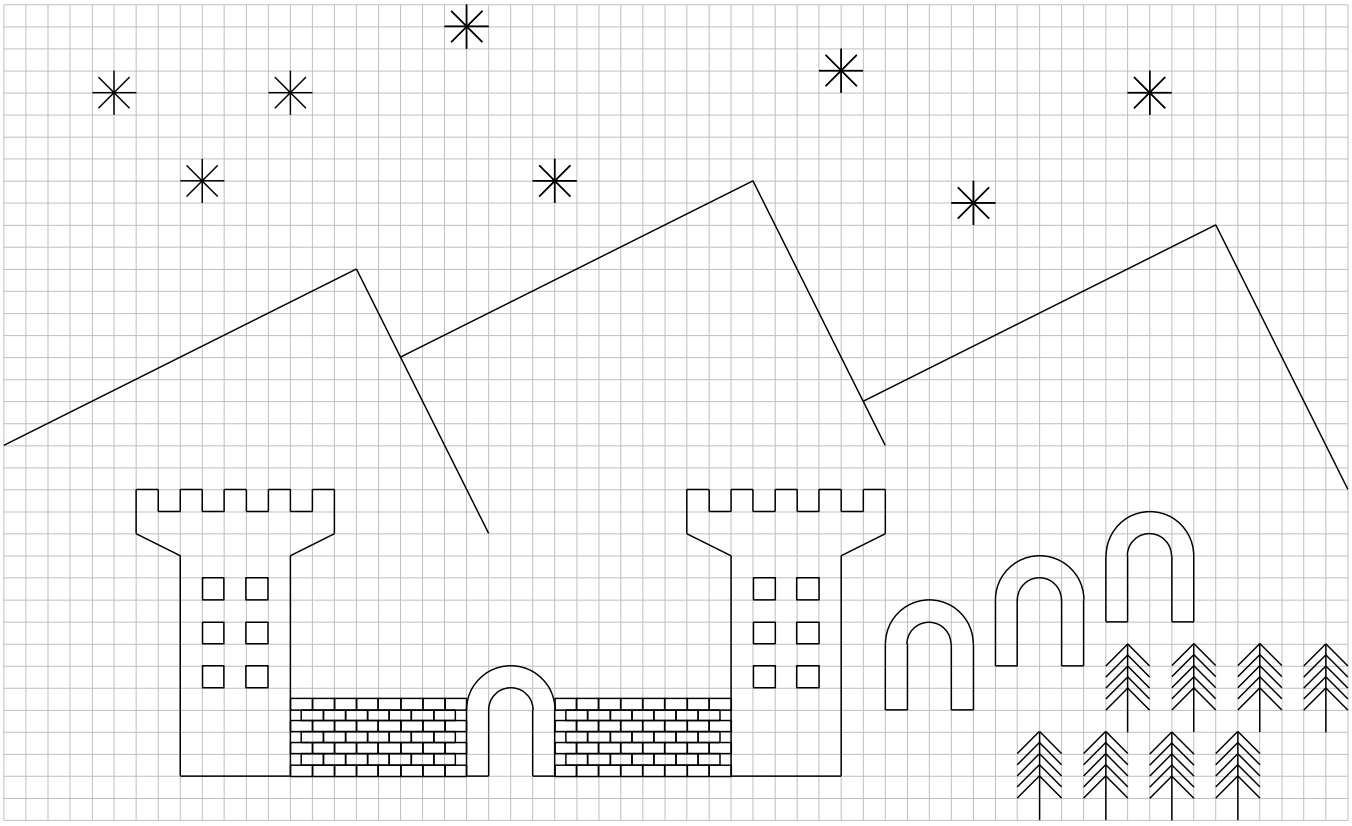
https://docs.google.com/forms/d/e/1FAIpQLSe9IihyNof67VFvRLeW56ADvD86haG2NB96QZ3BssfdHjnq6Q/viewform

The answers to the survey will not be graded: you will receive credit just for completing it. Please answer the questions honestly. The purpose of this survey is to know the students in this course, and possibly adapt the course based on these answers.

Problem 1 rubric:
2 points for completing the survey.

Problem 2: Medieval town (18 points).

