

Cadenas

1. Describa qué es para usted una cadena.
2. ¿Qué diferencias y similitudes hay entre una cadena y un arreglo?
3. ¿De qué sirve el carácter `'\0'` en las cadenas?
4. ¿Dónde empieza y dónde termina la cadena almacenada en el arreglo `str` si se lo inicializa con el siguiente arreglo?

```
{'9', '5', '1', '1', '\n', '7', '5', '\0', '2', '\0', 'R', 'u', 'L', 'Z',  
'\0', 'a', ' ', 'v', 'e', 'c', 'e', 's'}
```

¿qué imprimiría la función `puts()`? ¿Termina en `'\0'` la cadena?

5. Escriba 3 porciones de código que inicialicen de 3 maneras distintas la variable `str` con la cadena `"Winter is coming"`. Comente las diferencias entre las 3 formas.
6. Para cada inciso, explique las diferencias en las definiciones mostradas:

a.

```
char manzana[] = "manzana";  
char * naranja = "naranja";
```

b.

```
char * frutas[] = {"uva", "kiwi", "caqui", "higo", "pera",  
"frambuesa"};  
char frutas[][10] = {"uva", "kiwi", "caqui", "higo", "pera",  
"frambuesa"};
```

Trabajando con cadenas

1. Escriba un programa que lea una cadena de caracteres ingresada por el flujo de entrada estándar y la imprima por pantalla. Almacena la cadena en un arreglo de no más de 100 caracteres. Utilice los siguientes métodos:
 - a. Itere hasta leer el carácter `'\n'` **inclusive**,
 - b. Lea utilizando `fgets()`.
2. Escriba un programa que lea una cadena de caracteres e imprima por pantalla, para cada elemento de la cadena: su posición dentro de la cadena, el carácter y su código en representación ASCII. Por ejemplo:

```

$ ./analizar-cadena
Ingrese la cadena: Fire and blood.
0      F      70
1      i     105
2      r     114
3      e     101
4              32
5      a      97
6      n     110
7      d     100
8              32
9      b      98
10     l     108
11     o     111
12     o     111
13     d     100
14     .      46
15              0

```

3. (**ctype.h**) Escriba un programa que lea una cadena de caracteres e imprima una tabla como la siguiente:

```

$ ./g06e10
Ingrese la cadena: 1. Usen ctype.h!
      i
      i i i s i i i i i i i
      s s s x s s s s s s s
      d a a d l u s c p p g
      i l l i o p p n u r r
      g p n g w p a t n i a
      i h u i e e c r c n p
      t a m t r r e l t t h
1 *      * *
.
      *
U      * *      *
s      * *      *
e      * * * *
n      * *      *
      *
c      * * * *
t      * *      *
y      * *      *
p      * *      *
e      * * * *
.
h      * *      *
!
?      * *

```

Tip

para imprimir la tabla exactamente igual, deberá reemplazar los caracteres no imprimibles por un signo de pregunta, como se ve en la última fila.

4. Escriba un programa que lea una cadena de caracteres y verifique si la misma se encuentra vacía.
5. Implemente sus propias versiones de las funciones de la biblioteca *string.h* que se detallan:

- a. `size_t strlen(const char *);`
- b. `char * strcpy(char *dest, const char *orig);`
- c. `char * strncpy(char *dest, const char *orig, size_t n);`
- d. `int strcmp(const char *s1, const char *s2);`
- e. `int strncmp(const char *s1, const char *s2, size_t n);`
- f. `char * strcat(char *dest, const char *orig);`
- g. `void * memmove(void *dest, const void *src, size_t n);`
- h. `void * memcpy(void *dest, const void *src, size_t n);`
- i. `int memcmp(const void *s1, const void *s2, size_t n);`

Si bien no pertenecen al estándar ISO sino al BSD:

- a. `int strcasecmp(const char *s1, const char *s2);` que funciona como `strcmp()` pero es *case insensitive*.
- b. `int strncasecmp(const char *s1, const char *s2, size_t n);`

6. Escriba un programa que lea una cadena de caracteres e imprima su longitud. Considere los siguientes casos de ejecución:

Usos con entrada correctamente formada

```
$ ./g06e08
Fire and blood.
16
$ ./g06e08
~!@#$%*( )_+| ]{ : , < z
19
$ ./g06e08

1
$ ./g06e08
0
$ ./g06e08
Fire and blood.15
```

7. Implemente una función que reciba una cadena y la invierta.

```
char * invertir(char *);
```

8. Implemente una función que reciba una cadena y retorne un booleano indicando si la misma es palíndroma o no.

```
bool_t es_palindroma(const char *);
```

Tip

puede considerar, haciendo el ejercicio un poco más difícil, el preprocesado de la cadena: eliminar los espacios, los puntos, las comas, etc., los signos de interrogación, exclamación, etc., quedándose únicamente con los caracteres alfanuméricos en minúsculas (o mayúsculas).

