

# Módulo Imperativo

## Práctica

### Semana 1: Ordenación

1. El administrador de un edificio de oficinas cuenta, en papel, con la información del pago de las expensas de dichas oficinas.

Implementar un programa modularizado que:

- a. Genere un vector, sin orden, con a lo sumo las 300 oficinas que administra. De cada oficina se ingresa el código de identificación, DNI del propietario y valor de la expensa. La lectura finaliza cuando se ingresa el código de identificación -1, el cual no se procesa.
- b. Ordene el vector, aplicando el método de inserción, por código de identificación de la oficina.

2. Un cine posee la lista de películas que proyectará durante el mes de octubre. De cada película se conoce: código de película, código de género (1: acción, 2: aventura, 3: drama, 4: suspenso, 5: comedia, 6: bélica, 7: documental y 8: terror) y puntaje promedio otorgado por las críticas.

Implementar un programa modularizado que:

- a. Lea los datos de películas y los almacene **por orden de llegada** y agrupados por código de género, en una estructura de datos adecuada. La lectura finaliza cuando se lee el código de película -1.
- b. Una vez almacenada la información, genere un vector que guarde, para cada género, el código de película con mayor puntaje obtenido entre todas las críticas.
- c. Ordene los elementos del vector generado en b) por puntaje.
- d. Luego de ordenar el vector, muestre el código de película con mayor puntaje y el código de película con menor puntaje.

3. Una librería requiere el procesamiento de la información de sus productos. De cada producto se conoce el código del producto, código de rubro (del 1 al 6) y precio.

Implementar un programa modularizado que:

- a. Lea los datos de los productos y los **almacene ordenados por código de producto** y agrupados por rubro, en una estructura de datos adecuada. El ingreso de los productos finaliza cuando se lee el precio -1  
Una vez almacenados, muestre los códigos de los productos pertenecientes a cada rubro:
- b. Genere un vector (de a lo sumo 30 elementos) con los productos del rubro 3. Considerar que puede haber más o menos de 30 productos del rubro 3. Si la cantidad de productos del rubro 3 es mayor a 30, almacenar los primeros 30 que están en la lista e ignore el resto.
- c. Ordene, por precio, los elementos del vector generado en b).
- d. Muestre los precios del vector ordenado.

## Semana 2: Recursión - Árboles

4. Escribir un programa que:

- a. Implemente un módulo que genere una lista a partir de la lectura de números enteros. La lectura finaliza con el número 0.
- b. Implemente un módulo recursivo que devuelva el mínimo valor de la lista.
- c. Implemente un módulo recursivo que devuelva el máximo valor de la lista.
- d. Implemente un módulo recursivo que devuelva verdadero si un valor determinado se encuentra en la lista o falso en caso contrario.

5. Escribir un programa que:

- a. Implemente un módulo que genere un vector de 20 números enteros.
- b. Implemente un módulo recursivo que devuelva el máximo valor del vector.
- c. Implementar un módulo recursivo que devuelva la suma de los valores contenidos en el vector.

6. Implementar un módulo que realice una búsqueda dicotómica en un vector, utilizando el siguiente encabezado:

*Procedure busquedaDicotomica (v: vector; ini,fin: indice; dato:integer; var pos: indice);*

Nota: El parámetro “pos” debe retornar la posición del dato o -1 si el dato no se encuentra en el vector.

7. Realizar un programa que lea números y que utilice un procedimiento recursivo que escriba el equivalente en binario de un número decimal. El programa termina cuando el usuario ingresa el número 0 (cero).

Ayuda: Analizando las posibilidades encontramos que: Binario (N) es N si el valor es menor a 2. ¿Cómo obtenemos los dígitos que componen al número? ¿Cómo achicamos el número para la próxima llamada recursiva? Ejemplo: si se ingresa 23, el programa debe mostrar: 10111.

8. Escribir un programa que:

- a. Implemente un módulo que lea números enteros y los almacene en un árbol binario de búsqueda. La lectura finaliza con el valor 0.
- b. Una vez generado el árbol, realice módulos independientes para:
  - i. Obtener el número más grande.
  - ii. Obtener el número más chico.
  - iii. Obtener la cantidad de elementos.
  - iv. Informar los números en orden creciente.
  - v. Informar los números pares en orden decreciente.

Nota: Tener en cuenta que cada número debe aparecer una única vez en el árbol.

9. Implemente un programa que invoque a un módulo que genere un árbol binario de búsqueda con nombres de personas que se leen desde teclado. La lectura finaliza con el

nombre 'ZZZ'. También debe invocar a otro módulo que reciba el árbol generado y un nombre, y retorne verdadero si existe dicho nombre en el árbol o falso en caso contrario.

**10. Implementar un programa que contenga:**

- a. Un módulo que lea información de alumnos de Taller de Programación y almacene en una estructura de datos sólo a aquellos alumnos que posean año de ingreso posterior al 2000. Esta estructura debe estar ordenada por legajo y debe ser eficiente para la búsqueda por dicho criterio. De cada alumno se lee legajo, apellido, nombre, DNI y año de ingreso.
- b. Un módulo que reciba la nueva estructura e informe el nombre y apellido de aquellos alumnos cuyo legajo sea superior a 12803.
- c. Un módulo que reciba la nueva estructura e informe el nombre y apellido de aquellos alumnos cuyo legajo este comprendido entre 2803 y 6982.

### Semana 3: Merge de listas

- 11.** Un cine posee la lista de películas que proyectará durante el mes de octubre. De cada película se conoce: código de película, código de género (1: acción, 2: aventura, 3: drama, 4: suspenso, 5: comedia, 6: bélica, 7: documental y 8: terror) y puntaje promedio otorgado por las críticas.

Implementar un programa que contenga:

- a. Un módulo que lea los datos de películas y los almacene ordenados por código de película y agrupados por código de género, en una estructura de datos adecuada. La lectura finaliza cuando se lee el código de película -1.
- b. Un módulo que reciba la estructura generada en el punto a y retorne una estructura de datos donde estén todas las películas almacenadas ordenadas por código de película.

- 12.** Implementar un programa que procese la información de las ventas de productos de una librería que tiene 4 sucursales. De cada venta se lee fecha de venta, código del producto vendido, código de sucursal y cantidad vendida. El ingreso de las ventas finaliza cuando se lee el código de sucursal 0.

Implementar un programa que:

- a. Almacene las ventas ordenadas por código de producto y agrupados por sucursal, en una estructura de datos adecuada.
- b. Contenga un módulo que reciba la estructura generada en el punto a y retorne una estructura donde esté acumulada la cantidad total vendida para cada código de producto.

- 13.** Un teatro tiene funciones los 7 días de la semana. Para cada día se tiene una lista con las entradas vendidas. Se desea procesar la información de una semana.

Implementar un programa que:

- a. Genere 7 listas con las entradas vendidas para cada día. De cada entrada se lee día (de 1 a 7), código de la obra, asiento y monto. La lectura finaliza con el código de obra igual a 0. Las listas deben estar ordenadas por código de obra de forma ascendente.

- b.** Genere una nueva lista que totalice la cantidad de entradas vendidas por obra. Esta lista debe estar ordenada por código de obra de forma ascendente.