

# Strings en C

Mg. Ing. Facundo S. Larosa

## Informática I

Instituto Universitario Aeronáutico  
Universidad de la Defensa Nacional (UNDEF)

# Ejemplo disparador

Diseñe un programa en el cual se ingresan los datos de un curso de no más de 100 alumnos, incluyendo sus nombres y apellidos, numero de legajo y promedio general. A continuación, imprima en orden de mérito los 10 mejores promedios.

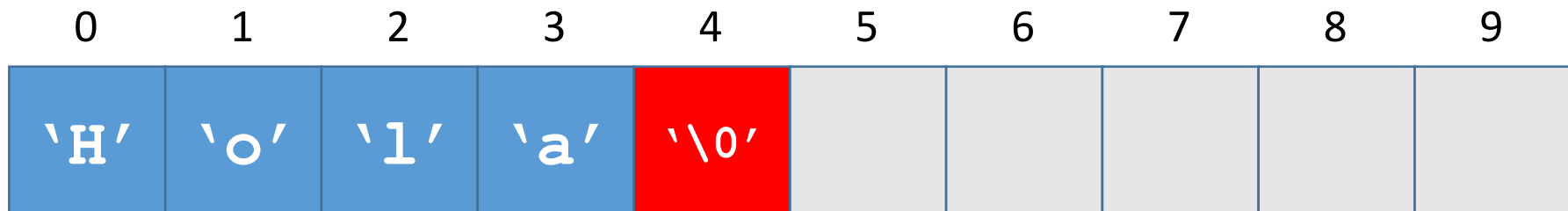
¿Cómo hacemos?

# String: Definición

Un string o cadena de caracteres es una secuencia arbitraria de caracteres terminados por un carácter especial (denominado carácter nulo o `'\0'`).

`//Declaración de un string`

`char texto[10]="Hola";`



# Primeros programas con strings

```
//El programa imprime "Hola mundo"
```

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

```
int main (void)
```

```
{
```

```
    char texto[15]="Hola mundo";
```

```
    printf("El texto es: %s", texto);
```

```
    return 0;
```

```
}
```

```
//Vamos a investigar un poco que hay dentro del vector 'texto'
```

# Primeros programas con strings

//Leemos e imprimimos un string

```
#include <stdio.h>
#define N 50
int main (void)
{
    char texto[N];

    printf("\nIngrese un corto texto: ");

    scanf ("%s", texto);

    printf("\n\nUsted ingreso el texto: %s", texto);

    return 0;
}
```

# Primeros programas con strings

//Leemos e imprimimos un string (variante)

```
#include <stdio.h>
#define N 50
int main (void)
{
    char texto[N];

    printf("\nIngrese un corto texto: ");

    gets(texto);

    printf("\n\nUsted ingreso el texto: ");
    puts(texto);

    return 0;
}
```

# Funciones estándar de strings: Librería string.h

La librería estándar “string.h” nos provee de una variedad de funciones que facilitan el uso de strings en nuestro programa. Nosotros vamos a ver las funciones de uso más frecuente:

- **strlen** : Devuelve el largo de un string
- **strcpy**: Permite copiar un string
- **strcmp**: Permite comparar dos strings
- **strcat**: Permite concatenar (unir) dos strings

# strlen (Devuelve el tamaño de un string)

La función strlen posee el siguiente prototipo:

```
size_t strlen (const char *s);
```

Recibe:

- s: Puntero al string

Devuelve:

- El número de caracteres (bytes) del string sin contar el elemento nulo

**A programar:** Realizar un breve programa que permita cargar un string e indique su tamaño



# strcpy (Copia un string)

La función strcpy posee el siguiente prototipo:

```
char * strcpy (char *dest, const char *src);
```

Recibe:

- dest: Puntero destino (adonde se quiere copiar)
- Src: Puntero origen (desde donde se quiere copiar)

Devuelve:

- Puntero al string de destino (no se suele utilizar)

# strcpy: Ejemplo

//Ejemplo de strcpy

```
#include <stdio.h>
#include <string.h>
#define N 20
int main (void)
{
    char texto1[N]="Hola", texto2[N] ;

    //Copio el contenido de texto1 en texto2
    strcpy(texto2, texto1) ;

    printf("\n\ntexto1: %s\ntexto2: %s\n", texto1, texto2) ;

    return 0;
}
```

