

NOMBRE		ETAPA / CICLO	CURSO
		CFGS DAW/DAM	1º
APELLIDOS		ASIGNATURA/MÓDULO	CONVOCATORIA
		PROGRAMACION	CONTINUA
DNI	FECHA	NOTA	
	22/11/2023		

1. **(3p)** Choose two of the following exercises and solve it recursively (0,75p each one), that is not using loops, and iteratively (0,75p each one) by using loops.

- a. Given a string and a non-empty substring sub, compute recursively the number of times that sub appears in the string, without the sub strings overlapping.

strCount("catcowcat", "cat") → 2

strCount("catcowcat", "cow") → 1

strCount("catcowcat", "dog") → 0

```
public static int strCount(String str, String sub) {
}
```

- b. Given a string, compute recursively the number of "55" substrings in the string. The "55" substrings should not overlap.

count55("55abc55") → 2

count55("abc55x55x55") → 3

count55("555") → 1

```
public static int count55(String str) {
}
```

- c. Given an integer, compute to obtain the binary number.

binary(10) → "1010"

binary(7) → "111"

binary(2) → "10"

```
public static String binary(int n) {
}
```

2. (3p) Create a java Class having the following functions:

- a. A main method and declare and initialize a bi-dimensional array of integers. The matrix is having 4 rows and 5 columns. This main method calls some of the following methods (you decide which ones).
- b. A method called fillMatrix which fills the matrix with random numbers from zero (0) to ninety-nine (99).
- c. A method called showMatrix returning a String with the values of each matrix element. Remember that “\n” prints a new line whilst “\t” prints a tabulation. If in the main method we show by console the result of calling this method, result will be something like (if number has 1 digit, will be printed with a leading zero):

```
85    33    84    71    13
55    88    76    38    77
63    01    48    70    90
91    08    44    46    71
```

- d. A method called primeNumbers returning an array with all the prime number elements from the matrix. The array will contain exactly the number of prime numbers without repetitions. For example, will return this for previous matrix:

```
[13, 71]
```

- e. To ease the previous method, create a method called removeDuplicates which receives an array of integers and will return an array without duplicates and fitting the length of the array. From this example taken from part c:

```
[71, 13, 71, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

The method will return this:

```
[71, 13, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

- f. To ease the previous method (part e) you can create a method called copyOfArray, having as parameters an array of integers and a length, to copy the specified array with the specified length into the new array which will be returned. Example:

```
From → [71, 13, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

```
Call → copyOfArray(array, 2);
```

```
Result → [13, 71]
```

- g. A method called isPrime returning whether a number is a prime number or not. Remember that 0 and 1 are not prime numbers.

3. (2p) Create the next functions:

- Create a function that returns only one character between 'a' and 'z'. You must ensure that the user only enters a lowercase character, displaying an error message if the user enters an invalid entry. You can solve it recursively and using supplementary functions, but it is not necessary.
- Create a function that displays a sequence of characters from the character passed by parameter to 'z' as shown in the example below.
- Create a main function and combine the previous functions.

EXAMPLE:

Give me a char between 'a' and 'z':

aa

Only 1 character! Try again.

Give me a char between 'a' and 'z':

A

Only the characters between 'a' and 'z' are allowed. Try again.

Give me a char between 'a' and 'z':

c

C d E f G h I j K l M n O p Q r S t U v W x Y z

4. **(2p)** Given a bidimensional array of String to store people and grades, show the following lists of people with the following requirements:
- List of people sorted by name.
 - List of people sorted by grades. Show in descending order.

Example of bidimensional array.

```
String[][] people = {  
    {"Juan", "5.75"},  
    {"Carlos", "8.70"},  
    {"Teresa", "7.25"},  
    {"Ana", "3.2"},  
    {"Antonio", "8"},  
    {"Francisco", "4.1"},  
    {"Luis", "6.75"},  
    {"Lorena", "9.9"},  
};
```

You can use the Selection sorting algorithm.

```
public static void sort(int[] array){  
    int menor;  
    for(int i=0;i<array.length-1;i++){  
        menor = i;  
        for(int j=i;j< array.length ; j++){  
            if(array[j]<array[menor])  
                menor=j;  
        }  
        if(i!=menor)  
            swap(array,i,menor);  
    }  
}
```