



MODALIDAD PRESENCIAL

FACULTAD DE INGENIERIAS Y ARQUITECTURAS

CARRERA DE COMPUTACION

ACTIVIDAD

TEMA: Construcción de programas usando programación estructurada y lectura de archivos

Nombre: Kelvin Sneider Sarango Chalán

Docente: Encalada Encalada Ángel Eduardo.

Paralelo: B.

PERIODO ACADEMICO

OCTUBRE 2022 – FEBRERO 2023

AÑO

2023

Taller 12

Respuestas

Materia: Introducción a la programación.	Fecha: 19 – 01 – 2023.
URL de la carpeta “Taller03” de su repositorio GitHub	
https://github.com/KelvinSrng/IP2022_Sarango_Kelvin.git	

A continuación, copie el código fuente Java de los programas desarrollados.

Problema 1: Repeticiones en vector

```
package ocurrenciasenvector1;

import java.util.Scanner;

public class OcurrenciasEnVector1 {

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

        //Declaracion de variables
        int num;

        //Ingreso de datos
        System.out.println("\tREPETICIONES EN VECTOR");

        System.out.println("=====");
        System.out.print("Ingrese el tamaño del vector:");
        num = sc.nextInt();
        int[] elemento = new int[num];

        for ( int i = 0; i < num ; i ++ ) {
            System.out.print("Ingrese el valor del elemento " + (i + 1) + ": ");
            elemento[i] = sc.nextInt();
        }
        elemento = OrdenarVector(elemento); //Invocar la funcion para ordenar el vector
        mostrarOcurrencias(elemento);
    }
    //Funcion para ordenar un vector
```

```

public static int[] OrdenarVector(int x[]) {
    int aux;
    for (int lim = x.length-1; lim >= 1; lim--) {
        for (int i = 0; i <= lim - 1; i++) {
            if (x[i] > x[i+1]) {
                aux = x[i];
                x[i] = x[i+1];
                x[i+1] = aux;
            }
        }
    }
    return x;
}

//Metodo para mostrar las ocurrencias del vector
public static void mostrarOcurrencias(int[] x) {
    int ocurrencias = 0, num = 0;
    System.out.printf("%nValor          Ocurrencias%n");
    System.out.printf("-----          -----%n");
    for (int i = 0; i < x.length; i++) {
        if (ocurrencias == 0) {
            num = x[i];
            ocurrencias++;
        }
        if (i != x.length - 1) {
            if (num != x[i + 1]) {
                System.out.printf("%5d%20d%n", num,
ocurrencias);
                ocurrencias = 0;
            } else {
                ocurrencias++;
            }
        } else {
            System.out.printf("%5d%20d%n", num,
ocurrencias);
        }
    }
}
}

```

Problema 2: Ordenar una matriz

```

package ordenarmatriz1;

import java.util.Scanner;

```

```

public class OrdenarMatriz1 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        //Declaracion de variables
        int n, m, c = 0;

        //Ingreso de datos
        System.out.println("\t\tORDENAR MATRIZ");

        System.out.println("=====
        =====");
        System.out.print("Ingrese el numero de filas para
        la matriz: ");
        n = sc.nextInt();
        System.out.print("Ingrese el numero de columnas
        para la matriz: ");
        m = sc.nextInt();
        int[][] elementos = new int[n][m]; int[] vector =
        new int[n * m];

        //Llenado de la matriz
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < m; j++) {
                System.out.print("Ingrese el valor del
                elemento [ " + (i + 1) + ", " + (j + 1) + " ]: ");
                elementos[i][j] = sc.nextInt();
                vector[c] = elementos[i][j];
                c ++;
            }
        }
        vector = OrdenarVector(vector); //Invocar la
        funcion para ordenar el vector
        c = 0;
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < m; j++) {
                elementos[i][j] = vector[c];
                c ++;
            }
        }
        System.out.println("\n\t\tMATRIZ ORDENADA ");
        MostrarMatriz(elementos);
    }
    //Mostrar matriz
    public static void MostrarMatriz(int [][] matriz) {
        for (int i = 0; i < matriz.length; i++) {
            for (int j = 0; j < matriz[0].length; j++) {
                System.out.print("+----");
            }
            System.out.print("+\n");
            for (int j = 0; j < matriz[0].length; j++) {

