CS1010E Programming Methodology

Semester 2 2019/2020

Assignment 01: Expressions

Release Date: 20/01/2020 00:00

Due Date: 26/01/2020 23:59 Total Experience: 100 Exp

Required Files

- `assignment1_template.py`

Notes

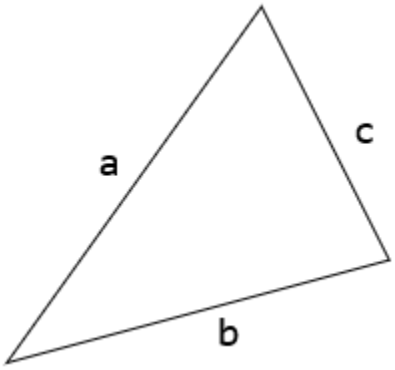
- The functions `sqrt` and `sin` has been imported for you using `from math import sqrt, sin`. You can use them freely in this exercise.
- You are NOT allowed to have import statement in your answer as this may interfere with the test cases on Coursemology.
- You are NOT allowed to change the input arguments to the functions (e.g., *number of arguments*).
- You are NOT allowed to change the function name.

Task 1: from Formula to Expression

In the first task of this assignment, you are going to be given formulas. Your job is simple, you are to rewrite the formula into Python expression. This will test you on your basic Python syntax.

Task 1A: Heron's Formula [30 Exp]

Heron is named after Heron of Alexandria. He was a mathematician and engineer who lived from around 10 AD to 70 AD. Heron's formula is a method for calculating the area of triangle given the lengths of all three sides as shown below. Let's call the sides `a`, `b`, and `c`.



Heron's formula is given as:

$$area = \sqrt{p(p-a)(p-b)(p-c)}$$

where p is half the perimeter: $p = \frac{a+b+c}{2}$

Your task is to write the function `heron(a, b, c)`. The function accepts three numbers `a`, `b`, and `c` that corresponds to the length of the three sides of a triangle. The function returns the `area` of the triangle.

Assumptions

- `a >= 0, b >= 0, c >= 0`
- `a + b >= c, a + c >= b, b + c >= a`

Note that the second assumption is called triangle inequality. It simply means that the value we are giving will always form a triangle. You can test yourself, you cannot create a triangle such that the lengths are 1, 1, and 3.

Coursemology Notes

Write your answer within the function `heron(a, b, c)` and replace the statement `pass` with your answer. Do not forget to include `return` keyword. Make sure you check that the indentations are consistent. As long as your answer are correct within 3 decimal places, we will accept your answer.

Example Runs

Consider the right-angled triangle with sides of length 1, 1, and $\sqrt{2}$ where 1 is the base and 1 is the height. The area of the triangle is then exactly 0.5. Using the Heron's formula above, you should also come up with the answer 0.5 (or something very close to 0.5).

```
>>> heron(1, 1, sqrt(2))
0.49999999999999983

>>> heron(3, 4, 5)
6.0
```

Task 1B: Projectile Motion [30 Exp]

