

Apellido y Nombre:

DNI:

Cant. Hojas:

1. Un dispositivo que toma imágenes de un fenómeno natural está corriendo sobre una plataforma tecnológica con recursos limitados. Cada imagen (representada por una matriz de NxM) está compuesta por píxeles con valores entre 0 y 255. Se desea implementar un algoritmo de compresión que comprima aquellas porciones de la imagen distintas del color negro (0 en la escala de valores del pixel). **Se debe implementar la solución para un arreglo** (luego dicha solución se reutilizará en un futuro para cada fila de la matriz). Para ello, se pide realizar un programa en JAVA que, dado un arreglo de tamaño M, para cada secuencia delimitada por uno o mas pixeles de color negro (valor 0) con más de X repeticiones de un valor de píxel (todos los elementos de la secuencia deben ser iguales), comprima la secuencia poniendo en la primera posición el valor negado de la cantidad de ocurrencias y a continuación el valor del píxel que se repite. El arreglo empieza y termina con un separador 0 (color negro). Implementar usando las buenas prácticas de programación estructurada vistas en la cátedra.

Ejemplo:

Arreglo que tiene una fila de la imagen:

0	67	67	67	67	67	67	67	67	0	14	0	33	33	33	33	0	5	98	0
---	----	----	----	----	----	----	----	----	---	----	---	----	----	----	----	---	---	----	---

El arreglo quedaría de la siguiente forma, para un X=3:

0	-8	67	0	14	0	-4	33	0	5	98	0	0	0	0	0	0	0	0	0
---	----	----	---	----	---	----	----	---	---	----	---	---	---	---	---	---	---	---	---

2. Para el siguiente fragmento de programa en JAVA, indique el estado final de las variables cantidadMayor e inicioMayor:

```
final int MAX=10;
final char PATRON='a';
final char[] letras={'a','a','4','a','a',';','a','a','a','a'};
int cantidad = 0, cantidadMayor = 0, inicioMayor=0;
for (int i=0; i < MAX; i++) {
    if (letras[i] == PATRON) {
        cantidad++;
    } else {
        if (cantidad > cantidadMayor) {
            cantidadMayor = cantidad;
            inicioMayor = i - cantidadMayor;
        }
        cantidad = 0;
    }
}
```

cantidadMayor:	
inicioMayor:	

//cantidadMayor=2, inicioMayor=0

POSIBLE RESOLUCIÓN

```
public class parcial_100623_TEMA1 {
    final static int MAX=20,SEPARADOR=0,X=3;
    public static void main (String[] args){
        // PUNTO 1
        int[] arreglolmagen={0,67,67,67,67,67,67,67,67,0,14,0,33,33,33,33,0,5,98,0};
        mostrarArreglo(arreglolmagen);
        procesar(arreglolmagen);
        mostrarArreglo(arreglolmagen);
    }
    public static void procesar(int[] arreglolmagen) {
        int inicio=0,fin=-1;
        while (inicio<MAX){
            inicio=buscarIni(arreglolmagen, fin+1);
            if (inicio<MAX){
                fin=buscarFin(arreglolmagen,inicio);
                if (fin-inicio+1>X && sonTodosIguales(arreglolmagen,inicio,fin)){
                    comprimir(arreglolmagen, inicio,fin);
                    fin=inicio+2;
                }
            }
        }
    }
    public static void comprimir(int[] arreglolmagen, int inicio, int fin) {
        int tamano=fin-inicio+1;
        while (inicio<fin-1){
            correrAlzquierda(arreglolmagen,inicio);
            fin--;
        }
        arreglolmagen[inicio]=-1*(tamano);
    }
    public static void correrAlzquierda(int[] arr, int pos) {
        for (int i = pos; i < MAX - 1; i++)
            arr[i] = arr[i + 1];
    }
    public static boolean sonTodosIguales(int[] arreglolmagen, int inicio, int fin) {
        while (inicio<fin && arreglolmagen[inicio]==arreglolmagen[inicio+1]){
            inicio++;
        }
        return (inicio==fin);
    }
    public static int buscarIni(int[] arrEnteros, int ini) {
        while (ini<MAX && arrEnteros[ini]==SEPARADOR)
            ini++;
        return ini;
    }
    public static int buscarFin(int[] arrEnteros, int ini) {
        while (ini<MAX && arrEnteros[ini]!=SEPARADOR)
            ini++;
        return ini-1;
    }
    public static void mostrarArreglo(int[] arr){
        for (int i=0;i<MAX;i++)
```

```
        System.out.print(arr[i] + "|");  
    System.out.println("");  
}  
}
```