15-112: Introduction to Programming and Computer Science, Fall 2019

Homework 2 Programming: Input, Output, Expressions, and Variables

Due: Tuesday, September 10, 2019 by 10:00pm

Welcome to 15-112! This programming homework is designed to get you more practice with reading values from the keyboard, using expressions to process these value, and showing results on the screen.

Your submission will be made through the web interface of Autolab. To do so, please create a file called yourandrewidhw2.py and write all the functions within that file You can submit this file at:

`https://autolab.andrew.cmu.edu/courses/15112q-f19`

This assignment has a total of 28 points. You are not allowed to use loops, conditional execution (if statements, or the str function) in this homework.

Digitization

1. (5 points) Write a function called setKthDigit, that takes three integers parameters: an integer n, and integer k and an integer d - where n is strictly positive, k is non-negative, and d is a digit between 0 and 9. Your function should return the number n but with the kth digit replaced with d. Counting starts at 1 and goes right-to-left, so the 1st digit is the rightmost digit. You can assume that k will never be less than 1. For example:

```
setKthDigit(468, 1, 1) returns 461
setKthDigit(468, 2, 1) returns 418
setKthDigit(468, 3, 1) returns 168
setKthDigit(468, 4, 1) returns 1468
setKthDigit(468, 5, 1) returns 10468
```

All About Triangles

2. In this problem, we will write functions that enable us to find the area of triangle formed by the intersection of three lines.

(a) (2 points) The first function you will write should be called $distance(x_1, y_1, x_2, y_2)$. This function takes x and y cordinates of two points and return the euclidean distance between these points. For example:

```
distance(0, 0, 0,0) returns 0.0
distance(0, 0, 3,4) returns 5.0
distance(1, 1, 5,8) returns 8.0622577483
distance(-5, 1, 6,-3) returns 11.704
```

(b) (3 points) The second function you will write is called *triangleAreaFromSides* that takes the three sides of the triangle as parameters and returns the area of the triangle. For Example:

```
triangleAreaFromSides(0, 0, 0) returns 0.0
triangleAreaFromSides(2, 2, 3) returns 1.9843134833
triangleAreaFromSides(3, 4, 5) returns 6.0
triangleAreaFromSides(8, 9, 15) returns 29.9332590942
```

(c) (3 points) The third function should use the first two functions to calculate the area of a triangle using the cordinates of the three corners. The function should be called *triangleAreaByCoordinates* and should take six parameters. For Example:

```
triangleAreaByCoordinates(0, 0, 3, 4, 5,5) returns 2.4999
triangleAreaByCoordinates(5, 6, 2, 3, -1, 2) returns 3.0
triangleAreaByCoordinates(-1,-2,-3,-2,0,0) returns 2.0
triangleAreaByCoordinates(-5, 1, 6,-3,2,80) returns 448.5
```

(d) (3 points) The next function is called lineIntersection. This function takes four parameters m1, b1, m2, and b2, where m1 and b1 represent one line and m2, b2 represent a second line according the following equations:

```
y = m1*x + b1
y = m2*x + b2
```

The function lineIntersection returns the x-value of the point at which these two lines intersect. You can assume that the lines are not parallel or identical - hence they will always intersect.

For Example:

```
lineIntersection(3, 3, 2.3, 4) returns 1.42857
```

(e) (5 points) The final function is called areaBetweenLines. This function takes six parameters representing the slopes and intercepts of three lines. The function areaBetweenLines(m1, b1, m2, b2, m3, b3), should return the area of triangle made by the intersection of these three lines. You must use the functions that you have already writtent to first find the point of intersection of each pair of lines.Then return the area of the triangle formed by connecting these three points of intersection. Again, you can assume that all three pairs of lines do intersect at distinct points and form a triangle.

For Example:

```
areaBetweenLines(2, -5, 0, 23, 4, 3) returns 161.99999999999983
```

Population Explosion

"When scientists talk about half-life, they are referring to how long it will take for half of a sample to decay. In the case of nuclear waste, it refers to how long it takes for half of the radioactive material to turn into lead.

Exponential growth is very similar, but deals with, you guessed it, growth, instead of decay. The most common example is the growth of bacteria colonies. Bacteria multiply at an alarming rate. If we assume that bacteria can double every hour and if we start with just a single bacteria, then after one day there will be over 16 million bacteria!

Obviously exponential growth, or decay for that matter, cannot continue indefinitely. Eventually there would no longer be any space or nutrients available for the bacteria, or the last atom of plutonium would decay into lead. As a result, exponential growth and decay only refers to the early stages of both processes.

The mathematics behind exponential growth and decay is rather simple. In fact, we use the same formula as for continuous compound interest. " Source (http://math.ucsd.edu/ wgarner/math4c/textbook/chapter4/expgrowthdecay.htm)

$N = N_0 e^{kt}$

where N is the final population, N_0 is the starting population, k is growth rate and t is time. For example, if current population of a species is 1000 (N_0), growth rate is 0.02 (k) and we want to know the population after 1 day (t=24), we can use the above formula to arrive at the answer (N).

$N = 1000 * e^{0.02*24} = 1616.07440219$

3. (3 points) Write a function called populationEstimation that takes the initial population, growth rate, and time as parameters and returns the population at the end of the time given. For Example:

```
populationEstimation(1000,0.02,24) return 1616
```

Pascal's Triangle

"In mathematics, Pascal's triangle is a triangular array of the binomial coefficients.The rows of Pascal's triangle are conventionally enumerated starting with row n = 0 at the top (the 0th row). The entries in each row are numbered from the left beginning with k = 0 and are usually staggered relative to the numbers in the adjacent rows. The triangle may be constructed in the following manner: In row 0 (the topmost row), there is a unique nonzero entry 1. Each entry of each subsequent row is constructed by adding the number above and to the left with the number above and to the right, treating blank

entries as 0. For example, the initial number in the first (or any other) row is 1 (the sum of 0 and 1), whereas the numbers 1 and 3 in the third row are added to produce the number 4 in the fourth row" [1]. The Pascal's Triangle with 8 rows is shown in the following figure:

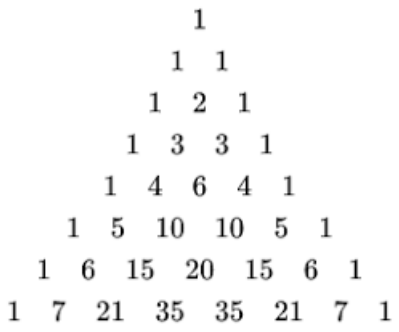

Figure 1: Pascal's Triangle with 8 rows (rows 0 - 7).

4. (4 points) Write a function called pascalsTriangle that takes a row (n) and entry (k) as inputs and returns the kth value in nth row of a Pascal's Triangle. You can assume that inputs will be always valid (non negative values) and k with be a valid entry for nth row. Both n and k start at 0. You cannot use loops or conditional execution in this question.

[1] *https://en.wikipedia.org/wiki/Pascal's_triangle*