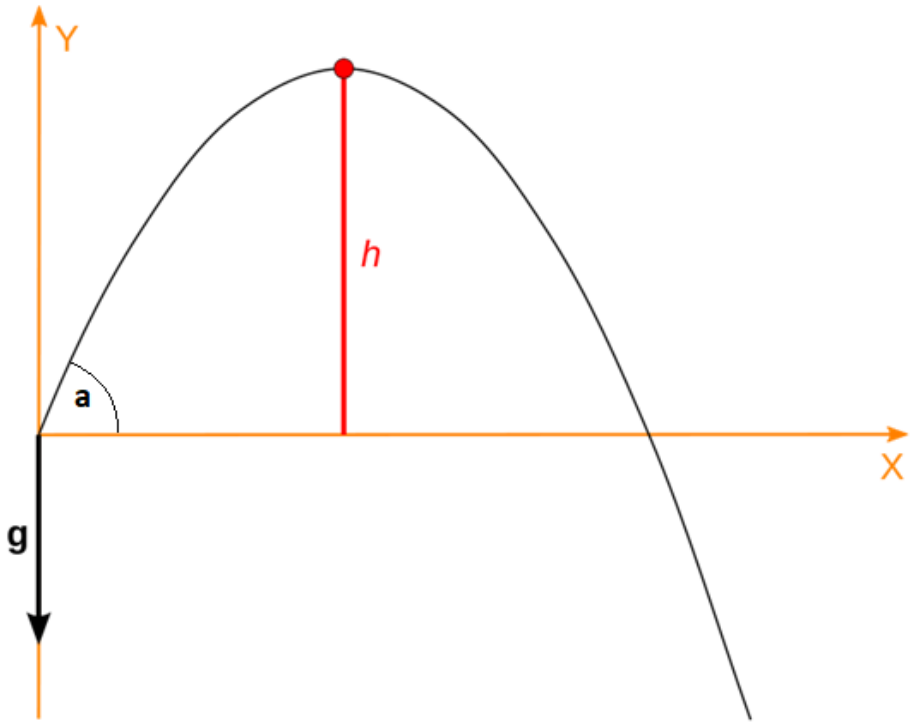


Figure 1: Image modified from Wikipedia.

A projectile motion is a form of motion experienced by a projectile near the Earth's surface. The projectile moves along a curved path under the action of gravity only. We assume the effect of air resistance to be negligible. This curved path is in fact a parabola. We can find the maximum height of the projectile by the equation:

$$h = \frac{v_0^2 \sin^2(a)}{2g}$$

where a is the angle in degrees, v_0 is the initial velocity in m/s, and $g = 9.8$.

Your task is to write the function `initial_velocity(max_height, angle)`. The function accepts two numbers `max_height` and `angle` given in degrees that corresponds to h and a respectively. The function returns the initial velocity v_0.

Assumptions

- `max_height >= 0`
- `0 < angle < 180`

Coursemology Notes

Write your answer within the function `initial_velocity(max_height, angle)` and replace the statement `pass` with your answer. Do not forget to include `return` keyword. Make sure you check that the indentations are consistent. As long as your answer are correct within 3 decimal places, we will accept your answer.

Example Runs

Given an initial velocity of 20 m/s and angle of 45 degrees, the maximum height is 10.20408 m.

```
>>> initial_velocity(10.204081632653063, 45)
20.0

>>> initial_velocity(12.085714354185443, 20)
45.0
```

Hints

Consider the function `sin`. What unit does the function `sin` takes in? The formula for converting degrees `deg` to radians `rad` is given as follows:

$$\text{rad} = \text{deg} \times \frac{\pi}{180°}$$

Furthermore, you may want to rearrange the formula into the form of $v_0 = \cdots$.

Task 1C [Hard]: Inverse Heron [10 Exp]

We already know that Heron's formula is given as:

$$area = \sqrt{p(p-a)(p-b)(p-c)}$$

where p is half the perimeter: $p = \frac{a+b+c}{2}$

Hopefully, computing the area given the length of the three sides of a triangle should not be a difficult task for you. In this task, we want to compute somewhat the opposite. What you will be given is basically the length of **two sides** as well as the **area**. Then you should compute the length of the missing side.

This task is non-trivial. To simplify the problem, we are going to assume that the *length of the two sides that are given are always equal*. For instance, if we let $a = c$, then we have two possibilities as below:

