



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

Formatos para prácticas de laboratorio

CARRERA	PLAN DE ESTUDIO	CLAVE ASIGNATURA	NOMBRE DE LA ASIGNATURA
TRONCO COMÚN	2003-1	5320	PROGRAMACIÓN

PRÁCTICA No.	LABORATORIO DE	Ciencias Básicas (Programación)	DURACIÓN (HORA)
PRAC-09	NOMBRE DE LA PRÁCTICA	Cadenas de Caracteres	02:00

1. INTRODUCCIÓN

A un array (vector) de tipo ***char*** se le suele llamar **cadena de caracteres**. Una cadena de caracteres, es una formación de caracteres cuyo último elemento es el caracter nulo (ó NULL, ó '\0').

Formuló	Revisó	Aprobó	Autorizó
ING. JUAN FRANCISCO ZAZUETA APODACA L.S.C. DULCE MARÍA ÁLVAREZ SÁNDEZ L.S.C. LIZBETH JAIME SOLORIO ING. EVA HERRERA RAMÍREZ ING. HÉCTOR JUVERA VELÁSQUEZ L.S.C. ELVIRA AMALIA REZA VALDEZ ING. MARIBEL ARACELI MEJÍA GORDILS LIC. HILDA OLIVIA ALBARRÁN PADILLA L.S.C. ELVIA CRISTINA MÁRQUEZ SALGADO	M. C. ENRIQUE RENÉ BASTIDAS PUGA	M.C. MAXIMILIANO DE LAS FUENTES LARA	M.C. MIGUEL ÁNGEL MARTÍNEZ ROMERO
Maestro	Coordinador de Programa Educativo	Subdirector de la Facultad	Director de la Facultad



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

Formatos para prácticas de laboratorio

se declara:

char nombre [tamaño];

char vocal[5] = {'a', 'e', 'i', 'o', 'u'};
vocal

0	1	2	3	4	7
a	e	i	o	u	'\0'

Una cadena puede incluir letras, dígitos y varios caracteres especiales (+, @,&, etc.).

2. OBJETIVO (COMPETENCIA)

Aplicar las funciones de cadenas de caracteres, elaborando programas que manipule una cadena de caracteres, para la solución de problemas reales en el área de ingeniería. Con disposición para trabajar en forma individual teniendo responsabilidad en el uso del equipo de cómputo del laboratorio.

3. FUNDAMENTO

Un vector es una porción de memoria que es utilizada para almacenar un grupo de elementos del mismo tipo. Una secuencia ordenada de elementos de tipo *char*, es lo que conocemos como un **string** o arreglo (vector) de tipo *char*.

La declaración de una cadena de caracteres se realiza de forma similar a la de un vector de cualquier otro tipo:

char cadena[dimensión];

donde *dimensión* es una literal que indica el número de bytes que se deben reservar para la cadena (recuerde que una variable tipo ***char*** ocupa un byte).

- La asignación de valores, cuando se crea una cadena, puede ser del mismo modo que la asignación de vectores:

char cadena[7] = {'p','a','l','a','b','r','a'};

- Podemos inicializar la cadena, sin necesidad de dimensionarla:

char cadena[] = "palabra";

el compilador se encarga de reservar ocho bytes para la variable *cadena*.

- Una forma de vaciar una cadena y asignarle el valor de cadena vacía es el siguiente;

cadena[0] = '\0'; donde, '\0' es el carácter nulo, el cual indica donde termina la cadena.



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

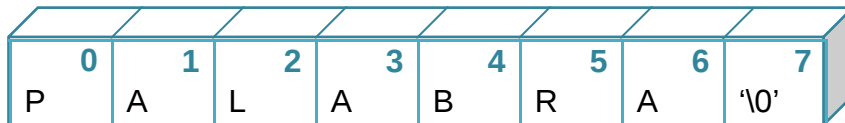
Formatos para prácticas de laboratorio

Definimos **longitud de la cadena** como el número de caracteres previos al caracter nulo. El caracter nulo no cuenta como parte para el cálculo de la longitud de la cadena. La cadena “palabra” necesita ocho variables **char** para ser almacenada, pero su longitud se dice que es siete. Asignando al elemento de índice 0 el valor nulo, tenemos una cadena de longitud cero. Cada elemento de la cadena se reconoce a través del índice entre corchetes. Cuando se quiere hacer referencia a toda la cadena en su conjunto, se emplea el nombre sin ningún corchete ni índice.

