Recursive Function

# Practice

* Write a recursive function which takes two positive integer arguments n,m and returns n % m. Don’t use %, \*, / operators.
* example:
  + mod(3, 2) –> 1

*Main method:*

public static void main(String[] args) {  
 System.out.println(mod(3, 2));  
}

*soloution:*

public static int mod(int a, int b) {  
 if(a<b)  
 return a;  
 else  
 return mod(a-b, b);  
}

* Write a recursive function which takes a string parameter and checks if all its characters appear only once.
* example:
  + isUnique(“pickle”) –> true
  + isUnique(“moon”) –> false
  + isUnique(“trash”) –> true

*Main method:*

public static void main(String[] args) {  
 System.out.println(isUnique("pickle"));  
 System.out.println(isUnique("moon"));  
 System.out.println(isUnique("trash"));  
}

*soloution:*

public static boolean isUnique(String s) {  
 if(s.length() == 1)  
 return true;  
 else {  
 for(int i=1; i<s.length(); i++)  
 if(s.charAt(0) == s.charAt(i))  
 return false;  
 return isUnique(s.substring(1));  
 }  
}

# Project

* Write a recursive function to reverse a string.

*main method:*

public static void main(String[] args) {  
 System.out.println(reverse("hello"));  
}

*soloution:*

public static String reverse(String s) {  
 if(s.length() < 2)  
 return s;  
 else  
 return s.charAt(s.length()-1) + reverse(s.substring(0, s.length()-1));  
}

* Write a recursive function to find the maximum value in an array of integers.

*main method:*

public static void main(String[] args) {  
 int[] arr = {1, 2, 3, 4, 5};  
 System.out.println(max(arr));  
}

*soloution:*

public static int max(int[] arr) {  
 if(arr.length == 1)  
 return arr[0];  
 else {  
 int[] newArr = new int[arr.length-1];  
 for(int i=1; i<arr.length; i++)  
 newArr[i-1] = arr[i];  
 return Math.max(arr[0], max(newArr));  
 }  
}

* Write a recursive function to find the greatest common divisor (GCD) of two numbers.

*main method:*

public static void main(String[] args) {  
 System.out.println(gcd(12, 18));  
}

*soloution:*

public static int gcd(int a, int b) {  
 if(a == b)  
 return a;  
 else if(a > b)  
 return gcd(a-b, b);  
 else  
 return gcd(a, b-a);  
}

* Write a recursive function to merge two sorted arrays into a single sorted array.

*main method:*

public static void main(String[] args) {  
 int[] arr1 = {1, 3, 5, 7, 9};  
 int[] arr2 = {2, 4, 6, 8, 10};  
 int[] arr = merge(arr1, arr2);  
 for(int i=0; i<arr.length; i++)  
 System.out.print(arr[i] + " ");  
}

*soloution:*

public static int[] merge(int[] arr1, int[] arr2) {  
 int[] arr = new int[arr1.length + arr2.length];  
 int i=0, j=0, k=0;  
 while(i<arr1.length && j<arr2.length) {  
 if(arr1[i] < arr2[j])  
 arr[k++] = arr1[i++];  
 else  
 arr[k++] = arr2[j++];  
 }  
 while(i<arr1.length)  
 arr[k++] = arr1[i++];  
 while(j<arr2.length)  
 arr[k++] = arr2[j++];  
 return arr;  
}