



**MODALIDAD PRESENCIAL**

**FACULTAD DE INGENIERIAS Y ARQUITECTURAS**

**CARRERA DE COMPUTACION**

**ACTIVIDAD**

**TEMA:** Construcción de programas usando arreglos bidimensionales.

**Nombre:** Kelvin Sneider Sarango Chalán

**Docente:** Encalada Encalada Ángel Eduardo.

**Paralelo:** B.

**PERIODO ACADEMICO**

OCTUBRE 2022 – FEBRERO 2023

**AÑO**

2023

## Taller 10

### Respuestas

<b>Materia:</b> Introducción a la programación.	<b>Fecha:</b> 05 – 01 – 2023.
<b>URL de la carpeta “Taller03” de su repositorio GitHub</b>	
<a href="https://github.com/KelvinSrng/IP2022_Sarango_Kelvin.git">https://github.com/KelvinSrng/IP2022_Sarango_Kelvin.git</a>	

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

#### Problema 1: Número capicúa

```
package numerocapicual;

import java.util.Scanner;

public class NumeroCapicual {

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

        //Declaracion de variables
        int num, inverso = 0, aux, cf;

        //Ingreso de datos
        System.out.println("VERIFICAR SI EL NUMERO ES CAPICUA");
        System.out.println ("-----");
        System.out.print ("Ingrese un numero: ");
        num = sc.nextInt();

        aux = num;

        while (aux != 0) {
            cf = aux % 10;
            inverso = inverso * 10 + cf;
            aux = aux / 10;
        }
        if (num == inverso) {
            System.out.println("\nEl numero " + num + " SI es capicua."); //Salida
        }else {
            System.out.println("\nEl numero " + num + " NO es capicua."); //Salida
        }
    }
}
```

```

    }
}

```

## Problema 2: Pi (Método Wallis)

```

package seriepiwallis1;

import java.util.Scanner;

public class SeriePiWallis1 {

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

        //Declaracion de variables
        int num;
        //double total =1 ;
        double b = 1;

        //Ingreso de datos
        System.out.println("\tMETODO WALLIS PI");
        System.out.println ("-----
----");
        System.out.print ("Ingrese un numero: ");
        num = sc.nextInt();

        for (int i = 1;i <= num; i++) {
            b = b * (double)((double)(2*i) /
(double)((2*i) - 1)) * (double)((double)(2*i) /
(double)((2*i)+1)));
            //System.out.println(b);
            //total = total*b;
        }
        System.out.println ("\nEl resultado es " + b*2);
    }
}

```

## Problema 3: Pirámide Invertida

```

package piramideinvertida1;

import java.util.Scanner;

public class PiramideInvertida1 {

```

```

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

        System.out.print("Introduzca numero de filas que
desea mostrar: ");
        int numFilas = sc.nextInt();
        sc.close();
        System.out.println();
        for(int numBlancos = 0, numAsteriscos =
(numFilas*2)-1; numAsteriscos>0; numBlancos++,
numAsteriscos -= 2){

            for(int i=0; i < numBlancos; i++){
                System.out.print(" ");
            }

            for(int j=numAsteriscos; j > 0; j--){
                System.out.print("*");
            }
            System.out.println();
        }
    }
}

```

#### Problema 4: Promedio columnas en matriz

```

package promediocolmatriz1;

import java.util.Scanner;

public class PromedioColMatriz1 {

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

        //Declaracion de variables
        int numF, numC, sumU, vm;
        float promedio = 0;

        //Ingreso de datos
        System.out.println("\tPROMEDIO          COLUMNAS          EN
MATRIZ");
        System.out.println("-----
-----");
        System.out.print("Ingrese el numero de filas de la
matriz: ");
    }
}

```

```

        numF = sc.nextInt();
        System.out.print("Ingrese el numero de columnas de
la matriz: ");
        numC = sc.nextInt();

        int matriz[][] = new int[numF][numC];

        //Llenado de la matriz
        System.out.println ("\nIngrese el contenido de la
matriz");
        for (byte i =0; i <= numF - 1; i++) {
            for (byte j = 0; j <= numC - 1; j++) {
                System.out.print ("Elemento [" + (i+1) +
", " + (j+1) + "]: ");
                matriz[i][j] = sc.nextInt();
            }
        }

        sumU= 0;

        for (byte i = 0; i <= numC - 1; i++) {

            vm = matriz[0][i];

            for (byte j = 0; j <= numF - 1; j++) {

                if(matriz[j][i] < vm){

                    vm = matriz[j][i];

                }

                sumU = sumU + matriz[j][i];
                promedio  =  (float)((float)  sumU
(numF));
            }

            //System.out.println("la suma de la columna "
+ (i+1)+ " ES: " + sumU);
            System.out.println("\nColumna " + (i+1) + ":
promedio = "+ promedio + ", valor mas bajo = " + vm);
            sumU = 0;
        }
    }
}