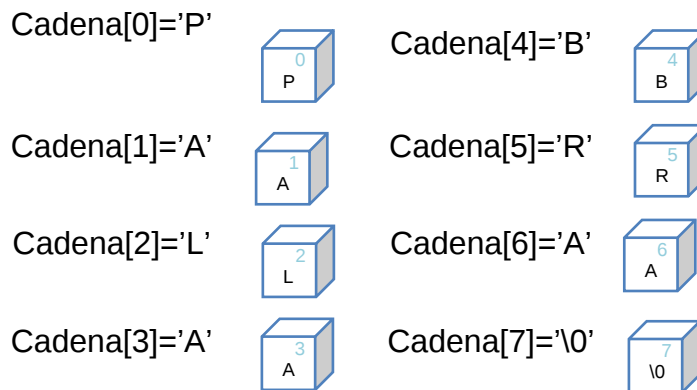
```
char cadena[ 7 ] = “palabra”;
```

El compilador inserta automáticamente un carácter nulo al final de la cadena, de modo que la secuencia real sería:

Cadena



La asignación se puede hacer también del siguiente modo:



sin embargo no se puede asignar del siguiente modo:

cadena=" PALABRA";





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

Formatos para prácticas de laboratorio

Entrada de caracteres.

Para recibir cadenas por teclado disponemos, entre otras, de la función **scanf**, esta función toma la cadena introducida por teclado, pero la corta a partir de la primera entrada de un caracter en blanco. La función **gets** está definida en la biblioteca **stdio.h**, y su prototipo es el siguiente:

gets(cadena);

Esta función asigna al string cadena todos los caracteres introducidos como cadena. La función queda a la espera de que el usuario introduzca la cadena de texto. Hasta que no se pulse la tecla intro. Al final de todos ellos, coloca el carácter nulo.

Ejemplo: programa que pregunta el nombre y, entonces saluda al usuario.

```
#include <stdio.h>
void main()
{
    char nombre[10];
    printf("¿Cómo te llamas? ");
    gets(nombre);
    printf("Hola, %s.", nombre);
}
```

hay que tener en cuenta que el nombre que se introduce puede, fácilmente, superar los 10 caracteres. Por ejemplo, si un usuario responde diciendo "José Antonio", ya ha introducido 13 caracteres: 4 por José, 7 por Antonio, 1 por el caracter en blanco, y otro más por el caracter nulo final.

getche(); espera del usuario un pulso de teclado. Devuelve el código ASCII del caracter pulsado y muestra por pantalla ese caracter.

getch(); espera del usuario un pulso de teclado. Devuelve el código ASCII del caracter pulsado. Esta función no tiene eco en pantalla, y no se ve el caracter pulsado.

puts(); toma una cadena como argumento y escribe la cadena añadiendo un caracter de línea nueva a la salida. El caracter nulo final no es escrito.

Operaciones con cadenas de caracteres y Funciones para su manejo.



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

Formatos para prácticas de laboratorio

Las operaciones que se pueden realizar con cadenas son muy diferentes a las que se realizan con vectores numéricos: concatenar cadenas, buscar una subcadena en una cadena dada, determinar cuál de dos cadenas es mayor alfabéticamente, etc.

Se encuentran definidas en la biblioteca **string.h**.

STRCPY· **Copia el contenido de una cadena en otra cadena:**

strcpy(destino,origen);

recibe como parámetros las dos cadenas, origen y destino, y devuelve la dirección de la cadena de destino.

```
#include <stdio.h>
#include <string.h>
void main()
{
    char original[100], copia[100];
    printf("Cadena original ... ");
    gets(original);
    strcpy(copia, original);
    printf("Original: %s\n",original);
    printf("Copia: %s\n",copia);
}
```

STRLEN· **Determina la longitud de una cadena:**

var_int= strlen();

Recibe como parámetro una cadena de caracteres y devuelve su longitud en una variable de tipo entero, excluyendo el carácter nulo.

```
#include <stdio.h>
#include <string.h>
void main()
{
    char original[100];
    int i;
    printf("Cadena original ... ");
    gets(original);
    i = strlen(original);
    printf("%s tiene longitud %d\n",original,i);
}
```



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

Formatos para prácticas de laboratorio

STRCAT · Concatena una cadena al final de otra:

strcat(destinoconcat,fuente);

Recibe como parámetros las cadenas *destinoconcat* de la concatenación y cadena *fuente*, y devuelve la dirección de la cadena que contiene la cadena original más la concatenada.

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

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

```
void main()
```

```
{
```

```
    char primera[100], segunda[100];
```

```
    puts("Cadena original ... ");
```

```
    gets(primeras);
```

```
    puts("Cadena a concatenar ... ");
```

```
    gets(segunda);
```

```
    strcat(primeras,segunda); /*Agrega la cadena segunda, al final de la cadena primera,
                                eliminando de primera el caracter nulo de fin de cadena*/
```

```
    printf("Texto concatenado: %s\n",primera);
```

```
}
```

También existe otra función, parecida a esta última, que concatena no toda la segunda cadena, sino hasta un máximo de caracteres, fijado por un tercer parámetro de la función:

strncat(primeras,segunda,tam);

STRCMP · Compara dos cadenas e indica cuál de ellas es mayor, menor o si son iguales:

var_int=strcmp(primeras,segunda);

Recibe como parámetros las cadenas a comparar y devuelve un valor negativo si primera < segunda; un valor positivo si primera > segunda; y un cero si ambas cadenas son iguales.

