



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA MEXICALI**

Formato para Prácticas de Laboratorio

PROGRAMA EDUCATIVO	PLAN DE ESTUDIO	CLAVE DE UNIDAD DE APRENDIZAJE	NOMBRE DE LA UNIDAD DE APRENDIZAJE
LSC	2009-2	11294	Programación Estructurada

PRÁCTICA No.	LABORATORIO DE	Licenciado en Sistemas Computacionales	DURACIÓN (HORAS)
1	NOMBRE DE LA PRÁCTICA	Manejo de Cadenas de Caracteres	2

1. INTRODUCCIÓN

Las cadenas de caracteres son un tipo de dato compuesto, en el cual podemos almacenar texto, este es un tipo de dato indispensable para cualquier lenguaje de programación, las cadenas de caracteres las podemos definir como un arreglo de caracteres el cual termina con el carácter "null" o "\0". En esta práctica se mostraran las funciones esenciales para la manipulación de cadenas de caracteres, así también como la manipulación de éstas como arreglo.

2. OBJETIVO (COMPETENCIA)

Elaborar un programa de cómputo, aplicando correctamente las cadenas de caracteres para la manipulación de datos alfanuméricos con actitud analítica, perseverante y responsable

3. FUNDAMENTO

Un arreglo, es un grupo de elementos del mismo tipo que se almacenan en memoria de forma consecutiva.

Las cadenas están definidas como un arreglo de caracteres o un apuntador a una porción de memoria conteniendo caracteres. Una cadena en C es una secuencia de cero o más caracteres seguidas por un carácter **NULL** o **\0**:

U	N	I	V	E	R	S	I	D	A	D	\0
---	---	---	---	---	---	---	---	---	---	---	----

Es importante preservar el carácter de terminación **NULL**, ya que con éste es como C define y maneja las

Formuló	Revisó	Autorizó
M.C. Eva Herrera Ramírez	Ing. Josefina Mariscal Camacho	Dr. David I. Rosas Almeida
Nombre y Firma del Maestro	Nombre y Firma del Responsable de Programa Educativo	Nombre y Firma del Director / Representante de la Dirección

Código: GC-N4-017 Revisión: 4



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD**

Formatos para prácticas de laboratorio

longitudes de las cadenas. Todas las funciones de la biblioteca estándar de C lo requieren para una operación satisfactoria.

Declaración de cadenas:

Las cadenas de texto se indican el tamaño máximo de la cadena + 1.

char <nombre_de_cadena> [tamaño_máximo +1];

Ejemplos:

char cabecera[10];



La variable **cabecera** puede contener una cadena de texto de tamaño máximo 9

char nombre[8];



La variable **nombre** puede contener una cadena de texto de tamaño máximo 7

Inicialización de cadenas :

char nombre[15]=" ";

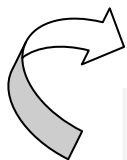


La variable **nombre** se inicializa a espacios en blanco

char nombre[]="Juan Perez ";



La variable **nombre** contiene la cadena "Juan Perez", automáticamente toma la longitud de cadena necesaria para almacenar lo que se le asigno.



Posición	0	1	2	3	4	5	6	7	8	9	10
Caracteres	J	u	a	N		P	e	r	e	z	\0

Representacion en memoria

MANEJO DE LA CADENA DE CARACTERES COMO UN ARREGLO:

De acuerdo a la definición de cadenas de caracteres estas son "arreglos", por lo mismo estas se pueden manejar como tal, como ya se vio los caracteres en memoria empiezan a almacenarse siempre desde la posición 0.



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD**

Formatos para prácticas de laboratorio

Posición	0	1	2	3	4	5	6	7	8	9	10
Caracteres	J	u	a	n		P	e	r	e	z	\0

Por lo que si se quiere imprimir únicamente el primer carácter de la cadena nombre, se escribirá:

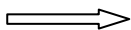
`printf("%c", nombre[0]);` \longrightarrow J

Si se quiere imprimir únicamente la última letra se escribirá:

`printf("%c", nombre[9]);` \longrightarrow z

Si se quiere imprimir carácter por carácter toda la cadena, dejando una <tab> entre cada carácter escribiríamos lo siguiente:

`for(x=0; nombre[x]; x++)`



J u a n P e r e z

`printf("%c\t", nombre[x]);`

Aunque si quiere imprimir el contenido completo, podemos utilizar la sintaxis para imprimir la cadena de caracteres normalmente:

