

## PROBLEMA 1

```
public static void main(String[] args) {
    String archivo = "data/Deudas-1.txt";
    try (BufferedReader br = new BufferedReader(new FileReader(archivo))) {
        String linea;
        while ((linea = br.readLine()) != null) {
            // Separar la línea por espacios
            datos = linea.split(" ");
            // Asignar datos a variables
            escribirArchivo();
        }
    } catch (IOException e) {
    }
}

public static void escribirArchivo() {
    try (BufferedWriter bw = new BufferedWriter(new FileWriter("data/" + "DeudaActualizada.txt", append:true))) {
        bw.write(datos[0] + " " + datos[1] + " Liberado " + datos[5] + "\n");
        bw.close();
    } catch (IOException e) {
        System.err.println("Error al escribir en el archivo: " + e.getMessage());
    }
}
```

LA3891 Loja Liberado USD490	LA3891 Loja Central de Riesgo USD490
TM9883 Loja Liberado USD456	TM9883 Loja Central de Riesgo USD456
MP0128 Quito Liberado USD200	MP0128 Quito Central de Riesgo USD200
MG1191 Zamora Liberado USD100	MG1191 Zamora Central de Riesgo USD100
AM4583 Cuenca Liberado USD303	AM4583 Cuenca Central de Riesgo USD303
PN6728 Quito Liberado USD123	PN6728 Quito Central de Riesgo USD123
LA3891 Loja Liberado USD490	HF3876 Cuenca Central de Riesgo USD239
TM9883 Loja Liberado USD456	MR5915 Zamora Central de Riesgo USD309
MP0128 Quito Liberado USD200	VM8928 Cuenca Central de Riesgo USD345
MG1191 Zamora Liberado USD100	CC0934 Zamora Central de Riesgo USD101
AM4583 Cuenca Liberado USD303	DS4900 Loja Central de Riesgo USD134
PN6728 Quito Liberado USD123	

## PROBLEMA 2

```
public static void escribirArchivo() {
    try (BufferedWriter bw = new BufferedWriter(new FileWriter("data/" + "Pacientes.txt", append:true))) {
        bw.write("Cédula de identidad: " + cedula + ";" +
            "Apellidos completos: " + apellidos + ";" +
            "Nombres completos: " + nombres + ";" +
            "Fecha de nacimiento: " + fecha + ";" +
            "Estatura: " + estatura + ";" +
            "Peso: " + peso + ";" +
            "Alergias: " + alergias + "\n");
        bw.close();
    } catch (IOException e) {
        System.err.println("Error al escribir en el archivo: " + e.getMessage());
    }
}

public static void mostrarPacientes() {
    try (BufferedReader br = new BufferedReader(new FileReader("data/Pacientes.txt"))) {
        String linea;
        System.out.println("\nListado de pacientes:");
        int c = 1;

        while ((linea = br.readLine()) != null) {
            datos = linea.split(";");
            System.out.println("Paciente " + c);
            for (int i = 0; i < datos.length; i++) {
                System.out.println(datos[i]);
            }
            c++;
            System.out.println("\n");
        }
    }
}
```

```
peuro
Ingrese la fecha de nacimiento del paciente:
5 de Abril 2009
Ingrese la estatura del paciente:
1,80
Ingrese el peso del paciente:
77
Ingrese las alergias del paciente;
Ibuprofeno
```

```
Listado de pacientes:
Paciente 1
Cédula de identidad: 9182091820912
Apellidos completos: camacho
Nombres completos: juan
Fecha de nacimiento: si
Estatura: 897.0
Peso: 987.0
Alergias: jklh
```

```
Paciente 2
Cédula de identidad: 9878979
Apellidos completos: sanchez
Nombres completos: pepe
Fecha de nacimiento: 987
Estatura: 98798.0
Peso: 989.0
Alergias: lkjals
```

```
Paciente 3
Cédula de identidad: 1900374928
Apellidos completos: Gonzalez
Nombres completos: Pedro
Fecha de nacimiento: 5 de Abril 2009
Estatura: 1.8
Peso: 77.0
Alergias: Ibuprofeno
```