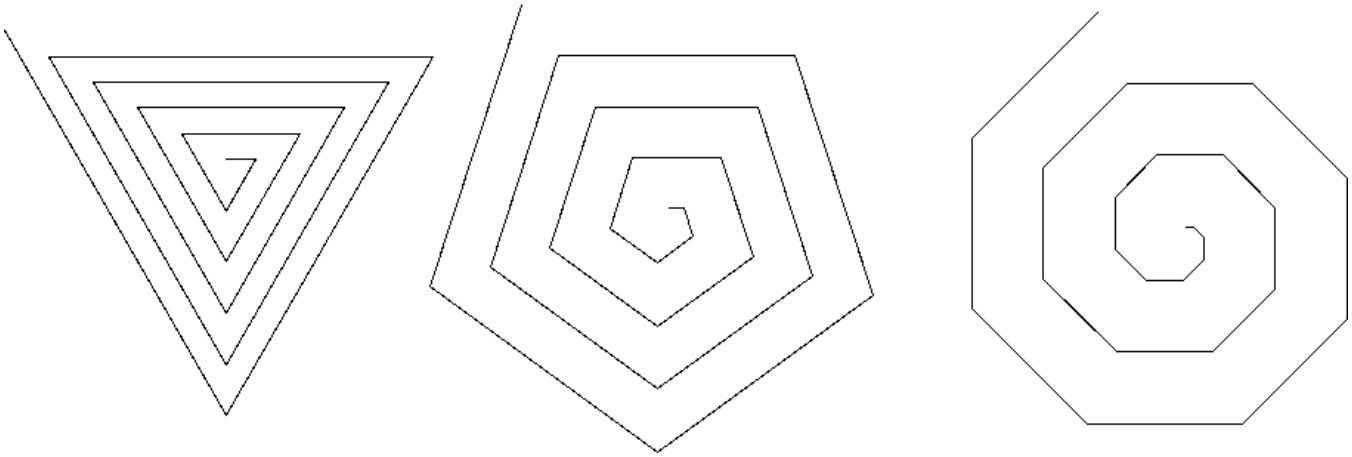
Write a Java program that draws the figure below using turtle graphics. Your program will not draw the grid, which is shown here only to help you measure the size of the figure (each grid square is 10 turtle steps long). Make sure your code is modular: organize the different components to be drawn in appropriate methods, then call those methods to compose the entire drawing. Your code should use iteration (loops) wherever appropriate (i.e., minimize copy and paste). Make sure that your methods have meaningful names, and you include appropriate comments in your code. Feel free to use turtle's coloring commands to make your drawing even more interesting. Hint for the arcs: `Math.PI` contains the constant number PI (3.1415...). The built-in class `Math` contains many useful trigonometry functions:
https://docs.oracle.com/javase/8/docs/api/java/lang/Math.html
You can approximate a circle with a polygon with many sides. This website describes the relationship between the radius of a polygon and the length of its side:
http://www.mathopenref.com/polygonradius.html



Problem 2 rubric:
programs that do not compile get zero points.
+3 for each object correctly drawn in the correct position (towers, walls, arches, trees, mountains, stars) (only 2 points if the object has imperfections, 0 if it is not a good likeness for the object).
-4 if the code does not use for loops where appropriate
-4 if the code is not modularized using methods
-2 if the methods don't have meaningful names
-2 if there are no comments, insufficient comments, or bad usage of comments

Problem 3: Polyspiral (6 points)

Write the method `polyspiral(Turtle t, int n, double base, int rounds)` that makes the turtle draw a spiral like those shown in the figure. The parameter `n` specifies the polygonal shape of the spiral (3 is triangular, 4 is quadrilateral, etc.). The parameter `base` indicates the size of the smallest side of the spiral. The parameter `rounds` specifies the number of revolutions of the spiral. See examples below.

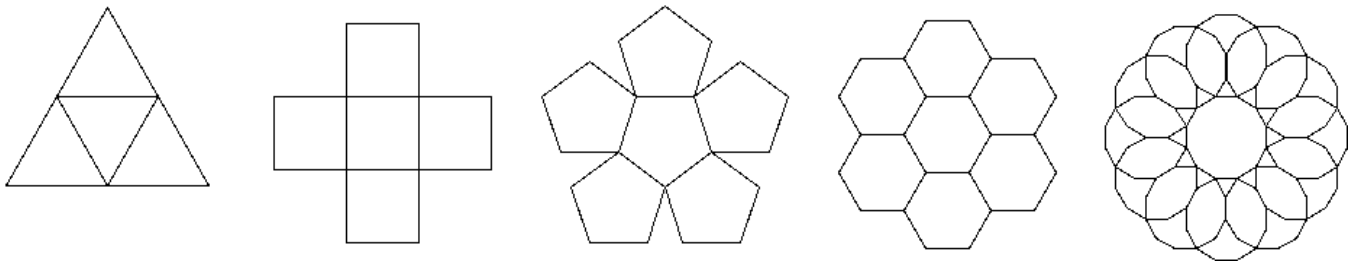


`polyspiral(t, 3, 20, 5)` `polyspiral(t, 5, 10, 4)` `polyspiral(t, 8, 5, 3)`

Rubric for Problems 3-7:
programs that do not compile get zero points.
partial credit for partially correct running solutions (3 points if there is one logical error, 0 for more than one logical error)
-1 if the method name and arguments do not match what is stated in the problem description.
-1 if there are no test cases
-1 if there are no comments, insufficient comments, or bad usage of comments
-1 if the turtle does not go back to the original position

