



UNIVERSIDAD NACIONAL DE INGENIERÍA

Facultad de Ciencias

Escuela Profesional de Ciencia de la Computación

► **Curso: Fundamentos de Programación**

► **Docente: Américo Chulluncuy Reynoso**

2025-II

Sesión 7:

Cadena de caracteres II

Contenido

1. Funciones para el manejo de cadenas
2. Funciones para caracteres
3. Funciones para convertir cadenas de caracteres

1. Funciones para el manejo de cadenas

Función	Descripción
✓ <code>strlen(s)</code>	Retorna la longitud (número de caracteres no nulos) de <code>c</code> .
✓ <code>strcpy(s1, s2)</code>	Modifica <code>s1</code> reemplazando sus caracteres por copias de los caracteres de <code>s2</code> .
<code>strncpy(s1, s2, n)</code>	Modifica <code>s1</code> reemplazando sus caracteres por los <code>n</code> primeros caracteres de <code>s2</code> . Si <code>strlen(s2)</code> es menor que <code>n</code> , entonces <code>s1</code> es rellenado con 'ceros' hasta que un total de <code>n</code> hayan sido escritos en él.
✓ <code>strcat(s1, s2)</code>	Modifica <code>s1</code> concatenando <code>s2</code> al final de <code>s1</code> .
<code>strncat(s1, s2, n)</code>	Modifica <code>s1</code> concatenando con los <code>n</code> primeros caracteres de <code>s2</code> . Si <code>strlen(s2)</code> es menor que <code>n</code> , entonces <code>s1</code> es rellenado con 'ceros' hasta que un total de <code>n</code> hayan sido concatenados al final de él.
✓ <code>strcmp(s1, s2)</code>	Compara <code>s1</code> y <code>s2</code> , retornando un entero negativo, <u>0</u> , o un entero positivo según <code>s1</code> es menor que, igual a, o mayor que <code>s2</code> . ==
✓ <code>strstr(s1, s2)</code>	Busca en <code>s1</code> por la primera ocurrencia de <code>s2</code> , <u>retornando un puntero al primer</u> caracter de esta ocurrencia o <code>nullptr</code> si <code>s2</code> no es encontrado en <code>s1</code> .

Debemos incluir la librería:

```
#include <cstring>
```

Ejemplo 1

Implemente una función que invierta una cadena (no utilice una cadena auxiliar)

```
void invertirCadena(char *cadena) {  
    char *inicio = cadena;  
    char *fin = cadena + strlen(cadena) - 1;  
    while (inicio < fin) {  
        char temp = *inicio;  
        *inicio = *fin;  
        *fin = temp;  
        //actualizamos los punteros  
        inicio++;  
        fin--;  
    }  
}
```

Ejemplo 2

Dado una lista de nombres almacenados en un arreglo de cadenas, utilice punteros a cadenas de caracteres y

1. Escriba una función que determine si un nombre está o no en el arreglo
2. Escriba una función que permita concatenar 2 nombres elegidos
3. Escriba una función que permita encontrar una subcadena

```
bool nombreEnArrerglo(const char *nombres[], int n, const char *nombre) {  
    for (int i = 0; i < n; ++i) {  
        if(strcmp(*(nombres + i),nombre) == 0){  
            return true;  
        }  
    }  
  
    return false;  
}
```

Ejemplo 3

Dada la siguiente cadena: "--Aqui tenemos, un ejemplo de cadena.", escribir una función que elimine los caracteres - , y .

La salida debe ser: "Aqui tenemos un ejemplo de cadena"

```
void eliminaCaracteres(char * cadena) {  
    char * recCadena = cadena;  
    char * resultado = cadena;  
    while (*recCadena != '\0') {  
        if (*recCadena != '-' && *recCadena != '.' && *recCadena != ',') {  
            *resultado = *recCadena;  
            resultado++;  
        }  
        recCadena++;  
    }  
    *resultado = '\0';  
}
```


2. Funciones para el manejo de caracteres

```
#include <cstring>
```

Función	Descripción
islower(ch)	Retorna true si ch es minúscula y false caso contrario.
isupper(ch)	Retorna true si ch es mayúscula y false caso contrario.
tolower(ch)	Retorna la minúscula equivalente a ch si ch es mayúscula.
toupper(ch)	Retorna la mayúscula equivalente a ch si ch es minúscula.