**PROBLEMA 3**

```
package ej3;
// Done
public class Ej3 {

    public static void main(String[] args) {
        int[] arrayA = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
        long[] arrayB = new long[10];

        System.out.println("Arreglo A:");
        for (int i = 0; i < arrayA.length; i++) {
            System.out.print(arrayA[i] + " ");
        }

        System.out.println("\nArreglo B (Factorial de los numeros del arreglo A):");
        for (int i = 0; i < arrayB.length; i++) {
            arrayB[i] = factorial(arrayA[i]);
        }
        for (int i = 0; i < arrayB.length; i++) {
            System.out.print(arrayB[i] + " ");
        }
        System.out.println();

        //Funcion que calcula el factorial con recursividad
        public static long factorial(int x) {
            long fact;
            if (x == 0) {
                fact = 1;
            } else {
                fact = x * factorial(x - 1);
            }
            return fact;
        }
    }
}
```

Arreglo A:

1 2 3 4 5 6 7 8 9 10

Arreglo B (Factorial de los numeros del arreglo A):

1 2 6 24 120 720 5040 40320 362880 3628800

## PROBLEMA 4

```
package ej4;
// Done
import java.util.Scanner;

public class Ej4 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Ingrese la cantidad de numeros a mostrar: ");
        int num = sc.nextInt();
        int[] arrayFib = new int[num];
        System.out.println("Serie fibonacci en un arreglo:");
        for (int i = 0; i < arrayFib.length; i++) {
            arrayFib[i] = fibonacci(i);
            System.out.print(arrayFib[i] + " ");
        }
    }
    public static int fibonacci(int x) {
        if (x == 0) {
            return 0;
        } else if (x == 1) {
            return 1;
        } else {
            return fibonacci(x - 1) + fibonacci(x - 2);
        }
    }
}

Ingrese la cantidad de numeros a mostrar:
10
Serie fibonacci en un arreglo:
0 1 1 2 3 5 8 13 21 34 BUILD SUCCESSFUL (total time: 1 second)
```

## PROBLEMA 5

Ingrese la cantidad de numeros primos a mostrar: 10

Arreglo de numeros primos:

2 3 5 7 11 13 17 19 23 29 BUILD SUCCESSFUL (total time: 2 seconds)

```
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int n;

    System.out.print("Ingrese la cantidad de numeros primos a mostrar: ");
    n = sc.nextInt();

    int[] arrayPr = new int[n];
    int j = 0;

    for (int i = 1; j < n; i++) {
        if (primo(i)) {
            arrayPr[j] = i;
            j++;
        }
    }

    System.out.println("Arreglo de numeros primos:");
    for (int i = 0; i < arrayPr.length; i++) {
        System.out.print(arrayPr[i] + " ");
    }
}
```

```
public static boolean primo(int num) {
    boolean esPrimo = false;
    int c;
    if (num >= 2) {
        c = num - 1;
        while (num % c != 0) {
            c = c - 1;
        }
        if (c == 1) {
            esPrimo = true;
        }
    }
    return esPrimo;
}
```

## PROBLEMA 6

```
package ej6;
// Done
import java.util.Scanner;

public class Ej6 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Ingrese el tamaño del arreglo a ingresar: ");
        int num = sc.nextInt();
        int[] array = new int[num];
        boolean orden = true;
        for (int i = 0; i < array.length; i++) {
            System.out.print("Ingrese el elemento #" + (i+1) + ": ");
            array[i] = sc.nextInt();
        }
        for (int i = 0; i < array.length-1; i++) {
            if (array[i] > array[i+1]) {
                orden = false;
                break;
            }
        }
        System.out.println("\nArreglo ingresado por teclado:");
        for (int i = 0; i < array.length; i++) {
            System.out.print(array[i] + " ");
        }

        if (orden) {
            System.out.println("\nEl arreglo esta ordenado");
        } else{
            System.out.println("\nEl arreglo no esta ordenado");
        }
    }
}
```

```
Ingrese el tamaño del arreglo a ingresar: 5
Ingrese el elemento #1: 1
Ingrese el elemento #2: 3
Ingrese el elemento #3: 5
Ingrese el elemento #4: 0
Ingrese el elemento #5: 2

Arreglo ingresado por teclado:
1 3 5 0 2
El arreglo no esta ordenado
```

