## 3.City Taxes

*This task is an extension of Problem 1, you may use your solution from that task as a base.*

You will receive a city’s **name** (string), **population** (number), and **treasury** (number)as arguments, which you will need to set as **properties** of an **object** and **return** it. In addition to the input parameters, the object must have a property taxRate with an initial value of **10**, and three **methods** for managing the city:

* collectTaxes() **-** Increase **treasury** by population \* taxRate
* applyGrowth(percentage) **-** Increase population by **given percentage**
* applyRecession(percentage) **-** Decrease treasury by **given percentage**

Round down the values after each calculation.

### Input

Your solution will receive three **valid** parameters. The methods that expect parameters will be tested with valid input.

### Output

Return an **object** as described above. The methods of the object modify the object and don’t return anything.

|  |  |
| --- | --- |
| **Input** | **Output** |
| **const city =**  **cityTaxes('Tortuga',**  **7000,**  **15000);**  **console.log(city);** | {  name: 'Tortuga',  population: 7000,  treasury: 15000,  taxRate: 10,  collectTaxes: [Function: collectTaxes],  applyGrowth: [Function: applyGrowth],  applyRecession: [Function: applyRecession]  } |
| **Testing with code** | |
| **Input** | **Output** |
| **const city =**  **cityTaxes('Tortuga',**  **7000,**  **15000);**  **city.collectTaxes();**  **console.log(city.treasury);**  **city.applyGrowth(5);**  **console.log(city.population);** | 85000  7350 |