Ejemplo 4

Escribir una función que convierta una cadena de caracteres de modo que las minúsculas se convierten en mayúsculas y viceversa, mientras que los no alfabéticos se mantienen sin cambios.

Ejemplo

Entrada: "Veamos Su funCionamiento #@ 123! "

Salida: "vEAMOS sU FUNcIONAMIENTO #@ 123! "

3. Funciones para convertir cadenas

Asociar \rightarrow integer

Función	Descripción
<code>atoi(s)</code>	Convierte <code>s</code> a un <code>int</code> (si es posible) y retorna este valor.
<code>atof(s)</code>	Convierte <code>s</code> a un <code>double</code> (si es posible) y retorna este valor.
<code>atol(s)</code>	Convierte <code>s</code> a un <code>long int</code> (si es posible) y retorna este valor.

Ejemplo 5

Escribir un programa que reciba la siguiente cadena

"123, 45.67, -89, 1001, 23.45, 6789, 12.45"

y calcule la suma de todos los números enteros contenidos en ella

```
int sumarEnteros(char * cadena) {  
    int suma = 0;  
    char* token = strtok(cadena, ", ");  
  
    while (token != nullptr) {  
        double n = atof(token);  
        if (floor(n) == n) {  
            suma += n;  
            cout << n << endl;  
        }  
        token = strtok(nullptr, ", ");  
    }  
    return suma;  
}
```


Ejercicios

1. Escribe una función que recorra una cadena de caracteres utilizando un puntero y vaya imprimiendo cada carácter en la pantalla.
2. Implementa una función que calcule la longitud de una cadena de caracteres utilizando punteros.
3. Crea una función que compare dos cadenas de caracteres utilizando punteros y devuelva un valor que indique si son iguales o no.
4. Escribe una función que copie una cadena de caracteres en otra utilizando punteros.

Ejercicios

5. Implementa una función que concatene dos cadenas de caracteres utilizando punteros.
6. Escribe una función que invierta una cadena de caracteres utilizando punteros.
7. Implementa una función que elimine los caracteres duplicados de una cadena de caracteres utilizando punteros.
8. Escribe una función que cuente el número de palabras en una cadena de caracteres utilizando punteros.

Ejercicios

9. Implementa una función que reemplace todas las ocurrencias de una subcadena con otra subcadena en una cadena dada, utilizando punteros.
10. Escribe una función que invierta una cadena de caracteres utilizando punteros.
11. Implementa una función que elimine los caracteres duplicados de una cadena de caracteres utilizando punteros.
12. Escribe una función que cuente el número de palabras en una cadena de caracteres utilizando punteros.

Ejercicios

13. Implementa una función que reemplace todas las ocurrencias de una subcadena con otra subcadena en una cadena dada, utilizando punteros.
14. Escribe una función que tome una cadena de entrada y un delimitador, y devuelva un arreglo de punteros a tokens.
15. Modifica la función anterior para que cuente el número de palabras en lugar de simplemente imprimir los tokens.
16. Crea una función que busque un token específico en una cadena y lo reemplace por otro token dado.
17. Escribe una función que tome una cadena de entrada con palabras separadas por un delimitador y las ordene alfabéticamente.

```

#include <iostream>
#include <cstring>
using namespace std;

void intercambio(char *a, char *b){
    char *temp;
    temp = a;
    a = b;
    b = temp;
}

void ordBurbuja(char *w[], int n){
    for(int i = 0; i < n - 1; i++){
        for(int j = 0; j < n - i - 1; j++){
            if(strcmp(w[j], w[j+1]) > 0){
                intercambio(w[j], w[j+1]);
            }
        }
    }
}

int main(){

    char v[][10] = {"Raul", "Luis", "Alberto", "Juan"};
    cout << sizeof(v) << endl;
    const char *w[] = {"Raul", "Luis", "Alberto", "Juan"};
    cout << sizeof(w) << endl;
    //ordBurbuja();

    return 0;
}

```