

15-110: Principles of Computing

HOMEWORK 02

Due: 16th August, 2022 at 10:00pm

- You need to complete the Python file for this assignment, and submit it to Gradescope.
- There are 100 points.
- You must solve the tasks **individually**, always abiding by the course and CMU's academic integrity policy.

1. (15 points) **Circle Area**

The area of a circle of radius r is given by the following formula:

$$A = \pi \times r^2$$

Complete the function `circleArea(r)` that returns the area of a circle with radius r .

Since there is no way to represent a real π in a machine with finite memory, you can use the approximation 3.141592.

2. (25 points) **Hotdog Purchase**

Hot dogs are an American tradition. Each year, Americans eat up to 20 Billion hot dogs. A classic hot dog is made up of two components: A frank (the meat) and a bun. Yet, for reasons that mystify mankind, the franks are typically sold in packs of ten and the buns in packs of eight. And, of course, you must buy full packages.

Write the function `hotdogPurchase(numHotdogs)` that takes as input the total number of hot dogs you want to make, and returns the number of packages of franks and the number of packages of buns you need to purchase. You may assume that the argument, `numHotdogs`, is a non-negative integer number and the function returns as a pair of integers the smallest number of packages of franks and buns that must be purchased. Hint: you shall make use of the `math.ceil()` function.

Some example cases:

- `hotdogPurchase(50)` returns (5, 7), meaning 5 packs of franks and 7 packs of buns;
- `hotdogPurchase(10)` returns (1, 2), 1 pack of franks and 2 packs of buns;
- `hotdogPurchase(5)` returns (1, 1), 1 pack of both;
- `hotdogPurchase(0)` returns (0, 0), 0 packs of both.

3. (25 points) **Hotdog Excess**

Write the function `hotdogExcess(numHotdogs)` that takes the total number of hot dogs you want to make (as a non-negative integer) and returns the number of excess franks and buns you will need to purchase.

Hint: you may want to use some of the code from `hotDogPurchase`, which you just wrote!

Some example cases:

- `hotdogExcess(1)` returns (9, 7), meaning 9 excess franks and 7 excess buns;
- `hotdogExcess(20)` returns (0, 4), no excess in franks and 4 excess buns;
- `hotdogExcess(10)` returns (0, 6), no excess in franks and 6 excess buns;
- `hotdogExcess(25)` returns (5, 7), 5 excess franks and 7 excess buns;
- `hotdogExcess(32)` returns (8, 0), 8 excess franks and 0 excess buns.

4. (35 points) **K^{th} digit**

Write the function `getKthDigit(n, k)` that takes a positive integer n and a non-negative integer k , and returns the k^{th} digit of n , starting from 0, counting from right to left. So:

- `getKthDigit(789, 0)` returns 9
- `getKthDigit(789, 1)` returns 8
- `getKthDigit(789, 2)` returns 7
- `getKthDigit(789, 3)` returns 0

You may assume that the digits to the left of the last non-zero digit are all zeros.

Note that you will (only) need to use the arithmetic operators `//`, `%`, `**`. The operator `//` is the floor division (or integer division) operator, while `%` operator is used to get the remainder of a division operation between integers. The following properties will be useful to tackle the question:

- An integer number with d digits is represented in base 10, meaning that each digit is multiplied by a power of 10 depending on its position in the number, where the rightmost position corresponds to the 0th power. For instance, the number 789 is equivalent to: $9 \times 10^0 + 8 \times 10^1 + 7 \times 10^2$.
- When dividing an integer by 10 using `//`, the smallest digit is dropped so that the result consists in all remaining digits shifted to the right by one digit place. For example, $743 // 10 = 74$.
- A positive integer modulo 10 will return the digit corresponding to the rightmost position: $743 \% 10 = 3$.