

UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE

Nombre: Richard Casa NRC: 29583

Fecha: 08/01/2026

Tema: Cadenas

1. Leer una frase y contar cuántas letras y cuántos espacios tiene.

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

// Función que recibe un puntero a cadena y cuenta letras y espacios
void contarLetrasEspacios(char *cadena, int *letras, int *espacios) {
    *letras = 0;
    *espacios = 0;

    while (*cadena != '\0') {
        if (isalpha(*cadena)) { // Usando isalpha() de ctype.h
            (*letras)++;
        } else if (*cadena == ' ') {
            (*espacios)++;
        }
        cadena++;
    }
}

int main() {
    char frase[100];
    int letras, espacios;

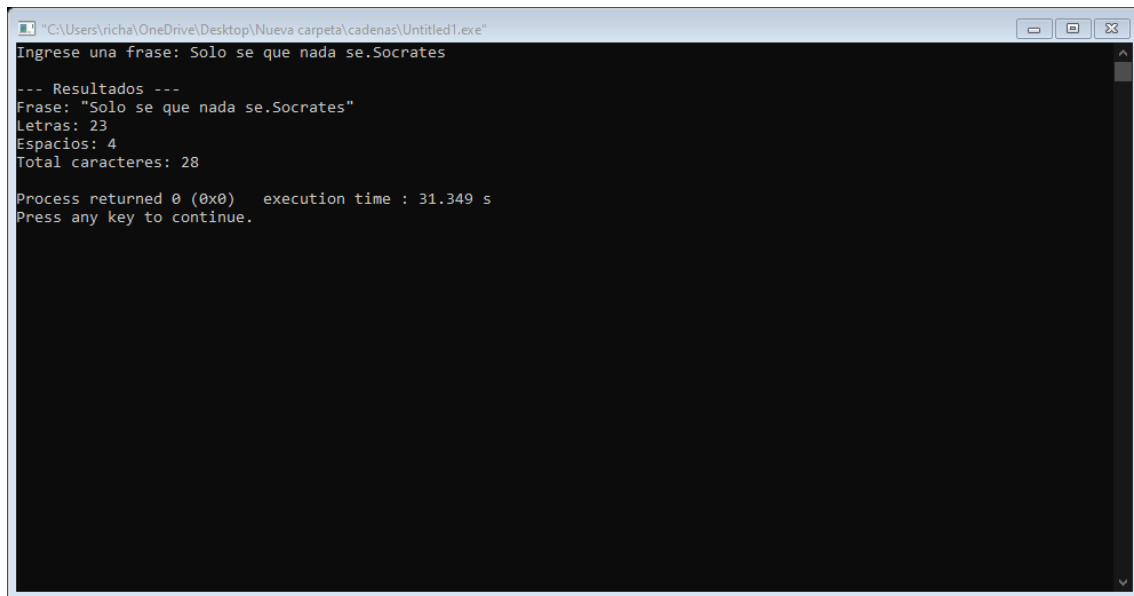
    printf("Ingrese una frase: ");
    fgets(frase, sizeof(frase), stdin);

    // Eliminar el salto de línea
    frase[strcspn(frase, "\n")] = 0;

    // Llamar a la función con puntero
    contarLetrasEspacios(frase, &letras, &espacios);

    printf("\n--- Resultados ---\n");
    printf("Frase: \"%s\"\n", frase);
    printf("Letras: %d\n", letras);
    printf("Espacios: %d\n", espacios);
    printf("Total caracteres: %zu\n", strlen(frase));

    return 0;
}
```



```
"C:\Users\richa\OneDrive\Desktop\Nueva carpeta\cadenas\Untitled1.exe"
Ingrese una frase: Solo se que nada se.Socrates

--- Resultados ---
Frase: "Solo se que nada se.Socrates"
Letras: 23
Espacios: 4
Total caracteres: 28

Process returned 0 (0x0)   execution time : 31.349 s
Press any key to continue.
```

<https://onlinegdb.com/O3Ffxm5hR>

2. Validar si una cadena es un palíndromo (ignorando espacios).

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

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