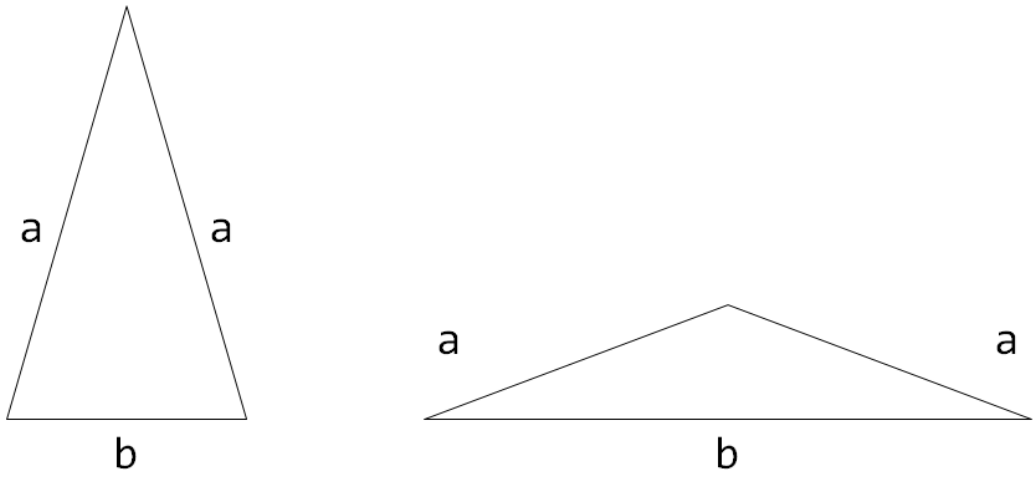


Figure 2: Two possible triangles with sides of length a, a, and b with the same area.

We call the triangle on the left as an acute angle triangle and the triangle on the right as an obtuse angle triangle. In some instances, if the angle is exactly 45°, then the acute angle and obtuse angle triangles are the same and they become the right-angled triangle.

Your task is to write the function `inverse_acute_heron(a, area)`. The function accepts two numbers `a` and `area` corresponding to the length a and the area of triangle respectively. The function returns the length of the missing side `b` corresponding to an *acute angle triangle*.

Assumptions

- `a > 0`
- `a + a >= b`
- `area > 0`

Again, the second assumption corresponds to the triangle inequality.

Coursemology Notes

Write your answer within the function `inverse_acute_heron(a, area)` and replace the statement `pass` with your answer. Do not forget to include `return` keyword. Make sure you check that the indentations are consistent. As long as your answer are correct within 3 decimal places, we will accept your answer.

Example Runs

Using the number from Task 1A, we can deduce that calling `inverse_acute_heron(1, 0.5)` should give us $\sqrt{2}$ since this gives us the right-angled triangle as described in Task 1A.

```
>>> inverse_acute_heron(1, 0.5)
1.4142135623730951
```

```
>>> inverse_acute_heron(1, heron(1, 1, 1))
0.9999999999999999
```

Hints

Similar to Task 1B, you should rewrite the Heron's formula into some other forms. It will be beneficial to simply let $L = b^2$. This will allow you to solve some form of quadratic equation, which you can simply use a modified version of `solve_qe` to solve.

A good starting point is

$$area^2 = \left(\frac{2a+b}{2}\right)\left(\frac{2a+b}{2} - a\right)\left(\frac{2a+b}{2} - a\right)\left(\frac{2a+b}{2} - b\right)$$

Task 2: Simple Turtle

You might have learned some of the Python's Turtle functions in lecture/tutorial. Here are a few questions to really test your understanding of how they work, and a very simple drawing to get you acquainted with drawing with Turtle!

Some of the functions covered are shown below:

- `pd()`: Pen down. Any movements thereafter will be traced.
- `pu()`: Pen up. Any movements thereafter will NOT be traced.
- `fd(d)`: Moves the turtle forward a number of units specified by distance `d`.
- `bk(d)`: Moves the turtle backward a number of units specified by distance `d`.
- `rt(d)`: Turns the turtle right by a certain number of degrees `d` specified.
- `lt(d)`: Turns the turtle left by a certain number of degrees `d` specified.

Note that the direction functions `fd`, `bk`, `rt`, and `lt` all are given in the shorthand format. The longer forms are `forward`, `backward`, `right`, and `left` respectively. You may use whichever function you prefer.