# strcpy: ¡Cuidado!

//Ejemplo de strcpy

```
#include <stdio.h>
#include <string.h>
#define N 20
int main (void)
{
    char texto1[N]="Hola", texto2[N] ;

    //Es un error muy común hacer esto... pero no funciona!!!
    texto2=texto1;

    printf("\n\ntexto1: %s\ntexto2: %s\n", texto1, texto2) ;

    return 0;
}
```

# strcmp (Compara dos strings)

La función strcmp posee el siguiente prototipo:

```
int strcmp (const char * s1, const char * s2);
```

Recibe:

- s1: Puntero a string 1
- s2: Puntero a string 2

Devuelve:

- 0 : Si los strings son iguales
- <0: Si s1 es menor que s2 (alfabéticamente s1 está antes que s2)
- >0: Si s1 es mayor que s2 (alfabéticamente s1 está después que s2)

## strcmp: Ejercicio

Realizar un breve programa en el que se ingresen dos strings por teclado. El programa luego indica si:

- Los strings son iguales
- El string 1 es menor que el string 2
- El string 1 es mayor que el string 2

# strcat (Une dos strings)

La función strcat posee el siguiente prototipo:

```
char * strcat (char * dest, const char * src);
```

Recibe:

- src: Puntero a concatenar (segundo en la secuencia)
- dest: Puntero a concatenar (primero en la secuencia)

Devuelve:

- Puntero a string de destino (no se suele utilizar)

La función guarda en el string “dest” la unión de “dest+src”

# strcat: Ejemplo

//Ejemplo de strcat

```
#include <stdio.h>
#include <string.h>
#define N 20
int main (void)
{
    char texto1[N]="Hola", texto2[N]="Chau";

    //Uno los strings y guardo el resultado en texto1

    strcat(texto1, texto2);

    printf("\n\ntexto1: %s\ntexto2: %s\n", texto1, texto2);

    return 0;
}
```

# strcat: Ejemplo

//Ejemplo de strcat

```
#include <stdio.h>
#include <string.h>
#define N 20
int main (void)
{
    char texto1[N]="Hola", texto2[N]="Chau";

    //Uno los strings y guardo el resultado en texto1

    strcat(texto1, texto2);

    printf("\n\ntexto1: %s\ntexto2: %s\n", texto1, texto2);

    return 0;
}
```



# “Vectores” de strings

En este punto, sería útil poder almacenar un conjunto de strings de igual forma que por medio de un vector de ints almacenábamos un conjunto de ints...

Pero los strings de por si se almacenan en vectores!

A continuación veremos como...

# “Vectores” de strings

```
#include <stdio.h>
#include <string.h>
#define M 5
#define N 20
int main (void)
{
    char texto[M][N];
    int i;

    for (i=0;i<M;i++)
    {
        printf("\n\nIngresa un string: ");
        gets(texto[i]);
    }

    for(i=0;i<M;i++)    printf("\n\nEl string %d es: %s",i,texto[i]);
}
```

# “Vectores” de strings

La definición : **char** texto[M] [N]

Implica una matriz de  $M \times N$  variables char

Con  $M=5$  y  $N=20$ :

**Cada elemento es una variable char**

## 20 columnas

## 5 filas

[illegible]

# “Vectores” de strings: Notación

```
char texto[M][N]
```

texto[1][14]

[illegible]



# Ejercicios

- 1) Realizar un programa en el que se ingresan los nombres de 10 alumnos. A continuación, se pide ingresar un nombre. Si este último está en la lista, el programa indica en que posición se ingresó o su inexistencia.
- 2) Realizar un programa en el que se ingresan los nombres de 10 alumnos. A continuación, los nombres se imprimen ordenados alfabéticamente.

# Ejercicios

3) Realizar un programa en el que se ingresan los nombres de 30 alumnos con sus promedios anuales. El programa imprime la lista de alumnos en orden de mérito, separando los alumnos en grupos de notas:  $(0,4)$  ,  $[4,7)$  y  $[7,10]$ .

4) Para un campeonato de dobles de tenis se ingresan los nombres de sus participantes, junto con su puntuación en el torneo anterior. El programa imprime las parejas, agrupando a los jugadores de mayor y menor puntaje.