9. Implemente una función que reciba una cadena y:

a. la convierta a mayúsculas: `char *strupper(char *)`;

b. la convierta a minúsculas: `char *strlower(char *)`;

c. invierta el *casing*: `char *strinvertcase(char *)`;

Los caracteres que se encuentren en mayúsculas los convierta a minúsculas y los que están en minúsculas a mayúsculas.

Observación: la cadena puede contener caracteres que no sean letras.

d. la convierta a *Title Case*: `char *titlecase(char *)`;

e. la convierta a *snake_case*: `char *snake_case(char *)`;

f. la convierta al horrible *CamelCase*: `char *puaj(char *)`;

g. borre todos los espacios a izquierda: `char *left_trim(char *)`;

h. borre todos los espacios a derecha: `char *right_trim(char *)`;

i. borre todos los espacios a izquierda y derecha: `char *full_trim(char *)`;

j. la ajuste a n caracteres por línea: `char *justify(char *, size_t n)`;

k. la centre agregando caracteres a izquierda y derecha: `char *center(char *, size_t n)`;

10. Implemente una función que reciba una cadena de caracteres, posiblemente representando un número entero, y convierta su contenido al número que representa.

Similar a `strtol()`.

11. Implemente una función que reciba un número y lo cargue en un arreglo de caracteres recibido como argumento. *Sugerencia: utilice la función `sprintf()` de la biblioteca estándar de entrada y salida.*

12. Escriba una función que reciba un número entero positivo y un arreglo de caracteres y cargue en este último la representación en binario del primero. Implemente además la

función inversa, que recibe una representación de un número en binario y carga un número con el valor que ésta representa.

13. Escriba una función que recibe dos cadenas y un número e inserta una cadena dentro de la otra en la posición dada por el número.

Por ejemplo:

- `s1 = "Frase corta", s2 = " menos", n = 5 -> s1 = "Frase menos corta"`
- `s1 = "Frase corta", s2 = " menos", n = 23 -> error.`
- `s1 = "Frase corta", s2 = " menos", n = -8 -> error.`

Arreglos de cadenas

Llamaremos *tablas de búsqueda* al conjunto {tipo enumerativo, arreglo} que tengan la misma cantidad de elementos, ordenandos de la misma forma, como se hará en los ejercicios siguientes.

1. Defina un arreglo de cadenas que contenga 10 frutas. Defina un tipo enumerativo `fruta_t` que contenga las 10 frutas mencionadas en el arreglo, en el mismo orden. Luego implemente un programa o función que imprima un elemento del arreglo, dado por una variable del tipo `fruta_t`.
2. Defina un tipo enumerativo `status_t` con los valores:

- `ST_OK,`
- `ST_ERROR_RADIO_NEGATIVO,`
- `ST_ERROR_ALTURA_NEGATIVA,` y
- `ST_ERROR_PUNTERO_NULO.`

Defina un arreglo de cadenas constante que contenga las cadenas:

- `"Ok",`
- `"El radio del cilindro no puede ser negativo",`
- `"La altura del cilindro no puede ser negativa",` y
- `"Se recibio un puntero nulo".`

Implemente una función, `imprimir_estado()`, que reciba una variable de tipo `status_t` e imprima el estado correspondiente por el flujo estándar de errores.

3. De forma similar, defina el tipo enumerativo `dia_t` con los valores
- `DIA_LUNES`
 - `DIA_MARTES`
 - `DIA_MIERCOLES`

- `DIA_JUEVES`
- `DIA_VIERNES`
- `DIA_SABADO`
- `DIA_DOMINGO`

Defina un arreglo de cadena con los 7 días de la semana, de manera similar al ejercicio anterior. Modularice el programa de forma tal que sea sencillo cambiar el idioma en el que se muestran los días, agregando únicamente archivos .h.

Peligro

Recuerde que no se definen variables en los archivos de encabezados.

