15-110: Principles of Computing

Homework 04

Due: 3$^{\text{rd}}$ October, 2020 at 23:59

- You must solve the tasks **individually**.
- There are 50 points.

1. (10 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 zero-th Fibonacci number is 1, the first is 1, the second is 2 (1+1), the third is 3 (1+2), and so on and so forth.

   Fibonacci is related to the *golden ration*, and it also appears in several patterns in nature, such as in flowers and pinecones. If you would like 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.

2. (10 points) **Change coins**

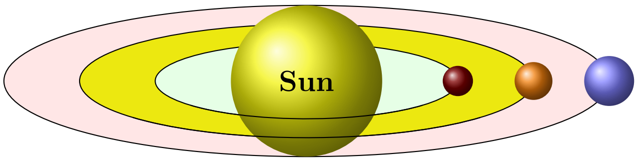
   You have an unlimited amount of coins with the values 25, 10, 5, and 1. You need to give n in change to someone, but you want to use the smallest number of coins possible.

   Implement the function `changeCoins(n)` that returns the minimum number of coins you will use to give n in change. For example, `changeCoins(142)` should return 9 (5 coins of 25, 1 coin of 10, 1 coin of 5, and 2 coins of 1).

3. (15 points) **Orbits**

   Planets are typically orbiting a big star such as the sun. But depending on each planet's mass and distance, it may take more or less time to complete one orbit.

   Suppose you know that three planets A, B, and C were aligned at some point in time, like the following figure:



   Given the time it takes for A, B, and C to complete one orbit, it is possible to calculate when they will be alligned again.

   Implement the function `orbits(a, b, c)` that takes as input the orbits or three aligned planets, and returns then first time when they will align again. For example, `orbits(12, 4, 5)` should return 60.

4. (15 points) **Last hit**

Đorđe and Igli are playing a computer game where they both need to defeat a monster who is keeping a treasure. According to the rules of the game, both players get points for defeating the monster, but only the player who delivers the final blow gets to keep the treasure.

At the beginning, both Đorđe and Igli attack, dealing d_D and d_I damage points to the monster. After each attack, both Đorđe and Igli need to wait exactly t_D and t_I seconds, respectively, to attack again. Whenever Đorđe and Igli attack together (such as in the beginning), Dorde has priority and attacks first. The monster is defeated when its life pointsbecome zero or less.

Implement the function `lastHit(dD, tD, dI, tI, m)` that takes as input:

- `dD`: Đorđe's damage
- `tD`: Đorđes recovery time
- `dI`: Igli's damage
- `tI`: Igli's recovery time
- `m`: monster's total life

and returns `"Dorde"` if Đorđe is the one giving the last blow, or `"Igli"` if Igli is giving the last blow.

For example, `lastHit(5, 3, 5, 3, 10)` should return `"Igli"`, since:

- Second 0: Đorđe attacks causing 5 points of damage. Now the monster has 5 life points.
- Second 0: Igli attacks causing 5 points of damage. The monster has 0 life points and Igli dealt the last blow.

Another example, `lastHit(5, 3, 5, 3, 11)` should return `"Dorde"`.

Yet another example: `lastHit(5, 3, 10, 7, 213)` should return `"Dorde"`.