1. **Create a function that takes three collections of arguments and returns the sum of the product of numbers.**

```
product(1,2)(1,1)(2,3)  ➞ 8
// 1 * 1 * 2 + 2 * 1 * 3

product(10,2)(5,0)(2,3)  ➞ 100
// 10 * 5 * 2 + 2 * 0 * 3

product(1,2)(2,3)(3,4)  ➞ 30
// 1 * 2 * 3 + 2 * 3 * 4

product(1,2)(0,3)(3,0)  ➞ 0
// 1 * 0 * 3 + 2 * 3 * 0
```

2. Create a function that calls an object property with procedural like style.

```
magic.replace("azerty", "a", "A")  ➞ "Azerty"

magic.length("hello word")  ➞ 10

magic.trim("  javascript is awesome  ")  ➞ "javascript is awesome"

magic.slice([1, 2, 3, 4, 5], 2, 4)  ➞ [ 3, 4 ]
```

3. Car Constructor
    - Write a Car constructor that initializes `model` and `milesPerGallon` from arguments.
    - All instances built with Car:
        1. should initialize with a `tank` at 0
        2. should initialize with an `odometer` at 0
    - Give cars the ability to get fueled with a `.fill(gallons)` method. Add the gallons to the `tank`. - STRETCH: Give cars the ability to `.drive(distance)`. The distance drove:
    - Should cause the `odometer` to go up.
    - Should cause the `tank` to go down taking `milesPerGallon` into account.
    - STRETCH: A car which runs out of `fuel` while driving can't drive any more distance:
    - The `drive` method should return a string "I ran out of fuel at x miles!" x being `odometer`.

4. Write a function that makes the **first number as large as possible** by swapping out its digits for digits in the second number.

```
 maxPossible(9328, 456) ➞ 9658
// 9658 is the largest possible number built from swaps from 456.
// 3 replaced with 6 and 2 replaced with 5.

maxPossible(523, 76) ➞ 763

maxPossible(9132, 5564) ➞ 9655

maxPossible(8732, 91255) ➞ 9755
```

5. Write a function that will work equivalent new keyword.