4. La [clasificación espectral](#) de estrellas (cuerpos celestes) permite agruparlas en función de su temperatura, como se muestra en la tabla siguiente

Clase	Temperatura	Color
O	28000 - 50000 K	Azul
B	9600 - 28000 K	Blanco azulado
A	7100 - 9600 K	Blanco
F	5700 - 7100 K	Blanco amarillento
G	4600 - 5700 K	Amarillo
K	3200 - 4600 K	Naranja
M	1700 - 3200 K	Rojo

- Defina el tipo enumerativo `espectral_t` con los símbolos correspondientes a las clases espectrales.
- Declare un arreglo de cadenas con los colores de cada tipo de estrella.
- Escriba un programa que le pida al usuario la temperatura de un cuerpo celeste y almacene la clase del mismo en una variable del tipo `espectral_t`.
- Imprima el color de la estrella.

5. Implemente una función que reciba un arreglo de palabras, un arreglo de definiciones y una palabra e imprima la definición de la palabra recibida.
6. Escriba una función que reciba y ordene un arreglo de cadenas de caracteres. Fuera de la función, imprima el vector ordenado.
7. **(menú)** En los siguientes items deberá implementar una función que muestre un menú en la terminal. Veremos distintas formas de hacerlo.
 - a. Implemente una función que reciba un arreglo de cadenas, constantes, que representan las opciones del menú, e imprima el menú por pantalla. El último elemento del arreglo debe ser el puntero nulo. La función debe retornar la opción seleccionada.
 - b. Implemente una función que reciba como argumentos una cadena y un arreglo de cadenas. La cadena recibida es el mensaje que se muestra al usuario, antes de mostrar el menú. El resto funciona como el inciso a.
 - c. Implemente una función que reciba un arreglo de cadenas representando las opciones del menú y un arreglo de caracteres que representa lo que el usuario debe presionar para seleccionar la opción. Retorne la opción seleccionada.
 - d. Implemente una función que reciba un arreglo de cadenas representando las opciones del menú y muestre un menú utilizando como opción el primer carácter alfanumérico de cada cadena. ¿Qué se podría hacer en caso de tener dos o más opciones con el mismo carácter inicial?
8. **(búsqueda)** Una tarea muy común, es buscar un elemento en un arreglo de datos. Un ejemplo típico se da en los argumentos de un programa, donde las posibilidades son un conjunto finito de datos y el usuario debe proporcionar parámetros válidos.
 - Implemente una función que reciba un arreglo de cadenas constantes, y una cadena, y retorne la posición de la cadena recibida dentro del arreglo.
 - Implemente una función que reciba un arreglo de números y un número, y retorne la posición del número recibido dentro del arreglo.
 - Implemente una función que reciba un arreglo de punteros a números y un número, y retorne la posición del número recibido dentro del arreglo.
 - Compare las 3 funciones implementadas.
 - Implemente una función que reciba un arreglo de punteros void y un puntero, y retorne la posición, dentro del arreglo, del elemento que apunta a la misma posición que el puntero recibido. Compare esta nueva función con las anteriores ¿hace lo mismo? ¿en qué se diferencia?

Aplicaciones

Info

En los ejercicios que siguen, si bien se pueden ejecutar normalmente y procesarlos, la salida será más clara si se imprime por pantalla el contenido de un archivo de texto y la salida se redige a la entrada del programa a desarrollar. Por ejemplo:

```
cat archivo.txt | ./g06e30
```

1. Escriba un programa que lea cadenas del flujo de entrada estándar hasta que el mismo se cierra o se produce un error e imprima cada una de las líneas leídas.
2. Escriba un programa que lea cadenas del flujo de entrada estándar hasta que el mismo se cierra o se produce un error e imprima cada una de las líneas leídas numerada.
3. Escriba un programa que lea cadenas del flujo de entrada estándar hasta que el mismo se cierra o se produce un error e imprima todas las líneas que **no** comienzan con el carácter '#'.
4. Escriba un programa que lea cadenas del flujo de entrada estándar hasta que el mismo se cierra o se produce un error e imprima todas las líneas que **no** comienzan con el carácter '#' y, si el carácter '#' aparece en otro lugar que no sea el comienzo, no imprima la línea de ese punto en adelante.

Variante: admita que el carácter '#' aparezca si se encuentra entre comillas (""), por ejemplo: "quiero que aparezca el # que figura aca".