```
#include <ctype.h>
```

```
// Función para quitar espacios de una cadena
```

```
void eliminarEspacios(char *destino, const char *origen) {
```

```
    char *d = destino;
```

```
    const char *s = origen;
```

```
    while (*s != '\0') {
```

```
        if (*s != ' ') {
```

```
            *d = tolower(*s); // Convertir a minúscula
```

```
            d++;
```

```
        }
```

```
        s++;
```

```
    }
```

```
    *d = '\0'; // Terminar la cadena
```

```
}
```

```
// Función para verificar palíndromo usando punteros
```

```
int esPalindromo(const char *cadena) {
```

```
    const char *inicio = cadena;
```

```
    const char *fin = cadena + strlen(cadena) - 1;
```

```
    while (inicio < fin) {
```

```
        if (*inicio != *fin) {
```

```
            return 0; // No es palíndromo
```

```
        }
```

```
        inicio++;
```

```
        fin--;
```

```
    }
```

```
    return 1; // Es palíndromo
```

```
}
```

```
int main() {
```

```
    char frase[100];
```

```
    char sinEspacios[100];
```

```
    printf("Ingrese una frase: ");
```

```
    fgets(frase, sizeof(frase), stdin);
```

```
    // Eliminar el salto de línea
```

```
    frase[strlen(frase)] = 0;
```

```
    // Crear una versión sin espacios y en minúsculas
```

```
    eliminarEspacios(sinEspacios, frase);
```

```
    printf("\nFrase original: \"%s\"", frase);
```

```
    printf("Frase sin espacios: \"%s\"", sinEspacios);
```

```

// Verificar si es palíndromo
if (esPalindromo(sinEspacios)) {
    printf("La frase ES un palíndromo (ignorando espacios)\n");
} else {
    printf("La frase NO es un palíndromo\n");
}

return 0;
}

```

```

C:\Users\richa\OneDrive\Desktop\Nueva carpeta\cadenas\Untitled 2.exe
Ingrese una frase: anita lava la tina
Frase original: "anita lava la tina"
Frase sin espacios: "anitalavalatina"
La frase ES un palíndromo (ignorando espacios)
Process returned 0 (0x0) execution time : 33.739 s
Press any key to continue.

```

<https://onlinegdb.com/z7btiCNo1>

3. Unir nombre y apellido en una sola cadena con un espacio entre ambos.

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

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

```
// Función para concatenar dos cadenas con un espacio entre ellas
```

```

void unirCadenas(char *destino, const char *cadena1, const char *cadena2) {
    char *d = destino;
    const char *s = cadena1;

```

```
// Copiar primera cadena
while (*s != '\0') {
    *d = *s;

    d++;
    s++;
}

// Agregar espacio
*d = ' ';
d++;

// Resetear puntero para segunda cadena
s = cadena2;

// Copiar segunda cadena
while (*s != '\0') {
    *d = *s;

    d++;
    s++;
}

// Terminar la cadena
*d = '\0';
}

int main() {
    char nombre[50];
    char apellido[50];
    char nombreCompleto[100];
```

```

printf("Ingrese su nombre: ");
fgets(nombre, sizeof(nombre), stdin);
nombre[strcspn(nombre, "\n")] = 0;

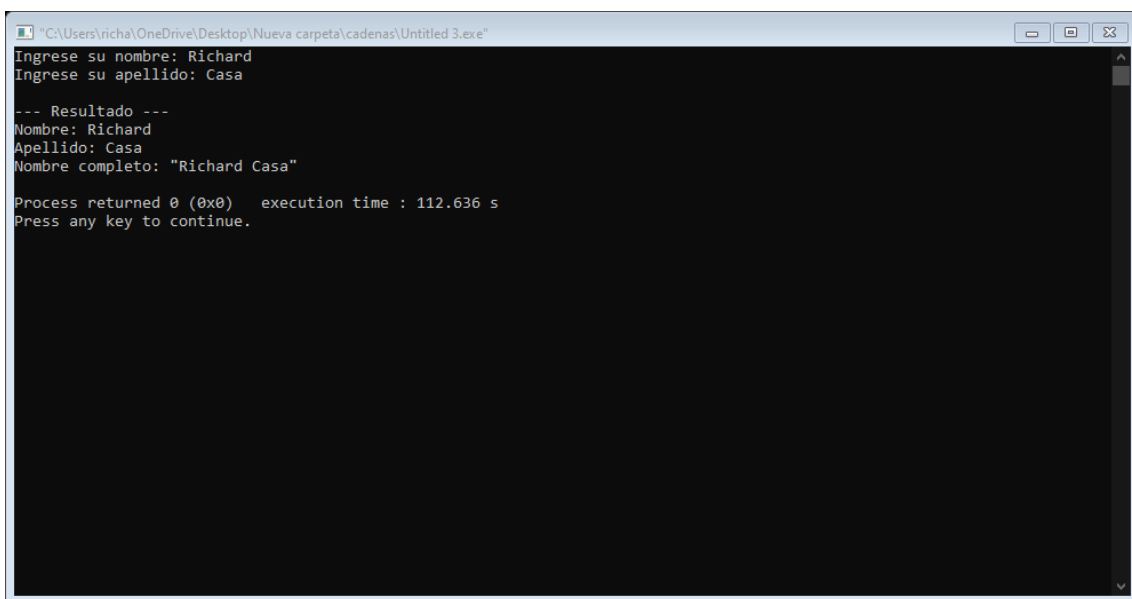
printf("Ingrese su apellido: ");
fgets(apellido, sizeof(apellido), stdin);
apellido[strcspn(apellido, "\n")] = 0;