## PROBLEMA 7

```
package ej7;
import java.util.Scanner;
// Done

public class Ej7 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n, sumaPares, sumaImpares;
        int[] lista;
        System.out.print("Cuantos # va a ingresar?: ");
        n = sc.nextInt();
        lista = new int[n];
        sumaPares = 0;
        sumaImpares = 0;

        for (int i = 0; i < lista.length; i++){
            System.out.print("Ingrese el #" + (i + 1) + " numero: ");
            lista[i] = sc.nextInt();
            if (lista[i] % 2 == 0){
                sumaPares = sumaPares + lista[i];
            } else{
                sumaImpares = sumaImpares + lista[i];
            }
        }

        System.out.println("\nLista ingresada:");
        for (int i = 0; i < lista.length; i++) {
            System.out.print(lista[i] + " ");
        }

        System.out.println("\nEn total, la suma de los numeros pares de la lista"
            + " es igual a " + sumaPares + " y la suma de los numeros "
            + "impares de la lista es igual a " + sumaImpares);
    }
}

Cuantos # va a ingresar?: 6
Ingrese el #1 numero: 1
Ingrese el #2 numero: 4
Ingrese el #3 numero: 10
Ingrese el #4 numero: 3
Ingrese el #5 numero: 5
Ingrese el #6 numero: 7

Lista ingresada:
1 4 10 3 5 7
En total, la suma de los numeros pares de la lista es igual a 14 y la suma de los numeros impares de la lista es igual a 16
```

## PROBLEMA 8

Impresion de notas por pantalla en formato:

Estudiante N: Nota1 / Nota2 / Nota3 / Nota4

Estudiante #1: 7 / 8 / 1 / 7 /

Promedio de cada estudiante:

Estudiante 1: 5.75

Estudiante 2: 3.75

Estudiante 3: 4.25

Promedio de cada asignatura:

Asignatura 1: 4.55

Asignatura 2: 4.45

Asignatura 3: 5.05

Asignatura 4: 4.25

Promedio general: 4.57

```

public static void main(String[] args) {
    // Constantes
    int nEstudiantes = 20;
    int nAsignaturas = 4;

    // Variables
    int[][] notas = new int[nEstudiantes][nAsignaturas];
    float[] promedioEstudiantes = new float[nEstudiantes];
    float[] promedioAsignaturas = new float[nAsignaturas];
    float promedioGeneral = 0;

    // Generación de datos aleatorios
    Random random = new Random();
    for (int i = 0; i < nEstudiantes; i++) {
        for (int j = 0; j < nAsignaturas; j++) {
            notas[i][j] = random.nextInt(11);
        }
    }

    // Cálculo del promedio de cada estudiante
    for (int i = 0; i < nEstudiantes; i++) {
        float promedioEstudiante = 0;
        for (int j = 0; j < nAsignaturas; j++) {
            promedioEstudiante += notas[i][j];
        }
        promedioEstudiantes[i] = promedioEstudiante / nAsignaturas;
    }

    // Cálculo del promedio de cada asignatura
    for (int j = 0; j < nAsignaturas; j++) {
        float promedioAsignatura = 0;
        for (int i = 0; i < nEstudiantes; i++) {
            promedioAsignatura += notas[i][j];
        }
        promedioAsignaturas[j] = promedioAsignatura / nEstudiantes;
    }

    // Cálculo del promedio general
    for (int i = 0; i < nEstudiantes; i++) {
        promedioGeneral += promedioEstudiantes[i];
    }
    promedioGeneral = promedioGeneral / nEstudiantes;

    // Impresión de matriz por pantalla
    System.out.println("Impresión de notas por pantalla en formato:");
    System.out.println("Estudiante N: Nota1 / Nota2 / Nota3 / Nota4");
    for (int i = 0; i < nEstudiantes; i++) {
        System.out.print("\nEstudiante #" + (i + 1) + ": ");
        for (int j = 0; j < nAsignaturas; j++) {
            System.out.print(notas[i][j] + " / ");
        }
        System.out.println();
    }

    // Impresión de resultados
    System.out.println("\nPromedio de cada estudiante:");
    for (int i = 0; i < nEstudiantes; i++) {
        System.out.printf("Estudiante %d: %.2f\n", i + 1, promedioEstudiantes[i]);
    }

    System.out.println("\nPromedio de cada asignatura:");
    for (int j = 0; j < nAsignaturas; j++) {
        System.out.printf("Asignatura %d: %.2f\n", j + 1, promedioAsignaturas[j]);
    }

    System.out.printf("\nPromedio general: %.2f\n", promedioGeneral);
}

