**UTN – FR Mar del Plata - Técnico Universitario en Programación**

**Programación II y Laboratorio II**

**Trabajo Práctico N° 7 contenidos extra: Librería fecha y cadena**

Crear una librería para manejo de strings con las siguientes funciones:

void fStrDelPosChar(char c[], int pos); /// Elimina el carácter ubicado en la posición dada

void fStrLTrim(char c[]); /// Elimina espacios en blanco a la izquierda de la cadena

void fStrRTrim(char c[]); /// Elimina espacios en blanco a la derecha de la cadena

void fStrAllTrim(char c[]); /// Elimina todos los espacios en blanco de la cadena

void fStrDelete(char c[]); /// Elimina el contenido de la cadena

void fStrToUpper(char c[]); /// Convierte la cadena en mayúsculas

void fStrToLower(char c[]); /// Convierte la cadena en minúsculas

void fStrToUpperFirst(char c[]); /// Convierte la primera letra de la cadena en mayúscula

void fStrToUpperAllFirstLetters(char c[]); /// Convierte la primera letra de cada palabra en mayúscula

void fStrFirstCharReplace(char c[], char o, char r); /// Reemplaza la primer ocurrencia de un carácter dado por otro

void fStrAllCharReplace(char c[], char o, char r); /// Reemplaza todas las ocurrencias de un carácter dado por otro

int fStrFindChar(char c[], char f); /// Busca un carácter en la cadena, retorna la posición

void fStrDelFirstChar(char c[], char e); /// Elimina la primer ocurrencia de un char

void fStrDelAllChar(char c[], char e); /// Elimina todas las ocurrencias de un char

void fStrReplace(char c[], char r, int pos); /// Reemplaza un carácter en una posición dada por otro

Recuerden que una cadena es un arreglo de caracteres que siempre se pasa por referencia y que termina con el carácter de escape ‘\0’

Crear una librería para el manejo de fechas incluyendo <time.h> con esta primera función que recibe un char de 160 caracteres donde se almacena la fecha actual en el siguiente formato “2019-01-23-3 17:23:15” dónde constan el año-mes-día-día de semana hora-minutos-segundos.

void fGetNow(char date[]){

time\_t tiempo = time(0);

struct tm \*tlocal = localtime(&tiempo);

strftime(date,128,"%Y-%m-%d- %w%H:%M:%S",tlocal);

}

Y a partir de este dato deberán codificar las siguientes funciones que reciben el arreglo char date[], ojo no usen las funciones de la librería time, generen sus propias funciones. Pueden generar otras funciones de cadena si lo consideran necesario.

void fGetNow(char date[]); /// genera una cadena con la fecha completa

int fGetAnioLarge(char date[]); /// retorna el año ej: **2019**

int fGetAnioShort(char date[]); /// retorna el año ej: **19**

int fGetMonthNumber(char date[]); /// retorna el mes ej: **10**

void fGetMonthLarge(char date[],char month[]); /// genera el mes ej: **Octubre**

void fGetMonthShort(char date[],char month[]); /// genera el mes ej: **Oct**

int fGetDayNumber(char date[]); /// retorna el día ej: **23**

int fGetDayOfWeek(char date[]); /// retorna el día de semana ej: **3**

void fGetDayLarge(char date[],char day[]); /// genera el día ej: **Miercoles**

void fGetDayShort(char date[],char day[]); /// genera el día ej: **Mie**

int fGetHour(char date[]); /// retorna la hora ej: **17**

int fGetMinute(char date[]); /// retorna los minutos ej: **23**

typedef struct{

int id;

char nombre[30];

char apellido[30];

char dni[14];

}stEmpleado;

Hacer una librería con las funciones necesarias para manejar esta estructura.

Funciones recomendadas:

stEmpleado cargarUnEmpleado();

**void mostrarUnEmpleado();**

(Para realizar con memoria dinámica. Investigar toString, CSV, JSON, función sprinf() )

**char\* empleadoToString(stEmpleado e);** /// esta función convierte un empleado en un string con todos sus datos.

**char\* empleadoToCsv(stEmpleado e);**  /// ídem a la anterior pero csv (datos separados por comas)

**char\* empleadoToJson(stEmpleado e);** /// idem anterior pero formato json