Textbook Assignment 1

Maximum earnable: 120 pt.
Due: 11:59 PM October 5, 2023

- Read the assignment carefully. *You will need to* ***write and execute several Python scripts****; and* ***submit your code work together with their results****.*
- For this assignment, you are required to **submit a report** and ***Python* code** *using a provided template* (in the *.py* format).
  - ○ Make sure your code produces the same results as the solutions included in your report; **otherwise, you will get penalties.**
- You are **allowed to re-use any of the code snippets from the lecture slides** while developing solutions to the problems.
- This is an individual work;

  ○ Submitting assignments or program codes written by others or acquired from the internet without explicit approval of the professor is regarded as cheating.

  ○ Showing or lending one's own homework to other student is also considered cheating that disturbs fair evaluation and hinders the academic achievement of the other student.

  ○ It is regarded as cheating if two or more students conduct their homework together and submit it individually when the homework is not a group assignment.

- When finished, submit your work to *Google Classroom.*

1. (2 pt. each) Use the eval() function to evaluate these strings as Python expressions. Which evaluations result in an error? Explain why.

Include your solutions in the report. Also, implement the solution in Python using the supplied file (Q1.py). Make sure all the results are reproducible for grading.

(a) '2 * 3 + 1'

(b) 'hello'

(c) "'hello' + ' ' + 'world!'" (d) "'ASCII'.count('I')"

(d) "'ASCII'.count('I')"

(e) 'x = 5'

2. (10 pt.) Implement a program that requests a nonempty list from the user and prints on the screen a message giving the first and last element of the list.

Include your solutions and explain your code in the report. Also, implement the solution in Python using the supplied file (Q2.py). Make sure all the results are reproducible for grading.

```
>>>
Enter a list: [3, 5, 7, 9]
The first list element is 3
The last list element is 9
```

3. (10 pt.) Implement a program that requests four numbers (integer or floating-point) from the user. Your program should compute the average of the first three numbers and compare the average to the fourth number. If they are equal, your program should print 'Equal' on the screen.

Include your solutions and explain your code in the report. Also, implement the solution in Python using the supplied file (Q3.py). Make sure all the results are reproducible for grading.

```
>>>
  Enter first number: 4.5
  Enter second number: 3
  Enter third number: 3
  Enter last number: 3.5
  Equal
```

4. (10 pt.) Implement a program that requests the user to enter the x and y coordinates (each between −10 and 10) of a dart and computes whether the dart has hit the dartboard, a circle with center (0, 0) and radius 8. If so, string It is in! should be printed on the screen.

Include your solutions and explain your code in the report. Also, implement the solution in Python using the supplied file (Q4.py). Make sure all the results are reproducible for grading.

```
>>>
Enter x: 2.5
Enter y: 4
It is in!
```

5. (15 pt.) Implement function pay() that takes as input two arguments: an hourly wage and the number of hours an employee worked in the last week. Your function should compute and return the employee's pay. Any hours worked beyond 40 is overtime and should be paid at 1.5 times the regular hourly wage.

Include your solutions and explain your code in the report. Also, implement the solution in Python using the supplied file (Q5.py). Make sure all the results are reproducible for grading.

```
>>> pay(10, 35)
350
>>> pay(10, 45)
475.0
```

6. (10 pt.) The probability of getting n heads in a row when tossing a fair coin n times is 2^{-n}. Implement function prob() that takes a nonnegative integer n as input and returns the probability of n heads in a row when tossing a fair coin n times.

Include your solutions and explain your code in the report. Also, implement the solution in Python using the supplied file (Q6.py). Make sure all the results are reproducible for grading.

```
>>> prob(1)
0.5
>>> prob(2)
0.25
```

7. (20 pt.) Implement function points() that takes as input four numbers x_1 , y_1 , x_2 , y_2 that are the coordinates of two points (x_1, y_1) and (x_2, y_2) in the plane. Your function should compute:

- The slope of the line going through the points, unless the line is vertical

- The distance between the two points

Your function should *print* the computed slope and distance in the following format. If the line is vertical, the value of the slope should be string 'infinity'. *Note:* Make sure you convert the slope and distance values to a string before printing them.

Include your solutions and explain your code in the report. Also, implement the solution in Python using the supplied file (Q7.py). Make sure all the results are reproducible for grading.

```
>>> points(0, 0, 1, 1)
The slope is 1.0 and the distance is 1.41421356237
>>> points(0, 0, 0, 1)
The slope is infinity and the distance is 1.0
```

8. (10 pt.) Implement function abbreviation() that takes a day of the week as input and returns its two-letter abbreviation.

Include your solutions and explain your code in the report. Also, implement the solution in Python using the supplied file (Q8.py). Make sure all the results are reproducible for grading.

```
>>> abbreviation('Tuesday') 'Tu'
```

9. (10 pt.) Write function lastF() that takes as input two strings of the form 'FirstName' and 'LastName', respectively, and returns a string of the form 'LastName, F.'. (Only the initial should be output for the first name.)

Include your solutions and explain your code in the report. Also, implement the solution in Python using the supplied file (Q9.py). Make sure all the results are reproducible for grading.

```
>>> lastF('Albert', 'Camus') 'Camus, A.'
```

10. (15 pt.) Write a function distance() that takes as input a number: the time elapsed (in seconds) between the flash and the sound of thunder. Your function should return the distance to the lightning strike in kilometers. The speed of sound is approximately 340.29 meters per second; there are 1000 meters in one kilometer.

Include your solutions and explain your code in the report. Also, implement the solution in Python using the supplied file (Q10.py). Make sure all the results are reproducible for grading.

```
    >>> distance(3)
    1.0208700000000002
```