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

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

```
void main()
```

```
{
```

```
    char c1[100] = "Texto de la cadena primera";
```

```
    char c2[100] = "Texto de la cadena segunda";
```



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

Formatos para prácticas de laboratorio

```
int comp;
puts("Primera Cadena ... ");
gets(c1);
puts("Segunda Cadena ... ");
gets(c2);
comp = strcmp(c1,c2);
if(comp < 0)
    puts("cadena02 > cadena01");
else
    if(comp > 0)
        puts("cadena01 > cadena02");
    else puts("Cadenas iguales");
}
```

También existe una función que compara hasta una cantidad de caracteres señalado, es decir, una porción de la cadena:

var_int=strcmp(primer,segunda,tam);

donde *tam* es el tercer parámetro, que indica hasta cuántos caracteres se han de comparar.

Hay mucha información sobre estas y otras funciones de la biblioteca **string** en cualquier ayuda de cualquier editor de C como **STRREV, STRLWR,STRUPR, etc.**

Se recomienda consultar esa ayuda para obtener información sobre ellas.

Escribir un programa que solicite del usuario la entrada de una cadena y muestre por pantalla en número de veces que se ha introducido cada una de las cinco vocales.

```
#include <stdio.h>
void main()
{
    char cadena[100];
    int a, e, i, o, u, cont;
    puts("Introduzca una cadena de texto ... ");
    gets(cadena);
    a = e = i = o = u = 0;
    for(cont = 0 ; cadena[cont] != NULL ; cont++)
    {
        if(cadena[cont] == 'a') a++;
        else if(cadena[cont] == 'e') e++;
    }
}
```



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

Formatos para prácticas de laboratorio

```

else if(cadena[cont] == 'i') i++;
else if(cadena[cont] == 'o') o++;
else if(cadena[cont] == 'u') u++;
}
puts("La cadena introducida ha sido ...");
puts(cadena);
puts("Y las vocales introducidas han sido ... ");
printf("a ... %d\n",a);
printf("e ... %d\n",e);
printf("i ... %d\n",i);
printf("o ... %d\n",o);
printf("u ... %d\n",u);
}

```

4. PROCEDIMIENTO (DESCRIPCIÓN)

A)	EQUIPO NECESARIO	MATERIAL DE APOYO
1.-	Computadora con drive 3.5"	Práctica PRAC04
2.-	Diskette de Trabajo 3.5" doble lado, Alta densidad con protector de plástico	
3.-	Software Lenguaje Turbo C Ver 3.0	

B)

DESARROLLO DE LA PRÁCTICA

- 1.- Escribir un programa que solicite del usuario la entrada de una cadena y muestre según su elección a) esa misma cadena en mayúsculas. b) esa misma cadena en minúsculas. c) la longitud de la cadena.
- 2.- Escribir un programa que solicite del usuario la entrada de una cadena y luego la imprima habiendo eliminado de ella los espacios en blanco.
- 3.- Escribir un programa que solicite un carácter y muestre por pantalla su código ASCII. Que la solicitud de caracteres se repita hasta que se pulse la tecla ESCAPE (ASCII 27).
- 4.- Escribir un programa que solicite el nombre, apellido, carrera y edad. Con estos datos crear un correo electrónico como se indica; donde



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

Formatos para prácticas de laboratorio

@labprog.uabc es una constante para todos El programa se terminara hasta que el usuario decida.

Enrique Castro Ingeniero Industrial 31 años
email: enroids1@labprog.uabc

5.- Escribir un programa que reciba una contraseña de caracteres, valide la entrada y le dé la bienvenida al usuario. Solamente 3 intentos permitidos.

Nota: la contraseña se distingue entre mayúsculas y minúsculas

6.- Escribir un programa que solicite del usuario una cadena y evalúe si esta es un palíndromo o no. Al finalizar el proceso deberá preguntar si desea continuar o terminar el programa.

Copiar el código en un editor de C y compilarlo no sirve para aprender. Este manual es de C, y no de mecanografía.

C)	CÁLCULOS Y REPORTE
-----------	---------------------------

- El alumno deberá depurar sus programas para eliminar los errores de compilación y ejecutar los programas con diversos valores, de tal forma que valide los resultados esperados.

5. RESULTADOS Y CONCLUSIONES

- El maestro revisará los programas proporcionando diferentes valores para determinar si el programa se ejecuta correctamente.

6. ANEXOS

Ninguno.



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

Formatos para prácticas de laboratorio

7. REFERENCIAS

Ninguna.