Note: You will need to use the trig formula:

\[
\text{length} = \frac{\text{height}}{\sin(\text{angle})}
\]

The `math` module `\text{sin}(\cdot)` function takes its input in radians. You will thus need to convert the angle given in degrees to the angle given in radians using:

\[
\text{radians} = \frac{\pi \times \text{degrees}}{180}
\]

2.21 Write an expression involving a three-letter string `s` that evaluates to a string whose characters are the characters of `s` in reverse order. If `s` is `’top’`, the expression should evaluate to `’pot’`.

2.22 Write an expression involving string `s` containing the last and first name of a person—separated by a blank space—that evaluates to the person’s initials. If the string contained my first and last name, the expression would evaluate to `’LP’`.

2.23 The range of a list of numbers is the largest difference between any two numbers in the list. Write a Python expression that computes the range of a list of numbers `lst`. If the list `lst` is, say, `[3, 7, -2, 12]`, the expression should evaluate to 14 (the difference between 12 and -2).

2.24 Write the relevant Python expression or statement, involving a list of numbers `lst` and using list operators and methods for these specifications:
   (a) An expression that evaluates to the index of the middle element of `lst`
   (b) An expression that evaluates to the middle element of `lst`
   (c) A statement that sorts the list `lst` in descending order
   (d) A statement that removes the first number of list `lst` and puts it at the end

2.25 Add a pair of parentheses to each expression so that it evaluates to `True`.
   (a) `0 == 1 == 2`
   (b) `2 + 3 == 4 + 5 == 7`
   (c) `1 < -1 == 3 > 4`
   For each expression, explain in what order the operators were evaluated.

2.26 Write Python statements that draw a square of side length 100 pixels using Turtle graphics. Make sure you import the module `turtle` first. Your first two and last statement should be as shown:

```python
>>> s = turtle.Screen()  # create screen
>>> t = turtle.Turtle()  # create turtle
...
# now write a sequence of statements
# that draw the square
>>> s.bye()  # delete the Screen when done
```

2.27 Using the approach from Problem 2.26, write Python statements that draw a diamond of side length 100 pixels using Turtle graphics.