`printf("%s", nombre);`

Todas las funciones para manejo de cadenas tienen su prototipo en:

#include <string.h>

Las funciones más comunes son descritas a continuación:

- `strcpy(cadena dest, cadena orig)` -- Copia la cadena de caracteres *orig* (incluyendo el carácter terminador '\0') a la cadena *dest*. Las cadenas no deben solaparse, y la de destino, debe ser suficientemente grande como para alojar la copia.
- `int strcmp(s1, s2)` -- Compara las dos cadenas de caracteres *s1* y *s2*. Devuelve un entero menor, igual o mayor que cero si se encuentra que *s1* es, respectivamente, menor que, igual a, o mayor que *s2*.
- `int strlen(s)` -- Calcula la longitud de la cadena de caracteres.
- `strcat(cadena dest, cadena orig)` -- Concatena la cadena de caracteres *orig* (incluyendo el carácter terminador '\0') a la cadena *dest*. Las cadenas no deben solaparse, y la de destino, debe ser suficientemente grande como para alojar su valor más lo que se le agrega.



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD**

Formatos para prácticas de laboratorio

El uso de muchas funciones es directo, por ejemplo:

```
#include <string.h>
#include <stdio.h>

int main()
{
    char s1[] = "Hola";
    char s2[5];
    int longitud;

    longitud = strlen("Hola"); /* long = 4 */

    strcpy(s2,s1); /* En s2 se copia el contenido de s1 */

    printf("%s",s2); /* s2= Hola */

    return 0;
}

#include <string.h>
#include <stdio.h>

int main()
{
    char s1[] = "Hola";
    char s2[10] = "Adios";
    int longitud;

    strcat(s2,s1); /* En s2 se agrega el contenido de s1 */
    printf("%s",s2); /* s2= AdiosHola */
    return 0;
}
```

Observar que tanto *strcat()* y *strcpy()* regresan una copia de su primer argumento, el cual es el arreglo destino. Observar también que orden de los argumentos es *arreglo destino* seguido por *arreglo fuente* lo cual a veces es una situación para hacerlo incorrectamente.

La función *strcmp()* compara lexicográficamente las dos cadenas y regresa:

- Menor que cero -- si s1 es léxicamente menor que s2;
- Cero -- si s1 y s2 son léxicamente iguales;
- Mayor que cero -- si s1 es léxicamente mayor que s2;



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD**

Formatos para prácticas de laboratorio

Ejemplo:

```
#include <string.h>
#include <stdio.h>

int main()
{
    char cad[15];
    printf("Teclea la capital de BC");
    scanf("%s",cad);
    if(strcmp(cad,"Mexicali")==0)
        printf("La capital es correcta");
    else
        printf("Error....");
    return 0;
}
```

Una biblioteca relacionada `#include <ctype.h>` la cual contiene muchas funciones útiles para convertir y probar caracteres *individuales*.

Las funciones más comunes para revisar caracteres tienen los siguientes prototipos:

- `int isalnum(c)` -- Verdad si `c` es alfanumérico.
- `int isalpha(c)` -- Verdad si `c` es una letra.
- `int isdigit(c)` -- Verdad si `c` es un dígito decimal.
- `int islower(c)` -- Verdad si `c` es una letra minúscula.
- `int isspace(c)` -- Verdad si `c` es un espacio
- `int isupper(c)` -- Verdad si `c` es una letra mayúscula.

Las funciones para conversión de caracteres son:

- `int tolower(c)` -- Convierte la letra `c` a minúsculas, si es posible
- `int toupper(c)` -- Convierte la letra `c` a mayúsculas, si es posible

4. PROCEDIMIENTO (DESCRIPCIÓN)

A) EQUIPO NECESARIO

PC con sistema operativo Linux y Visio instalado

MATERIAL DE APOYO

Práctica impresa o en USB y anteriores Apuntes de clase

B) DESARROLLO DE LA PRÁCTICA



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD**

Formatos para prácticas de laboratorio

Día Martes:

Programa 1.

Escriba un programa que lea una cadena de caracteres de una longitud máxima de 80, e imprima las letras mayúsculas en minúscula y las minúsculas en mayúsculas.

Programa 2.

