Laboratorio 4

Programación de Computadores (2025-2)

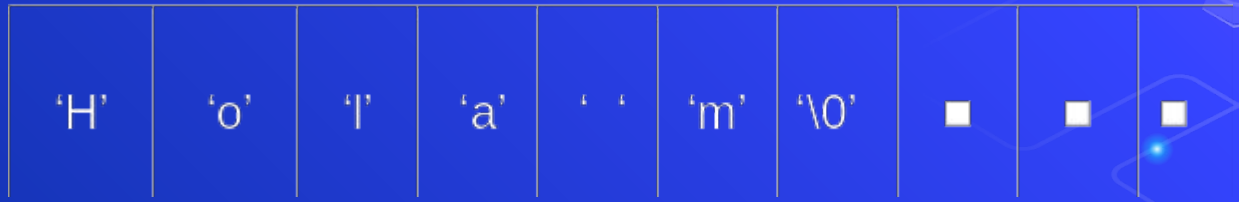
Temas del Laboratorio 4

- Cadenas de caracteres
- Uso de ***string.h***
- Otra funciones para strings
- Ejercicios

Strings en C

Un string en C es un arreglo de chars que contiene el caracter nulo ‘\0’, el cual indica el término del string.

char mi_string[10] = “Hola m”



Algunas funciones de string.h

Función	Significado
`strlen(s)`	Obtiene la cantidad de caracteres del string
`strcat(s1,s2)`	Concatena el string s2 al final del string s1
`strncat(s1,s2,n)`	Concatena los primeros “n” caracteres de s2 a s1
`strcmp(s1,s2)`	Compara si los string s1 y s2 son iguales
`strncmp(s1,s2,n)`	Compara los primeros “n” caracteres de s1 y s2
`strcpy(destino,origen)`	Copia el string “origen” en “destino”
`strncpy(destino,origen,n)`	Copia los primeros “n” caracteres de “origen” en “destino”
`strtok(s,c)`	Divide la cadena “s” según el delimitador “c”

strlen(s)

```c
char texto[20] = "Programa";

printf("%ld\n", strlen(texto));
```

Obtiene la cantidad de caracteres del string.

Esta función omite el carácter de término **‘/0’**

strcat(s1,s2)

```
char s1[30] = "Programar";
char s2[20] = " es divertido";

strcat(s1, s2);

printf("%s\n", s1);
```

El arreglo de destino debe tener el espacio necesario para recibir el string a concatenar

Esta función elimina el carácter **‘/0’** del string de destino.

strncat(s1,s2,n)

```
char s1[30] = "Gatitos";
char s2[20] = "Perritos";

strncat(s1, s2, 5);
printf("%s\n", s1);
```

Se define la cantidad de caracteres “n” a concatenar

strcmp(s1,s2)

```c
char s1[30] = "Hola";
char s2[20] = "Mundo";

if(strcmp(s1, s2)==0){
    printf("Las cadenas son iguales\n");
}
else{
    printf("Las cadenas son diferentes\n");
}
```

Devuelve el valor **0** si las cadenas de caracteres son iguales

strncmp(s1,s2,n)

```c
char s1[30] = "Cats";
char s2[20] = "Catapult";

if(strncmp(s1, s2, 3) == 0){
    printf("Son iguales\n");
}else{
    printf("Son distintas\n");
}
```

Devuelve el valor **0** si los primeros "**n**" caracteres de ambas cadenas de texto son iguales

strcpy(s2,s1)

```
char s1[30] = "Gatito";
char s2[20];

strcpy(s2, s1);
printf("%s\n", s2);
```

El arreglo de destino (s2) debe tener el espacio necesario para recibir la cadena s1

strncpy(s2,s1,n)

```
char s1[30] = "Paralelepipedo";
char s2[30];

strncpy(s2, s1, 4);
printf("%s\n", s2);
```

Copia los primeros “**n**” caracteres

strtok(s,c)

```c
char frase[500];
char *token;

printf("Ingrese una frase:\n");
scanf("%[^\n]s", frase);

// Primera llamada a strtok
token = strtok(frase, " ");

// Recorre y muestra los tokens
while (token != NULL) {
    printf("Token: %s\n", token);
    // Llamadas sucesivas para obtener tokens siguientes:
    token = strtok(NULL, " ");
}
```

Se utiliza para dividir (o tokenizar) una cadena en partes más pequeñas o tokens, basándose en un delimitador específico.

La función **strtok** se llama primero con la cadena original (***frase***) y el delimitador (**" "**). Luego, se llama sucesivamente con **NULL** para obtener tokens adicionales.

```c
scanf("%[^\n]s", frase);
```

El modificador **“%[^\n]s”** permite capturar todos los caracteres excepto el salto de línea.

Como se guarda la entrada en un arreglo de chars no se debe colocar el “**&**” antes del nombre del arreglo.

Otras funciones para usar con strings

```c
#include <stdlib.h>

char s[] = "102.3";
int n1 = atoi(s);          // Cadena de caracteres a int
long n2 = atol(s);         // Cadena de caracteres a long
long long n3 = atoll(s); // Cadena de caracteres a long long
float n4 = atof(s);        // Cadena de caracteres a double
```

Conversión de strings a números con **stdlib.h**

```c
#include <ctype.h>

char c = '8';
isdigit(c);    // Chequea si c es un dígito
isalpha(c);    // Chequea si c es una letra
islower(c);    // Chequea si c es una letra minúscula
isupper(c);    // Chequea si c es una letra mayúscula
tolower(c);    // Convierte letras mayúsculas a minúsculas
toupper(c);    // Convierte letras minúsculas a mayúsculas
```

Verificación de un carácter con **ctype.h**

Ejercicio 1

Escriba un programa que reciba una frase ingresada por el usuario y que cambie las letras de minúsculas a mayúsculas y viceversa. (Se debe verificar que los caracteres ingresados sean letras)

Input:

HOLa, Esta ES Una fRAse

Output:

holA, eSTA es uNA FraSE

```
Escribe una frase:
HOLa, Esta ES Una fRAse
La frase convertida es:
holA, eSTA es uNA FraSE
```

Ejercicio 2

Escriba un programa que pida:

- Ingresar una frase (**f**)
- Ingresar una palabra (**p**)
- Ingresar un número entero (**n**) que debe ser menor o igual a la cantidad de caracteres de la palabra **p**.

Luego el programa debe:

- Imprimir la palabra más corta y más larga de la frase **f**
- Indicar si hay una coincidencia de los primeros **n** caracteres
- de la palabra **p** en alguna palabra de la frase **f**

Ejemplo del Ejercicio 2

Input:

```
Ingrese una frase: Esta es una frase de ejemplo para el ejercicio

Ingrese una palabra: ejem

Ingrese un número entero: 4
```

Output:

```
La palabra más corta es: es

La palabra más larga es: ejercicio

Existe coincidencia de los primeros 4 caracteres de la palabra
'ejem' en la frase.
```