// Usar función personalizada
unirCadenas(nombreCompleto, nombre, apellido);

printf("\n--- Resultado ---\n");
printf("Nombre: %s\n", nombre);
printf("Apellido: %s\n", apellido);
printf("Nombre completo: \"%s\" \n", nombreCompleto);

return 0;
}

```



```

C:\Users\richa\OneDrive\Desktop\Nueva carpeta\cadenas\Untitled 3.exe
Ingrese su nombre: Richard
Ingrese su apellido: Casa

--- Resultado ---
Nombre: Richard
Apellido: Casa
Nombre completo: "Richard Casa"

Process returned 0 (0x0)   execution time : 112.636 s
Press any key to continue.

```

<https://onlinegdb.com/yFmNwx93N>

4. Buscar una palabra dentro de un texto e indicar la posición donde aparece por primera vez.

```
#include <stdio.h>

#include <string.h>
```

```
int main() {

    char t[100], p[50];

    int i, pos = -1;


    printf("Texto: "); gets(t);
    printf("Palabra: "); gets(p);


    for (i = 0; t[i]; i++) {
        if (strncmp(&t[i], p, strlen(p)) == 0) {
            pos = i;
            break;
        }
    }


    printf(pos != -1 ? "Posicion: %d\n" : "No encontrado\n", pos);


    return 0;
}
```

```
"C:\Users\richa\OneDrive\Desktop\Nueva carpeta\cadenas\Untitled 4.exe"
Texto: solo se que nada se.Socrates
Palabra: Socrates
Posicion: 20

Process returned 0 (0x0)   execution time : 33.491 s
Press any key to continue.
```

<https://onlinegdb.com/SHjxAVzgR>

5. Convertir una cadena numérica a entero y calcular su cuadrado.

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

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

```
int main() {
```

```
    char numero_str[20];
```

```
    int numero_int = 0;
```

```
    int i, digito;
```

```
    printf("Ingrese un numero como cadena: ");
```

```
    fgets(numero_str, sizeof(numero_str), stdin);
```

```
    numero_str[strcspn(numero_str, "\n")] = 0;
```

```
    // Convertir cadena a entero
```

```
    for (i = 0; numero_str[i] != '\0'; i++) {
```

```
        // Verificar si es un dígito
```

```
        if (numero_str[i] >= '0' && numero_str[i] <= '9') {
```

```
            digito = numero_str[i] - '0'; // Convertir char a int
```

```
            numero_int = numero_int * 10 + digito;
```

```
        } else {
```

```
            printf("Error: Caracter no valido '%c'\n", numero_str[i]);
```

```
            return 1;
```

```
        }
```

```
    }
```

```
    printf("\nResultados:\n");
```

```
    printf("Cadena original: \"%s\"\n", numero_str);
```

```
    printf("Numero entero: %d\n", numero_int);
```

```
    printf("Cuadrado: %d * %d = %d\n", numero_int, numero_int, numero_int *  
numero_int);
```



```
return 0;
```

```
}
```

<https://onlinegdb.com/P37mZ7Dcp>

6. Invertir una cadena y mostrar la original y la invertida

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

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

