

## Práctica 1

### Temario:

- Estructuras de Control
- Estructuras de Control anidadas
- Arreglos estáticos unidimensionales
- Arreglos estáticos bidimensionales (matrices)

### Ejercicios:

#### Ejercicio 1.

Escribe un programa que solicite al usuario ingresar un carácter y un número entero positivo. Luego imprime en pantalla un triángulo formado por el carácter ingresado, con altura igual al número que se ingresó.

Ejemplo:

```
Ingresa un carácter: #  
Ingresa un número entero positivo: 5  
#  
##  
###  
####  
#####
```

#### Ejercicio 2.

Escribe un programa que imprima la tabla de multiplicar del 1 al 10 utilizando estructuras de control anidadas.

#### Ejercicio 3.

Escribe un programa que imprima un cuadrado de asteriscos de tamaño `n x n`, donde `n` es un número ingresado por el usuario.

Ejemplo:

```
Ingresa el tamaño del cuadrado: 5  
* * * * *  
* * * * *  
* * * * *  
* * * * *
```

## PROGRAMACIÓN IMPERATIVA

## Trabajo Práctico nº 1

Escuela de Tecnología | Área Algoritmos y Lenguajes

\* \* \* \* \*

**Ejercicio 4.**

Cargar un arreglo con 20 números enteros ingresados por teclado de manera que el arreglo siempre se encuentre ordenado en forma ascendente. El arreglo se cargará por completo.

Hacer dos variantes, suponiendo que el usuario carga los datos de las siguientes formas:

- a) El usuario ingresa los números en orden ascendente (Inserción al final)
- b) El usuario ingresa los números sin un orden en particular (Inserción ordenada)

**Ejercicio 5.**

Implementar la función **Buscar elemento** que reciba un arreglo de números enteros, su dimensión lógica y un número por parámetro e indique si dicho número existe en el arreglo.

**Ejercicio 6.**

Escribir una definición para una función llamada **mas\_uno** que tiene un parámetro formal para un arreglo de enteros e incrementa en uno el valor de cada elemento del arreglo. Añadir otros parámetros formales que se necesiten.

**Ejercicio 7.**

Escribir una función llamada **cantidad\_ocurrencias** que reciba como parámetro un arreglo de tipo int, su dimensión lógica y un parámetro con un valor de tipo int también. La función debe retornar la cantidad de ocurrencias del valor en el arreglo.

**Ejercicio 8.**

Implementar la función **eliminar** que reciba un arreglo de números enteros, su dimensión lógica y un número por parámetro y elimine dicho número si existe en el arreglo.

**Ejercicio 9.**

Implementar una función **eliminar\_ocurrencias** que reciba un arreglo de números enteros, su dimensión lógica y un número por parámetro y elimine todas las ocurrencias de dicho número que existan en el arreglo.

## PROGRAMACIÓN IMPERATIVA

## Trabajo Práctico nº 1

Escuela de Tecnología | Área Algoritmos y Lenguajes

**Ejercicio 10.**

Escribir un programa que permita al usuario gestionar los pagos de la cuota de un club. Para ello, almacenar en un arreglo los números de DNI de los socios que pagaron. El club no tiene más de 200 socios.

Permitir al usuario buscar un DNI en el arreglo y, si el DNI está en el arreglo, imprimir "Cuota al día". Si no está, imprimir "Socio con deuda".

También se debe permitir al usuario eliminar un DNI del arreglo, en caso de haberlo ingresado erróneamente. Finalmente, imprimir todo el arreglo.

**Ejercicio 11.**

Escribir un programa que permita al usuario ingresar números enteros positivos (los números ingresados no deben almacenarse). Al finalizar, informar la cantidad de veces que apareció cada dígito (del 0 al 9) en todos los números.

Ejemplo: Si el usuario ingresa los números 1156, 23, 73364, 988, 1003, 5, se debe informar que el 0 apareció 2 veces, el 1 apareció 3 veces, el 2 apareció 1 vez, el 3 apareció 4 veces, etc.

**Ejercicio 12.**

Escribir las sentencias necesarias para:

- A. Declarar un arreglo de enteros de 7 filas y 10 columnas
- B. Asignar el valor 105 a la posición que se encuentra en la segunda fila, quinta columna.
- C. Asignar en todas las posiciones de la cuarta fila el valor 5.
- D. Imprimir todos los valores de la matriz, colocando cada fila en una nueva línea.
- E. Permutar las columnas 3 y 5.
- F. Sumar todos los elementos de las filas 2 y 6.
- G. Imprimir todos los valores de la tercera columna.
- H. Hallar en qué fila y columna se encuentra almacenado el mayor elemento.

**Ejercicio 13.**

Escribe un programa que sume dos matrices de 3x3 y muestre la matriz resultante.

**Ejercicio 14.**

Una matriz cuadrada  $A$  se dice que es simétrica si  $A(i,j) = A(j,i)$  para todo  $i, j$ . Escribir un programa que decida si una matriz dada es o no simétrica.

**Ejercicio 15.**

Escribe un programa que busque un número en una matriz dada y devuelva su posición. Si el número no se encuentra, debe indicarlo.

**Ejercicio 16.**

Escribir un programa para ser utilizado en una ventanilla de venta de entradas de un pequeño teatro con capacidad para 50 personas. Las filas están identificadas de la 0 a la 4 y cada fila tiene asientos numerados de 1 a 10. Se debe registrar, por cada asiento, si está libre o no. Al iniciar el programa, todos los asientos están libres.

Cuando alguien desea comprar una entrada, se deben informar qué localidades están libres. Entonces el cliente deberá elegir su asiento (indicando fila y número de asiento) y éste se marcará como vendido. Si ya se vendieron todas las entradas, no se debe permitir elegir asiento.