**TDA String / Texto**

Realizar un TDA que represente la idea de “texto” que soporte 128 caracteres. Como ya sabemos en C no existe este tipo básico, con lo cual nuestras aplicaciones siempre quedan acopladas a la idea de que un string es una cadena de caracteres, por lo que se manejan con punteros y posiciones.

A continuación, una definición básica del TDA (pueden ustedes definir más operaciones si quisieran).

**Tipo abstracto Texto**

operacion crear() -> Texto

operacion destruir(Texto t)

función tamaño(Texto t) -> int

función caracterEn(Texto texto, int posicion) -> char

precondición: posición >= 0 y posición < tamaño(texto)

función indiceDe(Texto texto, carácter c) -> int

función estaVacio(Texto texto) -> boolean

procedimiento reemplezar(Texto texto, char caracterSale, char caracterEntra)

función aMayusculas(Texto) -> Texto

función aMinusculas(Texto) -> Texto

procedimiento concatenar(Texto textoOriginal, Texto otroTexto)

procedimiento mostrar(Texto texto)

**TDA Fecha**

Realizar un TDA Fecha, abstracción de las fechas, que consisten en un año, un mes y un día y permiten abstraer de la representación interna y operar fechas:

**Tipo abstracto Fecha**

operacion crearDesdeTextoAñoDosDigitos(char[6] fechaEnTexto) -> Fecha

operacion crearDesdeTextoAnioCompleto(char[8] fechaEnTexto) -> Fecha

operacion crear(int dia, int mes, int año) -> Fecha

funcion dia(Fecha f) -> int

funcion mes(Fecha f) -> int

funcion año(Fecha f) -> int

funcion esMayor(Fecha f, Fecha k) -> boolean

funcion esMenor(Fecha f, Fecha k) -> boolean

funcion esIgual(Fecha f, Fecha k) -> boolean

procedimiento mostrar(Fecha f)

**TDA PERIODO**

**Tipo abstracto Periodo**

operacion crear(int dias) -> Periodo

operacion crear(int meses, int dias) -> Periodo

operacion crear(int anios, int meses, dias) -> Periodo

funcion traducirADias(Periodo p) -> int

funcion traducirAMeses(Periodo p) -> float

funcion traducirAAños(Periodo p) -> float

procedimiento mostrar(Periodo p)

**TDA Boolean**

Realizar un TDA para el tipo booleano en C. Recordar que en C no existe este tipo como tipo básico, por lo cual muchas veces nuestro código pierde legibilidad y a veces el uso de números lleva a errores. En C se asume que el número 1 es verdadero y el 0 es falso.

Describimos el TDA como:

tipo abstracto boolean

operacion create(int condicionComoInt) -> boolean

operacion and(boolean b1, boolean b2) -> boolean

operacion or(boolean b1, boolean b2) -> Boolean

operacion negacion(boolean b) -> Boolean

mostrar(boolean b)

**TDA PUNTO**

Realizar un TDA para representar un punto en la pantalla de una computadora, los elementos básicos de un punto son coordenada x, coordenada y, etc.

Las operaciones básicas de este TDA son:

**Tipo abstracto Punto**

Operación crear(int coordX, int coordY);

Operación mover(int coordX, int coordY)

Operacion distancia(Punto a, Punto b);

**TDA Dupla**

Codificar el TDA para representar una "dupla". Una dupla es un conjunto ordenado de tamaño fijo de 2 elementos. Para acotar la implementación podrían realizar una Dupla de dos enteros (int), a fin de simplificarlo. Con lo cual no sería una Dupla genérica que podrá tener cualquier tipo de elemento.

Las operaciones definidas sobre la dupla deberán ser:

crear(int primero, int segundo) -> Dupla

primero(Dupla) -> int

segundo(Dupla) -> int

multiplicar(Dupla d, int multiplo) -> Dupla

dividar(Dupla d, int divisor) -> Dupla