```

## PROBLEMA 9

```

        System.out.println("Ingrese el nombre del cliente " + (i + 1) + ": ");
        clientes[i] = sc.nextLine();
    }

    boolean[][] asistencia = new boolean[numClientes][6];

    for (int i = 0; i < numClientes; i++) {
        System.out.println("Registrando asistencia para " + clientes[i] + ":");
        for (int j = 0; j < 6; j++) {
            String dia = obtenerDia(indice: j);
            System.out.println("¿Asistió el " + dia + "? (s/n)");
            String respuesta = sc.nextLine();
            asistencia[i][j] = respuesta.equalsIgnoreCase("s");
        }
    }

    // Calcular y mostrar clientes con descuento
    System.out.println("\nClientes con descuento del 20%:");
    for (int i = 0; i < numClientes; i++) {
        int diasAsistidos = 0;
        for (int j = 0; j < 6; j++) {
            if (asistencia[i][j]) {
                diasAsistidos++;
            }
        }
        if (diasAsistidos >= 3) {
            System.out.println(clientes[i]);
        }
    }
}

```



```
run:
Ingrese la cantidad de clientes:
1
Ingrese el nombre del cliente 1:
jOSE
Registrando asistencia para jOSE:
¿Asistió el lunes? (s/n)
s
¿Asistió el martes? (s/n)
n
¿Asistió el miércoles? (s/n)
n
¿Asistió el jueves? (s/n)
s
¿Asistió el viernes? (s/n)
s
¿Asistió el sábado? (s/n)
n
Clientes con descuento del 20%:
jOSE
```

## PRBLEMA 10

```
System.out.println(×:"Ingrese el número de fila:");
int fila = sc.nextInt() - 1;
System.out.println(×:"Ingrese el número de asiento:");
int asiento = sc.nextInt() - 1;

if (opcion == 1) {
    if (asientos[fila][asiento] == 'V') {
        asientos[fila][asiento] = 'O';
        System.out.println(×:"Asiento ocupado.");
    } else {
        System.out.println(×:"El asiento ya está ocupado.");
    }
} else if (opcion == 2) {
    if (asientos[fila][asiento] == 'O') {
        asientos[fila][asiento] = 'V';
        System.out.println(×:"Asiento liberado.");
    } else {
        System.out.println(×:"El asiento ya está vacío.");
    }
}

static void mostrarAsientos(char[][] asientos) {
```

```
Seleccione una opción:
1. Ocupar asiento
2. Liberar asiento
3. Salir
1
Ingrese el número de fila:
2
Ingrese el número de asiento:
2
Asiento ocupado.
Estado actual de los asientos:
V V V
V O V
V V V
Seleccione una opción:
1. Ocupar asiento
2. Liberar asiento
3. Salir
```

## PROBLEMA 11

```
for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
        matriz[i][j] = rand.nextInt( bound: 99) + 1;
    }
}

System.out.println( x:"Matriz original:");
mostrarMatriz(matriz);

for (int i = 0; i < n; i++) {
    for (int j = 0; j < n - i; j++) {
        int temp = matriz[i][j];
        matriz[i][j] = matriz[n - 1 - j][n - 1 - i];
        matriz[n - 1 - j][n - 1 - i] = temp;
    }
}

System.out.println( x:"Matriz transpuesta (sobre la diagonal secundaria:");
mostrarMatriz(matriz);
```

```
Ingrese el numero de filas y columnas de la matriz:
5
Matriz original:
39 61 47 1 64
66 71 71 89 38
35 21 94 35 50
99 7 83 14 56
42 93 67 62 98
Matriz transpuesta (sobre la diagonal secundaria):
98 56 50 38 64
62 14 35 89 1
67 83 94 71 47
93 7 21 71 61
42 99 35 66 39
```

## PROBLEMA 12

```
Scanner sc = new Scanner( source:System.in);
System.out.println( x:"Ingrese el numero de cursos: ");
int numCursos = sc.nextInt();
System.out.println( x:"Ingresa el numero de alumnos: ");
int numAlumnos = sc.nextInt();
int[][] notas = new int[numAlumnos][numCursos];

for (int i = 0; i < numAlumnos; i++) {
    System.out.println("Ingrese las notas del alumno " + (i + 1) + ":");
    for (int j = 0; j < numCursos; j++) {
        System.out.println("Nota del curso " + (j + 1) + ":");
        notas[i][j] = sc.nextInt();
    }
}

System.out.println( x:"Matriz de notas:");
mostrarMatriz( matriz:notas);
```

```
Ingrese las notas del alumno 2:
Nota del curso 1:
8
Nota del curso 2:
6
Ingrese las notas del alumno 3:
Nota del curso 1:
10
Nota del curso 2:
10
Matriz de notas:
10 9
8 6
10 10
```