```

```

        System.out.printf("|%3d  ",matriz[i][j]);
    }
    System.out.print("|\\n");
}
for (int j = 0; j < matriz[0].length; j++) {
    System.out.print("+-----");
}
System.out.print("+\\n");

}
//Funcion para ordenar un vector
public static int[] OrdenarVector(int x[]) {
    int aux;
    for (int lim = x.length-1; lim >= 1; lim--) {
        for (int i = 0; i <= lim - 1; i++) {
            if (x[i] > x[i+1]) {
                aux = x[i];
                x[i] = x[i+1];
                x[i+1] = aux;
            }
        }
    }
    return x;
}
}

```

Problema 3: Convertir decimal a binario (Solución recursiva)

```

package decimalabinario3;

import java.util.Scanner;

public class DecimalABinario3 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        //Declaracion de variables
        int num;

        //Ingreso de datos
        System.out.println("\\tCONVERTIR          DECIMAL          A
BINARIO");

        System.out.println("=====
=====");
        System.out.print("Ingresa un numero entero: ");
    }
}

```

```

        num = sc.nextInt();
        System.out.print("\nEl numero " + num + "
convertido a binario es : " );
        decimalAbinario(num);

    }
    public static void decimalAbinario(int n) {
        if (n < 2) {
            System.out.print("1");
        } else {
            decimalAbinario(n / 2);
            System.out.print(n % 2);
        }
    }
}

```

Problema 4: Máximo Común Divisor – Método Euclides (Solución iterativa)

```

package maximocomundivisor1;

import java.util.Scanner;

public class MaximoComunDivisor1 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        //Declaracion de variables
        int num1, num2;

        //Ingreso de datos
        System.out.println("MAXIMO COMUN DIVISOR DE DOS
NUMEROS ENTEROS");
        System.out.println("METODO EUCLIDES");

        System.out.println("=====
=====");
        System.out.print("Ingrese el primer numero:");
        num1 = sc.nextInt();
        System.out.print("Ingrese el segundo numero:");
        num2 = sc.nextInt();
        System.out.println("\nEl MCD es " + MCD(num1,
num2));
    }

    public static int MCD(int a, int b) {
        int aux;
        if ( a < b ) {

```

```

        aux = a;
        a = b;
        b = aux;
    }
    while ( a % b != 0) {
        aux = a % b;
        a = b;
        b = aux;
    }
    return b;
}
}

```

Problema 5: Máximo Común Divisor – Método Euclides (Solución recursiva)

```

package maximocomundivisor2;

import java.util.Scanner;

public class MaximoComunDivisor2 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        //Declaracion de variables
        int num1, num2, aux;

        //Ingreso de datos
        System.out.println("MAXIMO COMUN DIVISOR DE DOS
NUMEROS ENTEROS");
        System.out.println("METODO EUCLIDES");

        System.out.println("=====
=====");
        System.out.print("Ingresa el primer numero:");
        num1 = sc.nextInt();
        System.out.print("Ingresa el segundo numero:");
        num2 = sc.nextInt();
        if ( num1 < num2 ) {
            aux = num1;
            num1 = num2;
            num2 = aux;
        }
        //Salida
        System.out.println("\nEl MCD es " + MCD(num1,
num2));
    }
}

```

```

public static int MCD(int a, int b) {
    if ( a % b == 0 ) {
        return b;
    } else {
        return MCD(b, a%b);
    }
}
}

