Laboratorio I

Strings ó
Cadenas de Caracteres

FECyT
Facultad de Ciencias
Exactas y Tecnológicas

UNSE
Universidad Nacional
de Santiago del Estero

Definiciones

- El lenguaje **C** no proporciona un tipo de dato **string** como lo hacen otros lenguajes.
- Por ello, se utiliza un **array** de elementos de tipo **char** para almacenar un **string**.
- Un **string o cadena de caracteres** es una secuencia de caracteres o símbolos encerrados entre comillas.

Declaración

- La sintaxis para la declaración de variables string es la siguiente:

```
char nombre[cantidad de caracteres];
```

En donde la cantidad de caracteres debe ser uno mas que la cantidad de caracteres que se desean almacenar.

Ejemplos:

```
char apellido[10];
char apellido[ ] = "SAYAGO";
char apellido[10] = "PEREZ";
```


Representación en Memoria

- La variable apellido[10]="PEREZ" se almacena en memoria de la siguiente manera:

- Si observa, un carácter con el valor cero se agrega automáticamente al final de cada arreglo para indicar su final. Dicho carácter es conocido como **carácter nulo** y se escribe como **\0.**(Esta es la causa de porque debe asignarse un espacio mas en el arreglo)

Entrada / Salida

- La especificación **%s** es para trabajar con los strings terminados en el carácter nulo, y puede ser utilizado tanto con **printf** como con **scanf**, pero este ultimo con la limitación de que su final viene dado por la existencia de un espacio o de la tecla enter.
- En ambas operaciones, ya que el string esta almacenado en un arreglo de char, solo debe utilizar el nombre del arreglo.
- Además, debido a que es un arreglo, este se comporta como tal y puede ser trabajado como tal, teniendo en consideración que el carácter \0 indicara el final del string.

Entrada / Salida

Ejemplos:

```
char apellido[20];
printf("Ingrese el apellido:");
scanf("%s", apellido);
printf("El apellido ingresado fue %s\n",apellido);
char letra;
int i = 0;
while (apellido[i]!='\0'){
    printf("%c\n",apellido[i]);
    i++;
}
```

Operaciones con String

- Debido a que el string es almacenado como un arreglo que termina en \0,, dentro de la librería estándar existe la **<string.h>** la cual una que contiene funciones especificas para el manejo de strings. Algunas de ellas son:

gets	**strlen**
strcpy	**strcmp**
strcat	**strchr**
strstr	

Función gets

- Función: **gets**
- Sintaxis: **gets(string)**;
- Descripción: permite ingresar un string desde el teclado.
- Nota: la ventaja con respecto a scanf es que puede ingresar incluso caracteres en blanco y el final viene dado por la tecla Enter.

Ejemplo:

```
char apellido[20];
printf("Ingrese el apellido:");
gets(apellido);
printf("El apellido ingresado fue %s\n",apellido);
```

Función strlen

- Función: **strlen**
- Sintaxis: **strlen(string)**
- Descripción: retorna la longitud de un string excluyendo el \0.

Ejemplo:

```
char apellido[20];
printf("Ingrese el apellido:");
gets(apellido);
printf("El apellido ingresado fue %s",apellido);
printf("  y tiene %c caracteres\n",strlen(apellido));
```


Función strcpy

- Función: **strcpy**
- Sintaxis: **strcpy(string1,string2)**
- Descripción: copia el string2 en el string1 reemplazando el que previamente existía en sting1
- Nota: es responsabilidad de quien lo usa garantizar de que el string2 cabe en el string1.

Ejemplo:

```
char provincia[20]="Tucuman";
printf("El valor de provincia es %s\n",provincia);
strcpy(provincia,"Santiago";
printf("Ahora el valor de provincia es %s\n",provincia);
```

Función strcat

- Función: **strcat**
- Sintaxis: **strcat(string1,string2)**
- Descripción: copia el string2 al final del string1.
- Nota: es responsabilidad de quien lo usa garantizar de que el string1 tiene el suficiente espacio para albergar el string resultante.

Ejemplo:

```
char oceano[20]="Oceano";
printf("El valor de oceano es %s\n",oceano);
strcat(oceano," Atlantico";
printf("Ahora el valor de oceano es %s\n",oceano);
```


Función strcmp

- Función: **strcmp**
- Sintaxis: **strcmp(string1,string2)**
- Descripción: compara dos string, retornando un valor entero que es menor que 0, igual a cero o mayor que cero, los cuales corresponden a str1<str2, str1==str2 y str1>str2 respectivamente.

Ejemplo:

```
char nombre1[20]="Ana";
char nombre2[20]="Ana";
if (strcmp(nombre1,nombre2)==0)
      printf("Ambos nombres son iguales\n");
else
      printf("Los nombres son distintos\n");
```

Función strcmp

Ejemplo:

```
char nombre1[20]="Analia";
char nombre2[20]="Ana";
if (strcmp(nombre1,nombre2)>0)
      printf("El primer nombre es mas largo\n");
else
      if (strcmp(nombre1,nombre2)<0)
         printf("El segundo nombre es mas largo\n");
```


Funciones para el Manejo de Caracteres <ctype.h>

islower()	es una letra minúscula
isupper()	es una letra mayúscula
isdigit()	es un digito
isalpha()	es un carácter alfanumérico
isblank()	es un espacio en blanco
isspace()	es un espacio en blanco('\t')
toupper()	convierte de minúscula a mayúscula
tolower()	convierte de mayúsculas a minúsculas.