

**Temas:**

- El tipo struct

**Objetivos:**

Que el estudiante logre...

- Emplear el tipo struct para la definición de registros de datos mediante el análisis y resolución de problemas.

**Condiciones de presentación.**

- Este trabajo práctico debe realizarse en forma individual.
- El código de los programas solicitados debe estar correctamente rotulado para su identificación, incluyendo comentarios de seguimiento y deberá ser enviado mediante la plataforma CUV.FCEYT hasta el día indicado como límite de presentación.
- La resolución completa de este trabajo práctico, incluyendo diagramas de flujo, deberá pasar a integrar la carpeta de práctica y autoevaluación del alumno.
- **Criterios de evaluación y aprobación.** Este trabajo práctico recibirá una calificación de aprobado o desaprobado. Para aprobar la presentación debe cumplir como mínimo con los siguientes ítems:
  - El trabajo práctico debe estar desarrollado completamente.
  - La codificación en lenguaje C debe realizarse siguiendo las recomendaciones de la cátedra.
  - La presentación del código de los enunciados solicitados deberá realizarse en tiempo y forma mediante la plataforma CUV.FCEYT: <http://cuv.unse.edu.ar>.

**Recursos Bibliográficos**

- Introducción al Lenguaje C - Santos Espino.
- Introducción a la Programación con C – Marzal.
- Diapositivas, lecciones, disponibles en plataforma del CUV.

**Tareas a desarrollar para cada enunciado**

- ❖ Desarrolle la diagramación de flujo y la codificación en Lenguaje C de los siguientes enunciados.
- ❖ La resolución de todos los enunciados deberá estar desarrollada en la carpeta de práctica de la asignatura.
- ❖ La entrega de los programas en lenguaje C de los enunciados indicados, deberán ser enviados mediante la plataforma CUV según las fechas y horarios de presentación de cada uno.

---

**Ejercicio 1: Calculando el promedio de horas trabajadas**

Una empresa desea registrar la información de 5 empleados y calcular el promedio de horas trabajadas en la semana. Para ello, se pide:

- Definir una estructura Empleado que contenga:
  - Nombre del empleado (cadena).
  - Legajo (entero).
  - Horas trabajadas en 5 días (arreglo de enteros).
- Ingresar por teclado los datos de los 5 empleados.
- Para cada empleado:
  - Mostrar su nombre, legajo y las horas trabajadas en cada día.
  - Calcular y mostrar el promedio de horas trabajadas en la semana.

**Ejercicio 2: Aplicando arreglos y estructuras**

Una librería desea registrar los datos de 3 libros que tiene en stock. Cada libro debe almacenar:

- Título (cadena de caracteres).
- Autor (cadena de caracteres).
- Precio (número real).

El programa debe:

- Definir una estructura Libro.
- Usar un arreglo de 3 elementos de tipo Libro.
- Solicitar al usuario ingresar los datos de los 3 libros.
- Mostrar en pantalla la información de cada libro ingresado.

<p>Ejercicios 1 y 2. Presentación: código en C. Fecha: 07/09</p>
--

### Ejercicio 3

Una cafetería universitaria desea calcular las ganancias obtenidas durante un día de trabajo por la venta de 5 tipos de productos (ejemplo: café, medialunas, sándwiches, jugos, tortas). De cada producto se debe almacenar:

- Código (entero).
- Nombre del producto (cadena de caracteres).
- Precio unitario de venta (real).
- Cantidad vendida en el día (entero).

El programa debe:

- Definir una estructura llamada Producto.
- Usar un arreglo de 5 elementos de tipo Producto.
- Solicitar al usuario ingresar los datos de cada producto.
- Calcular y mostrar la ganancia individual de cada producto (precio × cantidad vendida).
- Calcular y mostrar la ganancia total de la cafetería al final del día.

### Ejercicio 4

Escriba un programa en lenguaje C que, tratando un número complejo como una estructura de dos componentes (parteReal y parteImaginaria), permita realizar las operaciones de suma y resta entre dos números complejos. El programa deberá:

- Definir una estructura Complejo con los campos necesarios.
- Preguntar al usuario la operación a realizar (suma, resta o terminar).
- Solicitar los datos de los dos números complejos (parte real e imaginaria).
- Mostrar el resultado en la forma  $a + bi$  o  $a - bi$  según corresponda.
- Repetir estos pasos hasta que se seleccione la opción terminar.

### Ejercicio 5: Ejercicio usando struct

Un polinomio de segundo grado se puede representar de la siguiente forma:  $P(x)=ax^2+bx+c$

donde  $a$ ,  $b$  y  $c$  son coeficientes reales.

Se pide:

- Definir una estructura llamada Polinomio que contenga tres campos de tipo double para representar los coeficientes  $a$ ,  $b$  y  $c$ .
- Desarrollar un programa que permita:
  - Ingresar los coeficientes de un polinomio cuadrático.
  - Ingresar un valor de  $x$ .
  - Calcular y mostrar el valor de  $P(x)$ .
  - Calcular y mostrar las raíces reales del polinomio, aplicando la fórmula cuadrática:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

- El programa debe repetirse hasta que el usuario indique que desea terminar.
- Controlar que si el discriminante ( $b^2 - 4ac$ ) es negativo, se indique que el polinomio no tiene raíces reales.

Ejercicio 5. Presentación: código en C.  
Fecha: 07/09

## ANEXO:

El cálculo de la raíz cuadrada en lenguaje C puede realizarse usando la función `sqrt` de la librería `math.h`.

La función `sqrt` recibe un `double` representando el número al que queremos sacarle la raíz cuadrada y regresa un número `double` que indica la raíz cuadrada del número que recibió.

Por ejemplo:

```
#include <math.h>
#include <stdio.h>
int main(int argc, char const *argv[])
{
    double numero = 2525.60;
    double raizCuadrada = sqrt(numero);
    printf("La raiz cuadrada de %lf es %lf\n", numero, raizCuadrada);
    return 0;
}
```

Para elevar un número a una determinada potencia, podemos utilizar la función `pow`, de la librería `math.h`. Esta función recibe dos argumentos. El primero es el número al que se desea elevar, y el segundo argumento es la potencia a la que se elevará el número.

Por ejemplo:

```
#include <math.h>
#include <stdio.h>
int main(int argc, char const *argv[])
{
    double numero, potencia; // Aquí guardaremos lo que ingrese el usuario
    printf("Ingresa el numero:\n");
    scanf("%lf", &numero);
    printf("Escribe la potencia para elevarlo:\n");
    scanf("%lf", &potencia);
    double elevado = pow(numero, potencia);
    printf("%lf elevado a la potencia %lf es %lf\n", numero, potencia, elevado);
    return 0;
}
```