## PROBLEMA 13

```
int indiceEstudiante = -1;
int indiceCurso = -1;

for (int i = 0; i < estudiantes.length; i++) {
    if (estudiantes[i].equalsIgnoreCase(anotherString:estudiante)) {
        indiceEstudiante = i;
        break;
    }
}

for (int i = 0; i < cursos.length; i++) {
    if (cursos[i].equalsIgnoreCase(anotherString:curso)) {
        indiceCurso = i;
        break;
    }
}

if (indiceEstudiante != -1 && indiceCurso != -1) {
    System.out.println(estudiante + " tiene " + notas[indiceEstudiante][indiceCurso] + " en " + curso);
} else {
    System.out.println("Estudiante o curso no encontrado.");
}
```

Ingrese el nombre del curso 2:

POO

Ingrese las notas de Daniel:

Nota en Estructura de Datos:

10

Nota en POO:

10

Ingrese el nombre del estudiante para buscar su nota:

Daniel

Ingrese el nombre del curso:

Estructura de Datos

Daniel tiene 10 en Estructura de Datos

## PROBLEMA 14

Ingrese la accion que desea realizar:

1. Ingresar nuevo cliente - 2. Atender Cliente - 3. Retirarse de la cola (Digite 4 para salir): 1

Ingrese la cedula del cliente a procesar: 1104

Ingreso correcto de cliente!

1101-1102-1103-1104-0-

Ingrese la accion que desea realizar:

1. Ingresar nuevo cliente - 2. Atender Cliente - 3. Retirarse de la cola (Digite 4 para salir): 2

Cliente Atendido!

1102-1103-1104-0-0-

Ingrese la accion que desea realizar:

1. Ingresar nuevo cliente - 2. Atender Cliente - 3. Retirarse de la cola (Digite 4 para salir): 3

Cliente retirado de la cola

1102-1103-0-0-0-

Ingrese la accion que desea realizar:

1. Ingresar nuevo cliente - 2. Atender Cliente - 3. Retirarse de la cola (Digite 4 para salir): 4

Saliendo del programa, Adios!

```

public static int[] arrayCed;
public static int topeC = 0;
public static boolean vacio = true;

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Ingrese la cantidad de clientes a procesar: ");
    int n = sc.nextInt();
    byte opc;
    arrayCed = new int[n];
    do {
        System.out.println("\nIngrese la accion que desea realizar: ");
        System.out.print("1. Ingresar nuevo cliente - 2. Atender Cliente - "
            + "3. Retirarse de la cola (Digite 4 para salir): ");
        opc = sc.nextByte();
        switch (opc) {
            case 1 -> {
                System.out.print("\nIngrese la cedula del cliente a procesar: ");
                int cedula = sc.nextInt();
                pushCola(cedula);
                mostrarCola();
            }
            case 2 -> {
                popCola();
                mostrarCola();
            }
            case 3 -> {
                popCliente();
                mostrarCola();
            }
            case 4 ->
                System.out.println("Saliendo del programa, Adios!");
            default ->
                System.out.println("Opcion incorrecta o fuera de rango");
        }
    } while (opc != 4);
}

```

```

public static void mostrarCola() {
    for (int i = 0; i < arrayCed.length; i++) {
        System.out.print(arrayCed[i] + "-");
    }
    System.out.println();
}

public static void pushCola(int cedula) {
    if (vacio || topeC < arrayCed.length) {
        arrayCed[topeC] = cedula;
        topeC++;
        System.out.println("Ingreso correcto de cliente!");
    } else {
        System.out.println("No se puede agregar otro cliente a la cola");
    }
    if (topeC == arrayCed.length - 1) {
        vacio = false;
    }
}

public static void popCliente() {
    if (vacio != true || topeC > 0) {
        for (int i = topeC - 1; i < arrayCed.length - 1; i++) {
            arrayCed[i] = arrayCed[i + 1];
        }
        arrayCed[arrayCed.length - 1] = 0;
        topeC--;
        System.out.println("Cliente retirado de la cola");
    } else {
        System.out.println("No hay clientes por retirar");
    }
    if (topeC == 0) {
        vacio = true;
    }
}

```

```

public static void popCola() {
    if (vacio != true || topeC > 0) {
        for (int i = 0; i < arrayCed.length - 1; i++) {
            arrayCed[i] = arrayCed[i + 1];
        }
        arrayCed[arrayCed.length - 1] = 0;
        topeC--;
        System.out.println("Cliente Atendido!");
    } else {
        System.out.println("No hay clientes por atender");
    }
    if (topeC == 0) {
        vacio = true;
    }
}

```