Warmup Task: [*Ungraded*]

In the following sequences of function calls, try to imagine what will be drawn. Then, run the calls yourself on Python. Did it draw what you had expected?

```
fd(100)
bk(100)
fd(100)
bk(100)
```

```
bk(100)
pu()
lt(90)
fd(100)
rt(90)
fd(100)
```

```
bk(100)
lt(90)
fd(100)
rt(90)
fd(100)
```

```
rt(90)
lt(180)
rt(315)
bk(100)
rt(90)
bk(100)
```

Restrictions

- You are only allowed to use the functions you have learned so far.
  - `pd`, `pu`, `fd`, `bk`, `rt`, `lt`, `forward`, `backward`, `right`, `left`.
- You should NOT make any redundant movement. For instance:
  - Going forward, backward, and then forward again.
  - Making left turn 45° followed by another left turn 45° to make a left turn 90°.
- You should NOT lift the pen up (i.e., `pu()`).

Task 2A: One Star [30 Exp]

Using only the functions that you have learnt so far, call for a sequence of Turtle commands to draw a *star* with each line having distance of *300 units*. Furthermore, the start and end point of the star is one of the pointy end that is located at the initial point of the turtle as shown in the image below:

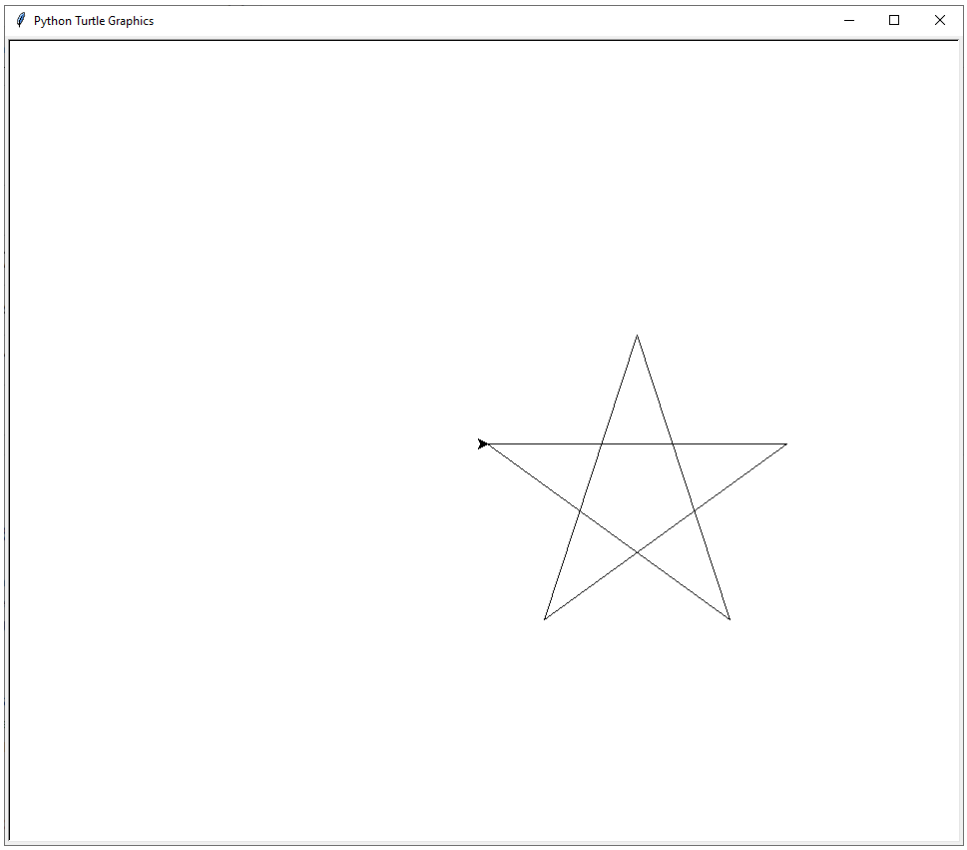


Figure 3: Turtle drawing of a star with sides of length 300 units.

Coursemology Note

Your answer on Coursemology should only be a sequence of direction or movement functions similar to the following but obviously longer. You should not include any `import` statement.

```
left(90)
right(90)
forward(100)
backward(100)
```