Pida el significado de las siguientes palabras en ingles:

Casa

Carro

Silla

Meza

Escuela

El programa deberá imprimir la cantidad de aciertos que tuvo el usuario, para cada palabra le deberá dar 2 oportunidades.

Programa 3.

Lea el nombre, edad, dirección y teléfono de un grupo de personas y deberá imprimir los datos de la persona de menor edad.

Día Jueves:

Programa 1

Escriba un programa que lea una cadena de una longitud máxima de 80 caracteres, e imprima la cantidad de:

Letras mayúsculas

Letras minúsculas

Números

Espacios en blanco

Que están contenidas en la cadena.

Programa 2

Escriba un programa que pida los datos siguientes de un grupo de personas:

Nombre, nacionalidad y estado civil.

Y con ello imprima lo siguiente:

- Número de personas de nacionalidad Mexicana casadas
- Número de personas de nacionalidad Mexicana solteros
- Número de personas extranjeras casadas
- Número de personas extranjeras soltera



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD**

Formatos para prácticas de laboratorio

Programa 3

Escriba un programa que lea una cadena de caracteres e imprima si la cadena es un palíndromo o no lo es (un palíndromo es una palabra que se lee de igual forma de izquierda a derecha que de derecha a izquierda)

Día Viernes:

Programa 1

Escriba un programa que lea una cadena de un máximo de 80 caracteres y determine e imprima cuantas palabras forman la cadenas, considere que cada palabra puede estar separada por punto(excepto si es el final de la cadena), punto y coma, coma o espacio en blanco.

Programa 2

Escriba un programa que evalúe el siguiente examen:

¿Que se celebra el 5 de febrero?

¿Que se celebra el 5 de mayo?

¿Que se celebra el 1 de mayo?

¿Que se celebra el 21 de marzo?

En caso de responder 1 mal deberá preguntar todas las fechas una vez más, el programa terminara cuando todas estén contestadas correctamente.

Programa 3

Lea el nombre, sueldo, genero y teléfono de un grupo de personas y deberá imprimir los datos de la mujer de mayor sueldo.

C) CÁLCULOS (SI APLICA) Y REPORTE

5. RESULTADOS Y CONCLUSIONES

El resultado de las prácticas deberá visualizarse en pantalla.

6. ANEXOS



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD**

Formatos para prácticas de laboratorio

Utilización de ncurses.h

La biblioteca ncurses nos sirve para utilizar algunas funciones para el manejo de consola, como lo es posicionarse en alguna coordenada, borrar pantalla, uso de colores, etc. También podemos utilizar la función `getstr()` para la lectura de cadenas de caracteres, en este anexo esta un ejemplo sencillo del uso de la biblioteca y algunas de las funciones que esta incluye.

```
#include <ncurses.h>
int main() {
    initscr();
    printw("Hola al ncurses...");
    refresh();
    getch();
    endwin();
    return 0;
}
```

Nótese que en este programa se ha incluido: `#include <ncurses.h>` , con lo cual se tiene acceso al uso de las siguientes funciones:

- `initscr()`: se la utiliza para inicializar un ventana en modo ncurses.
- `printw()`: es para imprimir texto en la ventana. Es importante usar `printw` y `scanfw` en lugar de `printf`, `scanf`, porque se tendrían salidas erróneas debido a los buffers intermedios de linux.
- `refresh()`: sirve para refrescar la pantalla, todos los `printw` se muestran efectivamente al hacer `refresh`.
- `getch()`: espera una sola pulsación de una sola tecla (sin tener que presionar "Enter"), devolviendo el código ASCII de la tecla pulsada como un entero.
- `endwin()`: finaliza el modo de ncurses. Es importante colocar esta instrucción, de lo contrario el terminal queda desconfigurado, obligando a cerrar la consola.

Compilar el programa "Hola ncurses".

Para compilar desde consola este programa se debe utilizar el parámetro `-lncurses`.



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD**

Formatos para prácticas de laboratorio

Ejemplo getstr()

```
#include <ncurses.h>
int main() {
    char cadena[15];
    initscr();
    printw("Teclea una cadena de caracteres");
    getstr(cadena);
    printw("La cadena de caracteres es  %s",cadena);
    refresh();
    getch();
    endwin();
    return 0;
}
```

getstr(): Lee una cadena de caracteres incluyendo los espacios en blanco.

7. REFERENCIAS

<http://yaqui.mxl.uabc.mx/~eherrera/contenido.htm>