

Nombre y Apellido: Axel Chino.

Ejercicio 1:

Crear un arreglo de N elementos enteros. Llenar el arreglo con N números enteros provistos por el usuario. Escribir una función que tome como parámetro el arreglo y devuelva la posición del elemento más chico del arreglo. Imprimir por pantalla el menor de los elementos del arreglo.

Respuesta:

```
import java.util.Scanner;
public class Main
{
    public static void main(String[] args) {
        Scanner tomarvalor = new Scanner(System.in);
        // Creación del vector;
        int vector[]=new int[4];
        int num1=0;
        int numero_menor=0;

        // Bucle para pedir datos y cargarlos en un vector.
        for(int i=0; i<vector.length; i++){
            System.out.println("Ingrese un numero");
            num1 = tomarvalor.nextInt();
            vector[i]=num1;
        }

        num1 = vector[0];
        // ingresamos el vector y el primer numero ingresado
        numero_menor = Buscarnumero_menor(vector,num1);

        System.out.println("El menor numero ingresado es: "+numero_menor);

    }

    public static int Buscarnumero_menor(int vector1[], int numero1){
        // Bucle para mostrar el contenido del vector.
        for(int i=1; i<vector1.length; i++){
            if(vector1[i]<numero1){
                numero1 = vector1[i];
            }
        }
        return numero1;
    }
}
```

## Ejercicio 2:

Crear un arreglo de N elementos enteros. Llenar el arreglo con N números enteros provistos por el usuario. Escribir una función que tome como parámetro el arreglo y devuelva la posición del elemento más grande del arreglo. Imprimir por pantalla el mayor de los elementos del arreglo.

Respuesta:

```
import java.util.Scanner;
public class Main
{
    public static void main(String[] args) {
        Scanner tomarvalor = new Scanner(System.in);
        // Creación del vector;
        int vector[]=new int[4];
        int num1=0;
        int numero_mayor=0;

        // Bucle para pedir datos y cargarlos en un vector.
        for(int i=0; i<vector.length; i++){
            System.out.println("Ingrese un numero");
            num1 = tomarvalor.nextInt();
            vector[i]=num1;
        }

        num1 = vector[0];
        numero_mayor = Buscarnumero_mayor(vector,num1);

        System.out.println("El mayor numero ingresado es: "+numero_mayor);

    }

    public static int Buscarnumero_mayor(int vector1[], int numero1){
        // Bucle para mostrar el contenido del vector.
        for(int i=1; i<vector1.length; i++){
            if(vector1[i]>numero1){
                numero1 = vector1[i];
            }
        }
        return numero1;
    }
}
```

### Ejercicio 3:

Crear un arreglo de N elementos enteros. Imprimir por pantalla el mensaje: "Todos positivos" si todos sus elementos son positivos. En caso contrario imprimir por pantalla el mensaje: "No todos son positivos". Nota: se debería utilizar una única sentencia if en la impresión del mensaje por pantalla.

```
import java.util.Scanner;
public class Main
{
    public static void main(String[] args) {
        Scanner tomarvalor = new Scanner(System.in);
        // Creación del vector;
        int vector_num[]=new int[4];
        int num1=0;
        String mensaje="";

        // bucle para pedir numeros y guardarlo en el vector_num
        for(int i=0; i<vector_num.length; i++){
            System.out.println("Ingrese un numero al arreglo");
            num1=tomarvalor.nextInt();
            vector_num[i]=num1;
        }

        mensaje = recorrer_vector(vector_num);

        System.out.println(mensaje);
    }

    public static String recorrer_vector(int vector[]){
        //bucle para recorrer el vector y determinar si todos los numros son positivos o
        negativos.
        for(int i=0; i<vector.length; i++){
            if(vector[i] < 0){
                // El return corta la ejecución(bucle) y retorna el valor asignado despues de ella.
                return "No todos los números son positivos. \n";
            }
        }

        return "Todos los números son positivos. \n";
    }
}
```

#### Ejercicio 4:

Crear un arreglo booleanos de N elementos. Calcular e imprimir por pantalla el producto de la cantidad de los elementos que en una posición par del arreglo son true por la cantidad de elementos que en posición impar son false.

```
public class Main
{
    public static void main(String[] args) {
        boolean vector1[]={true,true,false,true,false};
        int resul;
        int cant_true=0, cant_false=0;

        resul = calcular_producto(vector1,cant_true,cant_false);
        System.out.println("El producto entre la cantidad de true en las posiciones pares\ny la
cantidad de false en las posiciones impares es: "+resul);
    }

    // función para calcular el producto
    public static int calcular_producto(boolean vector_boolean[], int cantidad_true, int
cantidad_false){
        // contamos cuantos true hay en el vector, pero solo en las posiciones pares, gracias a
que saltamos de 2 en 2 con (i=i+2);
        for (int i=0; i<vector_boolean.length; i=i+2){
            if(vector_boolean[i]==true){
                cantidad_true=cantidad_true+1;
            }

        }

        // contamos cuantos false hay en el vector, pero solo en las posiciones impares,
gracias a que saltamos de 2 en 2 pero empesando desde 1;
        for (int i=1; i<vector_boolean.length; i=i+2){
            if(vector_boolean[i]==false){
                cantidad_false=cantidad_false+1;
            }

        }

        return (cantidad_true*cantidad_false);

    }

}
```

### Ejercicio 5:

Crear dos arreglos de enteros de N elementos cada uno. Generar e imprimir por pantalla un tercer arreglo que contenga los elementos de posiciones impares del primer arreglo y los de posiciones pares del segundo arreglo.

