1. Create a function that takes a string and returns a string in which each character is repeated once.

   Examples
   doubleChar("String") → "SSttrriinngg"

   doubleChar("Hello World!") → "HHeellllloo WWoorrlldd!!"

   doubleChar("1234!_ ") → "11223344!!__ "

2. Write a JavaScript program to get the following output.

   Input: "!trams era uoY"
   Output: “You are smart!”

3. Create a function that always returns true for every item in a given array. However, if an element is the word "bridgeon", switch to always returning the opposite boolean value.

   Examples
   bridgeonSwitch(["bridge", "bridgeon", "on"]) → [true, false, false]
   bridgeonSwitch(["bridgeon", 10101, 3.14, 53, "bridgeon"]) → [false, false, false, false, true]
   bridgeonSwitch([false, false, "bridgeon", true]) → [true, true, false, false]

4. Print the following Reverse Triangle pattern.

   54321
   5432
   543
   54
   5

5. Create a function that takes two numbers as arguments (num, length) and returns an array of multiples of num until the array length reaches length.

   Examples
   arrayOfMultiples(7, 5) → [7, 14, 21, 28, 35]

   arrayOfMultiples(12, 10) → [12, 24, 36, 48, 60, 72, 84, 96, 108, 120]

   arrayOfMultiples(17, 6) → [17, 34, 51, 68, 85, 102]

6. Create a function that takes an array of arrays with numbers. Return a new (single) array with the largest numbers of each.

   Examples
   findLargest([[4, 2, 7, 1], [20, 70, 40, 90], [1, 2, 0]]) ➞ [7, 90, 2]

   findLargest([[-34, -54, -74], [-32, -2, -65], [-54, 7, -43]]) ➞ [-34, -2, 7]

   findLargest([[0.4321, 0.7634, 0.652], [1.324, 9.32, 2.5423, 6.4314], [9, 3, 6, 3]]) ➞ [0.7634, 9.32, 9]

7. Print the following Reverse Triangle pattern.

   54321
   4321
   321
   21
   1

8. Write a program to print the following Hollow Triangle Star Pattern.

   *
   * *
   * *
   * *
   * * * * *

9. A function that takes an array of objects which includes name and age of family members. Find the difference in age between the oldest and youngest family members, and return the youngest member's age, the oldest member's age and the age difference respectively as an array.

   Examples
   ageMembers([
   { name: "John", age: 15 },
   { name: "Steve", age: 50 }
   ]) ➞ [15, 50, 35]

   ageMembers([
   { name: "Mark", age: 38 },
   { name: "Charlie", age: 10 },
   { name: "Anna", age: 25 }
   ]) ➞ [10, 38, 28]

ageMembers([
{ name: "Rachel", age: 20 },
{ name: "Jennifer", age: 45 },
{ name: "Jack", age: 43 },
{ name: "Maria", age: 3 }
]) ➞ [3, 45, 42]

10. Create a function that takes an array of objects (groceries) which calculates the total price and returns it as a number. A grocery object has a product, a quantity and a price.

    Examples
    totalPrice([
    { product: "Milk", quantity: 1, price: 10 }
    ]) ➞ 10

    totalPrice([
    { product: "Milk", quantity: 3, price: 10 },
    { product: "Biscuit", quantity: 1, price: 20 }
    ]) ➞ 50

    totalPrice([
    { product: "Milk", quantity: 1, price: 10 },
    { product: "Eggs", quantity: 12, price: 5 },
    { product: "Bread", quantity: 2, price: 20 },
    { product: "Cheese", quantity: 1, price: 30 }
    ]) ➞ 140