```

### Problema 5: Ordenar números

```
package ordenarnumeros1;

import java.util.Scanner;

public class OrdenarNumeros1 {

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

        //Declaracion de Variables
        int tmn, aux;

        //Ingreso de datos
        System.out.println("\tORDENAR NUMEROS DE MENOR A MAYOR");
        System.out.println("-----");
        System.out.print("¿Cuantos      numeros      desea ingresar?: ");
        tmn = sc.nextInt();

        int [] lista_numeros = new int[tmn];

        //Llenado del vector
        for (int i = 0; i < tmn; i++) {
            System.out.print("Ingrese el numero "+ (i+1) + ": ");
            lista_numeros[i] = sc.nextInt();
        }

        //Ordenar numeros de menor a mayor
        for (int i = 0; i < tmn; i++) {
            for (int j = i + 1; j < tmn; j++) {
                if (lista_numeros[i] > lista_numeros[j]) {
                    aux = lista_numeros[i];
                    lista_numeros[i] = lista_numeros[j];
                    lista_numeros[j] = aux;
                }
            }
        }

        //Salida
        System.out.println("\nEl vector rodenado es: ");
        for (int i = 0; i < tmn; i++) {
            System.out.println(lista_numeros[i]);
        }
    }
}
```

### Problema 6: Listar números abundantes entre 0 y N (con funciones)

```
package listanumabundantes2;

import java.util.Scanner;

public class ListaNumAbundantes2 {

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

        //Declaracion de variables
        int n, nat;

        //Ingreso de datos
        System.out.println("\tLISTA DE NUMEROS ABUNDANTES
ENTRE 0 Y N");
        System.out.println("-----
-----");
        System.out.print("Ingrese hasta que numero quiere
mostrar: ");
        n = sc.nextInt();

        //Proceso
        System.out.println("\nLos números abundantes entre
0 y " + n + " son: ");
        for (nat = 0; nat <= n; nat++) {
            if (validarAbundante(nat)) {
                System.out.println(nat);
            }
        }

        //Validar si un numero Abundante
        public static boolean validarAbundante(int num) {
            //Validar si nat es abundante
            int nat = num;
            int suma_div = 0;
            for (int c = 1; c < nat; c++) {
                if (nat % c == 0) {
                    suma_div += c;
                }
            }
            if (suma_div > nat) {
                return true;
            } else {
                return false;
            }
        }
    }
}
```

```
}  
}
```

### Problema 7: Contar primos en vector

```
package cuentaprimos2;  
  
import java.util.Scanner;  
  
public class CuentaPrimos2 {  
  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
  
        //Declaracion de variables  
        byte tmn, contador;  
        int p;  
  
        System.out.println("\t\tCONTADOR DE PRIMOS");  
        System.out.println("-----  
-----");  
        System.out.print("Ingrese la cantidad de numeros a  
digitar: ");  
        tmn = sc.nextByte();  
  
        //Crear vector  
        int [] lista_numeros = new int[tmn];  
  
        //Llenado del vector  
        for (int i = 0; i <= tmn -1; i++) {  
            System.out.print("Ingrese el numero " + (i+1)  
+ ": ");  
            lista_numeros[i] = sc.nextInt();  
        }  
  
        contador = 0;  
        for (int i = 0; i < tmn; i++) {  
            p = lista_numeros[i];  
            if (primo(p)) { //Invocacion a la funcion  
                contador++;  
            }  
        }  
  
        //Salida  
        System.out.println("\nDe los elementos ingresados  
" + contador + " son primos");  
    }  
}
```



```

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

```

#### Problema 8: Invertir decimal a binario

```

package decimalabinario2;

import java.util.Scanner;

public class DecimalABinario2 {

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

        //Declaracion de variables
        int num;

        //Titulo del programa
        System.out.println("\tCONVERTIR      DECIMAL      A
BINARIO");
        System.out.println("-----
-----");

        // Pedimos el número entero positivo a convertir
        do {
            System.out.print("Ingrese un número entero
positivo: ");
            num = sc.nextInt();
            if (num < 0) {
                System.out.println();
                System.out.println(";ERROR!POR      FAVOR
INGRESE UN NUMERO POSITIVO: ");
                System.out.println();
            }
        } while (num < 0);
    }
}

```

```

    }
    } while(num < 0);
    sc.close();

    //Llamada a la funcion y Salida de Datos
    System.out.println("\nEl número " + num + "
representado en binario es " + convertidor(num));
}

public static String convertidor(int num) {
    //Declaracion de Variables
    int n = num;
    int resto;
    StringBuilder binario = new StringBuilder();
    //Proceso
    while (n > 0) {
        // Calculamos el resto de la división de n
entre 2
        resto = n % 2;
        // Agregamos el resto a la cadena binario
        binario.insert(0, resto);
        // Dividimos n entre 2
        n = n / 2;
    }
    return binario.toString();
}
}

```