Problem 4: Polywheel (6 points)

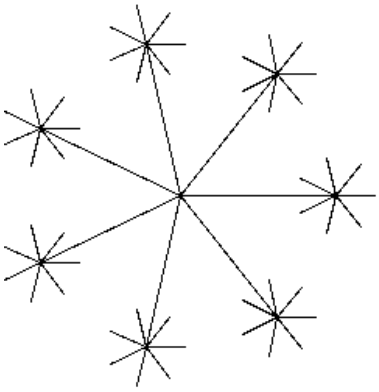
Write the method `polywheel(Turtle t, int numSides, double length)` that makes the turtle draw a polygonal wheel like those shown in the figure. The parameter `numSides` specifies the number of sides of the base polygon. The parameter `length` indicates the size of the side of each polygon. See examples below.



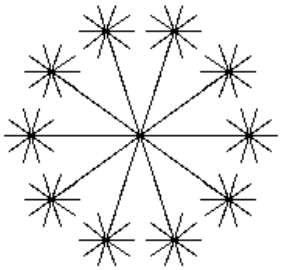
`polywheel(t,3,70)` `polywheel(t,4,50)` `polywheel(t,5,40)` `polywheel(t,6,30)` `polywheel(t,12,15)`

Problem 5: Multistar (6 points)

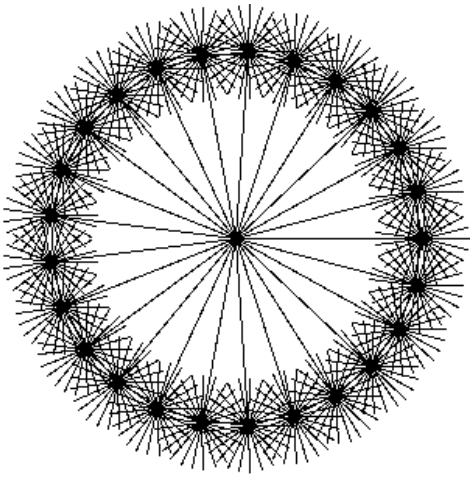
Write the method `multistar(Turtle t, int n, double length)` that makes the turtle draw a star like those shown in the figure. The parameter `n` specifies the number of rays. The parameter `length` indicates the size of the long rays (the short rays are 1/4 the length of the long rays). See examples below.



`multistar(t, 7, 100)`



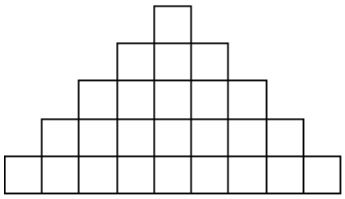
`multistar(t, 10, 70)`



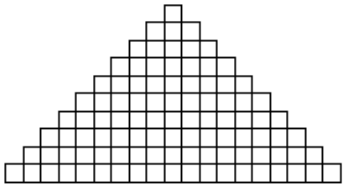
`multistar(t, 25, 120)`

Problem 6: Pyramid (6 points)

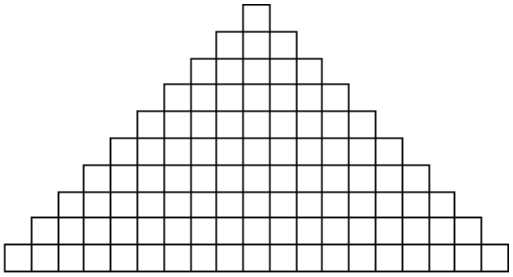
Write the method `pyramid(Turtle t, double base, int levels)` that makes the turtle draw a pyramid like those shown in the figure. The parameter `base` specifies the total length of the base of the pyramid. The parameter `levels` indicates the number of layers of blocks that build up the pyramid. See examples below.



`pyramid(t, 200, 5)`



`pyramid(t, 200, 10)`



`pyramid(t, 300, 10)`

Problem 7: Houseline (6 points)

Write the method `houseline(Turtle t, int numHouses)` that makes the turtle draw a line of houses where the second house is half as big as the first one, the third one is 1/3 of the first one, the fourth one is 1/4 of the first one, and so on. The parameter `numHouses` is the total number of houses in the line. See example below. You can use the house code from the course website, adding a parameter to scale the size of a house by an arbitrary factor. The first house is twice as big as the house from the course website. The space between each pair of houses is equal to the size of a window of the first house in that pair.

`houseline(t, 20)`

Bonus points: Early submission

If you submit the entire homework no later than 48 hours before the deadline, and the total score on the rest of this homework assignment is at least 20 points, you will receive 2 bonus points. The bonus points will be added to the total score of this homework assignment.

Good luck and have fun!