

9. Estructuras

- 1) Definir una estructura de datos que permita representar el tiempo, es decir horas, minutos y segundos. Desarrollar las funciones de manipulación del tiempo tales como:
 - a) `tiempo_crear`: genera una estructura del tipo `tiempo_t` inicializado en 0 horas, 0 minutos y 0 segundos.
 - b) `tiempo_mostrar`: imprime las horas, minutos y segundos en el siguiente formato `hh:mm:ss`
- 2) Construir una estructura de datos que represente a un automóvil. Un automóvil es una estructura que tiene los siguientes atributos:
 - `numero_dominio`: el número de patente,
 - `numero_motor`: número que trae grabado el motor,
 - `marca`: marca del automóvil
 - `modelo`: el modelo de automóvil
 - `tamano_motor`: número que indica la potencia del motor.
 - `color`: el color de la carrocería
- 3) Escribir las siguientes funciones para manipular un automóvil como el definido en el ejercicio anterior:
 - `auto_crear`: genera una instancia de tipo Automóvil con valores de inicialización adecuados para cada uno de sus atributos.
 - `auto_cambiar_color`: modifica el valor del atributo color de un automóvil que recibe
 - `auto_mostrar`: imprime todos los atributos de un automóvil en particular
 - `auto_clonar`: es una función que reproduce un automóvil en otra variable de tipo Automóvil (función `clone()`)
- 4) Escribir un programa que permita al usuario generar una lista de automóviles, los muestre ordenados según el número de dominio, permita modificar cualquiera de los atributos de un automóvil, ofrezca la posibilidad de agregar un nuevo automóvil a la lista, de eliminar un automóvil a la lista, permita determinar cuál es el automóvil de mayor potencia presente en la lista.
- 5) Definir un tipo de dato basado en estructuras que permita almacenar la información de una tarjeta de crédito: número de tarjeta, nombre del titular, fecha de expiración).
- 6) Definir un tipo de dato basado en estructuras anidadas que permita almacenar la siguiente información de un cliente:
 - Nombre y apellido.
 - ID de cliente.
 - Fecha de alta.
 - Fecha de última modificación.
 - Domicilio particular.
 - Domicilio laboral.
 - Número de tarjeta de crédito.
- 7) Idem para estructuras referenciadas.
- 8) Definir una estructura de datos que represente una fecha y el conjunto de funciones de manipulación de una fecha siguiente:
 - a) `fecha_crear`: función que devuelve una fecha inicializada en alguna fecha particular,
 - b) `fecha_mostrar`: función que imprime una fecha que recibe,
 - c) `fecha_copiar`: función que recibe una fecha y devuelve la misma en otra variable,
 - d) `fecha_dia_sig`: función que aumenta una fecha en un día,

- e) `fecha_dia_ant`: función que disminuye una fecha en un día.
- 9) Escribir un programa que permita manejar información sobre Estudiantes. Se necesita conocer el padrón (identificador único para cada Estudiante), cinco calificaciones en punto flotante, fecha de ingreso a la Facultad, Carrera en la que están inscriptos. Las funciones para manipular un Estudiante son:
- a) `estudiante_crear`: genera un Estudiante con un número de padrón que recibe.
 - b) `estudiante_agregar_calificacion`: agrega una nota de las cinco notas que puede tener un estudiante,
 - c) `estudiante_calcular_promedio`: devuelve el promedio de las notas que pertenecen a un Estudiante.
- 10) Escribir una función que permita visualizar la información de los ejercicios 7) y 8). Elegir la estrategia de impresión de forma que el agregado de un nuevo campo en la estructura tenga un impacto mínimo en el código previamente desarrollado y probado.
- 11) Escribir un programa invocable por línea de órdenes que obtenga la fecha actual del sistema y la presente por `stdout` para los siguientes formatos:
- a) AAAAMMDD
 - b) AAAAMMDD HHmmSS
 - c) HH:mm:SS del DD/MM/AAAA
- Ayuda: usar la biblioteca `<time.h>`
- 12) Modificar el ejercicio 11) para que el formato de salida pueda ser informado al programa por medio de comandos en línea de órdenes (`sysdate`).
- 13) Escribir un programa invocable en línea de órdenes que reciba dos fechas, y presente por `stdout` la diferencia entre ellas, en una unidad de tiempo indicada también como argumento en línea de órdenes (`timediff`).