```
int main() {
```

```
    char cadena[100];
```

```
    char invertida[100];
```

```
    int i, j, longitud;
```

```
    printf("Ingrese una cadena: ");
```

```
    fgets(cadena, sizeof(cadena), stdin);
```

```
    cadena[strlen(cadena)] = 0;
```

```
    longitud = strlen(cadena);
```

```
    // Invertir la cadena
```

```
    j = 0;
```

```
    for (i = longitud - 1; i >= 0; i--) {
```

```
        invertida[j] = cadena[i];
```

```

        j++;
    }

    invertida[j] = '\0'; // Terminar la cadena invertida

    printf("\nResultados:\n");
    printf("Original: \"%s\"\n", cadena);
    printf("Invertida: \"%s\"\n", invertida);
    printf("Longitud: %d caracteres\n", longitud);

    return 0;
}

```

```

C:\Users\richa\OneDrive\Desktop\Nueva carpeta\cadenas\Untitled 6.exe
Ingrese una cadena: 123456

Resultados:
Original: "123456"
Invertida: "654321"
Longitud: 6 caracteres

Process returned 0 (0x0)   execution time : 19.679 s
Press any key to continue.

```

<https://onlinegdb.com/hfQYNthiU>

7. Crear una función que reciba una cadena y devuelva cuántas vocales tiene.

```

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

```

```

int main() {
    char texto[100];
    int vocales = 0;
    int i;

    printf("Ingrese un texto: ");
    fgets(texto, sizeof(texto), stdin);
    texto[strcspn(texto, "\n")] = 0;
}

```

```

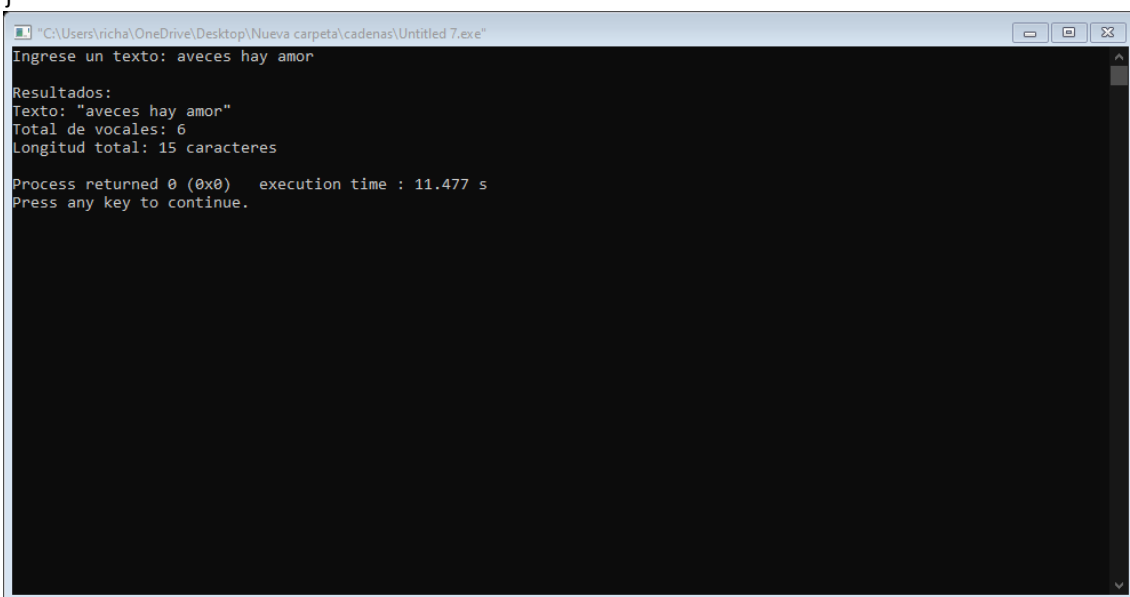
// Contar vocales
for (i = 0; texto[i] != '\0'; i++) {
    char letra = texto[i];

    if (letra == 'a' || letra == 'A' ||
        letra == 'e' || letra == 'E' ||
        letra == 'i' || letra == 'I' ||
        letra == 'o' || letra == 'O' ||
        letra == 'u' || letra == 'U') {
        vocales++;
    }
}

printf("\nResultados:\n");
printf("Texto: \"%s\"\n", texto);
printf("Total de vocales: %d\n", vocales);
printf("Longitud total: %d caracteres\n", (int)strlen(texto));

return 0;
}

```



```

C:\Users\richa\OneDrive\Desktop\Nueva carpeta\cadenas\Untitled 7.exe
Ingrese un texto: aveces hay amor

Resultados:
Texto: "aveces hay amor"
Total de vocales: 6
Longitud total: 15 caracteres

Process returned 0 (0x0)   execution time : 11.477 s
Press any key to continue.

```

<https://onlinegdb.com/oxTZdPM3b>