```

Problema 6: Mostrar primos de vector

```

package muestraprimos1;

import java.util.Scanner;

public class MuestraPrimos1 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        //Declaracion de variables
        int num, m = 0;

        //Ingreso de datos
        System.out.println("\tMOSTRAR PRIMOS DE VECTOR");

        System.out.println("=====
        =====");
        System.out.print("¿Cuantos      numeros      desea
        ingresar?: ");
        num = sc.nextInt();
        int [] elementos = new int[num];

        for ( int i = 0; i < num; i ++) {
            System.out.print("Ingresa el valor "+ (i + 1) +
            ": ");
            elementos[i] = sc.nextInt();
        }
        ordenarVector(elementos);
        System.out.println("\nLos      numeros      primos
        encontrados en el vector son: ");
        for ( int i = 0; i < num; i ++) {
            if (primo (elementos[i]) && elementos[i] != m)
            {
                System.out.println(elementos[i]);
                m = elementos[i];
            }
        }
    }
}

```



```

    }
    //Funcion para ordenar un vector
    public static int[] ordenarVector(int[] x) {
        int aux, c = x.length - 1;
        while (c >= 1) {
            for (int i = 0; i < c; i++) {
                if (x[i] > x[i + 1]) {
                    aux = x[i];
                    x[i] = x[i + 1];
                    x[i + 1] = aux;
                }
            }
            c--;
        }
        return x;
    }
    //Funcion para primo
    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 7: Ordenar vector (con recursividad)

```

package ordenarnumeros3;

import java.util.Scanner;

public class OrdenarNumeros3 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        //Declaracion de variables
        int num;

        //Ingreso de datos
    }
}

```

```

        System.out.println("\t ORDENAR VECTOR");

System.out.println("=====
=====");
        System.out.print("¿Cuantos      numeros      desea
ingresar?: ");
        num = sc.nextInt();

        int[] vector = new int[num];

        for (int i = 0; i < num; i++) {
            System.out.print("Ingrese el valor " + (i + 1)
+ ": ");
            vector[i] = sc.nextInt();
        }
        //Salida
        System.out.println("\nEl vector ordenado es:");
        System.out.print("[");
        vector = ordenarVector(vector, num - 1);
        for (int i = 0; i < num; i++) {
            System.out.print(vector[i] + " ");
        }
        System.out.print("]");
    }
    //Funcion para ordenar vector con recursividad
    public static int[] ordenarVector(int[] x, int
posicion) {
        int mayor = 0, aux, index = 0;
        for (int i = 0; i <= posicion; i++) {
            if (i == 0 || x[i] > mayor) {
                mayor = x[i];
                index = i;
            }
        }
        aux = x[posicion];
        x[posicion] = mayor;
        x[index] = aux;
        if (posicion == 1) {
            return x;
        } else {
            return ordenarVector(x, posicion - 1);
        }
    }
}

```

Problema 8: Personas de más de 50 años

```
package personas4;

import java.util.Scanner;
import java.util.ArrayList;
import java.io.File;

public class Personas4 {

    public static void main(String[] args) {
        try {
            String[] datos;
            ArrayList <String> nombres = new ArrayList();
            ArrayList <Byte> edades = new ArrayList();
            File archivo = new File("DATA/Personas.txt");
            Scanner sc = new Scanner(archivo);
            while (sc.hasNextLine()) {
                datos = sc.nextLine().split(",");
                if (Byte.parseByte(datos[2]) > 50) {
                    nombres.add(datos[0]);
                    edades.add(Byte.parseByte(datos[2]));
                }
            }
            System.out.println("Listado personas de más de
50 años");
            System.out.println("Nombre                Edad");
            System.out.println("-----");
            for (byte i = 0; i < nombres.size(); i++) {
                System.out.printf("%-21s%d\n",
nombres.get(i), edades.get(i));
            }
        } catch (Exception e) {
            System.err.println(e);
        }
    }
}
```