

NOMBRE		ETAPA / CICLO	CURSO
		CFGS DAW/DAM	1º
APELLIDOS		ASIGNATURA/MÓDULO	CONVOCATORIA
		PROGRAMACION	RECUPERACIÓN
DNI	FECHA	NOTA	
	31/01/2023		

1. (1,50p → 0,75 each) Recursive functions

- a. Given a string, compute recursively (no loops) the number of "33" substrings in the string. The "33" substrings should not overlap. Examples:

count33("33abc33") → 2

count33("abc33x33x33") → 3

count33("333") → 1

Hint: public int count33(String str) { }

- b. Given an array of ints, compute recursively the number of times that the value 73 appears in the array. We'll use the convention of considering only the part of the array that begins at the given index. In this way, a recursive call can pass index+1 to move down the array. The initial call will pass in index as 0.

array73([1, 2, 73], 0) → 1

array73([73, 73], 0) → 2

array73([1, 2, 3, 4], 0) → 0

Hint: public int array73(int[] nums, int index) { }

2. (1p → 0,50 each) Iterative functions

- a. Create a function to solve exercise 1a in iterative mode using a while loop.
- b. Create a function to solve exercise 1b in iterative mode using a do-while loop.

3. (1,5p → 0,50 each) Given an array of Strings, containing numbers from 00 to 49, perform the following functions.

Matrix example:

```
09 08 05 04
11 45 35 20
32 33 07 00
```

- a. Function returning the mirror matrix. Expected result:

```
04 05 08 09
20 35 45 11
00 07 33 32
```

- b. Function returning the transposed matrix. Expected result:

```
09 11 32
08 45 33
05 35 07
04 20 00
```

- c. Function returning the sum of the matrix with its mirror array. Expected result:

```
13 13 13 13
31 80 80 31
32 40 40 32
```

4. (1p → 0,50 each) Identity matrix. Identity matrix means a matrix having row and column the same dimension, and having a 1 where number of row and column are equals, otherwise 0. Example of identity matrix 4:

```
1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1
```

- a. (0.5 p.) Create the ident method capable of generating the identity matrix:
`public static int[][] ident(int size)`
- b. (0.5 p.) Create a method to display the identity matrix and create a main method to display the identity matrix of size 6;

```
1 0 0 0 0 0
0 1 0 0 0 0
0 0 1 0 0 0
0 0 0 1 0 0
0 0 0 0 1 0
0 0 0 0 0 1
```

5. (4p → 1 each) Given a `String[][]` with names and height:

```
String[][] person = {{ "Carlos", "159.9"}, { "Luis", "180.6"}, { "Teresa", "175.2"}, ...};
```

- a. Sort the array by names using a bubble algorithm (including the swap function).
b. Given a name, return its height or an error message if it isn't in the array. Solve it iteratively.
c. Find the tallest person in the array.
d. Find the smallest person in the array.

6. (1p) Transform the if statement in the next code into a switch statement

```
public class E6 {  
  
    public static void main(String[] args) {  
  
        int i=(int) (Math.random()*10);  
  
        if(i>0 && i<4) {  
            if (i == 3 || i == 2)  
                System.out.println("A");  
            System.out.println("B");  
        }else if(i==5 || i<7){  
            System.out.println("C");  
        }else  
            System.out.println("D");  
  
    }  
}
```

Bubble algorithm

```
public class Bubble {  
    public static void sort(int[] v) {  
        for (int j = 0; j < v.length - 1; j++)  
            for (int i = 0; i < v.length - 1; i++) {  
                if (v[i] > v[i + 1])  
                    Tool.swap(v, i, i + 1);  
            }  
    }  
}
```