<https://www.online-java.com/DvRAVQbeUJ>

```
public class Main
{
    public static void main(String[] args) {
        int vector_N1[]={4,5,7,2};
        int vector_N2[]={6,3,8,1};
        int vector_N3[]= tercerVector_impares(vector_N1,vector_N2);
        for(int i=0; i< vector_N3.length; i++){
            System.out.println(vector_N3[i]);
        }
        System.out.println("Welcome to Online IDE!! Happy Coding :)");
    }

    // cramos la función para cargar el vector_N3 en las posiciones impares del primer vector
    // con un vector y
    // con otro for cargamos el vector_N3 con las posiciones pares del segundo vector.
    public static int[] tercerVector_impares(int vector1[], int vector2[]){
        // creamos un vector temporal
        int vectortem3[]= new int[4];

        // parece bien
        for(int i=1; i<vector1.length; i=i+2){
            vectortem3[i]=vector1[i];
        }
        for(int i=0; i<vector2.length; i=i+2){
            vectortem3[i]=vector2[i];
        }
        return vectortem3;
    }
}
```

### Ejercicio 6:

Crear dos arreglos de enteros de N elementos cada uno. Si el primer arreglo se referencia con la variable A y el segundo arreglo se referencia con la variable B. Generar e imprimir por pantalla un tercer arreglo en el cual sus elementos están dados por la siguiente fórmula:  $C(i) = A(i) * B(N - i)$ , si C es el nombre de la variable que referencia al tercer arreglo.

<https://www.online-java.com/kUPr2eladm>

```
public class Main
{
    public static void main(String[] args) {
        int numeros_A[]={4,5,7,2};
        int numeros_B[]={6,3,8,1};
        int numeros_C[]= tercerVector_C(numeros_A,numeros_B);
        for(int i=0; i<numeros_C.length; i++){
            System.out.println("Posición "+"("+i+""): "+numeros_C[i]);
        }

    }

    public static int[] tercerVector_C(int vector_A[], int vector_B[]){
        int vectortem_C[]={0,0,0,0};
        for(int i=0; i<vectortem_C.length; i++){
            vectortem_C[i]= (vector_A[i] * vector_B[(vector_B.length-1)-i]);
        }
        return vectortem_C;
    }
}
```

Ejercicio 7:

Crear un arreglo de enteros de N elementos. Generar e imprimir por pantalla un arreglo que contenga los elementos del primer arreglo en orden inverso.

link: <https://www.online-java.com/TZ30SdijDE>

```
public class Main
{
    public static void main(String[] args) {
        int vec_N[]={7,2,5,3};
        vec_N= función_inverso(vec_N);

        for(int i=0; i<vec_N.length; i++){
            System.out.println("Posición "+"("+i+") "+vec_N[i]);
        }

    }

    public static int[] función_inverso(int numeros_vec[]){
        int vec_temporal[]=new int[4];
        for(int i=0; i<numeros_vec.length; i++){
            vec_temporal[i]=numeros_vec[(numeros_vec.length-1)-i];
        }
        return vec_temporal;
    }
}
```

#### Ejercicio 8:

Crear un arreglo de enteros de N elementos. Generar otro arreglo que es el resultado del producto del arreglo por el escalar 3.14. El producto de un arreglo por un escalar se define como la multiplicación de cada una de sus posiciones por el escalar. Imprimir por pantalla los elementos del arreglo resultante.

link: <https://www.online-java.com/GZpUWFSxtA>

```
public class Main
{
    public static void main(String[] args) {
        int vec_num[]={4,7,9,4};
        double vec_resultado[]=función_producto(vec_num);

        for(int i=0; i<vec_resultado.length; i++){
            System.out.println("Posición "+"("+i+" "+vec_resultado[i]);
        }

    }

    public static double[] función_producto(int arreglo_num[]){
        double vec_temporal[]=new double[4];
        for(int i=0; i<arreglo_num.length; i++){
            vec_temporal[i]=(arreglo_num[i]*3.14);
        }
        return vec_temporal;
    }

}
```



### Ejercicio 9:

link: <https://www.online-java.com/TBmQp0852F>

```
public class Main
{
    public static void main(String[] args) {
        int vec_num[]={4,7,9,4};
        int vec_num2[]={7,5,5,8};
        int suma_vectores=función_escalar(vec_num,vec_num2);

        System.out.println("El producto escalar de dos arreglos es: "+suma_vectores);

    }

    public static int función_escalar(int vec_numeros[], int vec_numeros2[]){
        int vec_temporal[]=new int[4];
        int acumulador=0;
        for(int i=0; i<vec_numeros.length; i++){
            vec_temporal[i]=(vec_numeros[i]*vec_numeros2[i]);
            acumulador=(acumulador+vec_temporal[i]);
        }
        return acumulador;
    }
}
```

#### Ejercicio 10:

Crear un arreglo de enteros de N elementos. Calcular e imprimir por pantalla el resultado del producto escalar entre el arreglo y un arreglo que contenga los elementos del primer arreglo en orden inverso.

usar 2 funciones anteriores