15-110: Principles of Computing

Homework 04

Due: 30th 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) **countMultiplesOfSeven**

   Write the function `countMultiplesOfSeven(a, b)` that takes two integers, a and b, and returns the number of multiples of seven that occur between a and b (including a and b). For example, `countMultiplesOfSeven(4, 16)` would return 2, since two multiples of 7 (7 and 14) occur within that range.

2. (15 points) **containsOddDigits**

   Write the function `containsOddDigits(x, n)` that takes an n-digit integer, x, and returns `True` if it contains any odd digits, and `False` otherwise. You can assume that the number of digits of x is n. For example, `containsOddDigits(42, 2)` returns `False`, and `containsOddDigits(15110,5)` returns `True`.

3. (20 points) **Fibonacci**

   The Fibonacci series is a famous series of numbers defined as:

$$\begin{aligned} F_0 &= 1 \\ F_1 &= 1 \\ F_n &= F_{n-1} + F_{n-2} \end{aligned}$$

   So, the zeroth Fibonacci number is 1, the first is 1, the second is 2 (1+1), the third is 3 (1+2), and so on.

   Fibonacci is related to the *golden ratio*, and it also appears in several patterns in nature, such as in flowers and pinecones. If you want to know more about it, check out its Wikipedia page (`https://en.wikipedia.org/wiki/Fibonacci_number`).

   Implement the function `fibonacci(n)` that returns the n-th Fibonacci number. For example, `fibonacci(4)` should return 5.

4. (25 points) **Prime numbers**

   A *prime number* is a number that is divisible only by two different numbers: 1 (one) and by itself. For example, the number 7 is Prime, because it can be divided only by 1 and 7.

   Implement the function `isPrime(n)` that returns `True` if `n` is a prime number, or `False` otherwise.

5. (25 points) **Towels**

   You have a towel factory whose special feature is producing perfectly square towels. Unfortunately, reams of cloth do not come as a perfect square, and each time you need to figure out what should be the square size so that no fabric goes to waste. In principle, you could always cut 1x1 squares, but you want your squares to be as big as possible.

   Implement the function `towels(w, h)` that takes as input the width and height of the ream of cloth, and returns the minimum number of towels you can cut.

   For example, if the ream of cloth is 27×18 meters, you can cut six 9×9 towels. So `towels(27, 18)` should return 6.

   Hint: Think about the problem first before implementing. If you want the minimum number of towels, they should be as big as possible. How can you figure out the maximum size of square towels to cut?