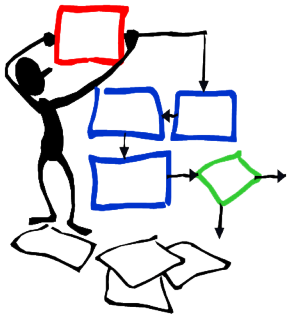
Instituto Politécnico Nacional

Escuela Superior de Cómputo



Algoritmia y programación estructurada

Clase 24: Manejo de cadenas en C

Prof. Edgardo Adrián Franco Martínez
http://computacion.cs.cinvestav.mx/~efranco
@efranco_escom
efranco.docencia@gmail.com

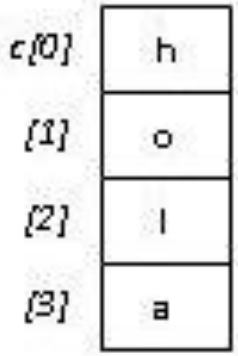
Contenido

- Cadenas en C
- Manejo de cadenas en C <string.h>
- Prueba y manejo de caracteres en C <ctype.h.h>

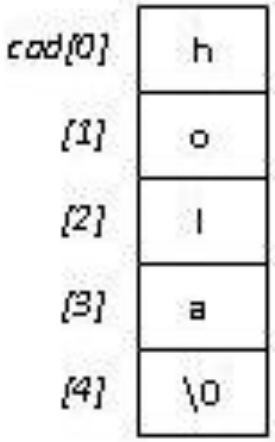
Cadenas en C

- A diferencia de otros lenguajes de programación que emplean un tipo denominado cadena **string** para manipular un conjunto de símbolos, en C, se debe simular mediante un arreglo de caracteres, en donde la terminación de la cadena se debe indicar con **nulo con valor 0**.

char c [4];



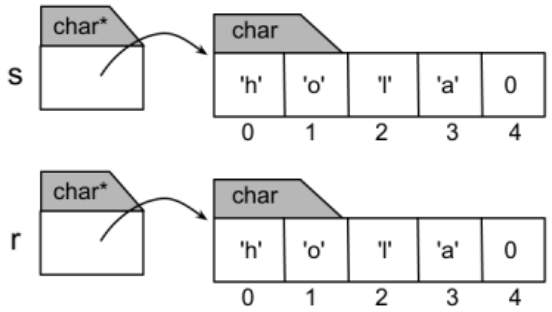
char cad[5];



- Un nulo se especifica como '\0'. Por lo anterior, cuando se declare un arreglo de caracteres para ser usado como cadena, se debe considerar un carácter adicional a la cadena más larga que se vaya a guardar.

- Por ejemplo, si se quiere declarar un arreglo cadena que guarde una cadena de diez caracteres, se hará como:

char cadena[11];



- Se pueden hacer también inicializaciones de arreglos de caracteres en donde automáticamente C asigna el carácter nulo al final de la cadena, de la siguiente forma:

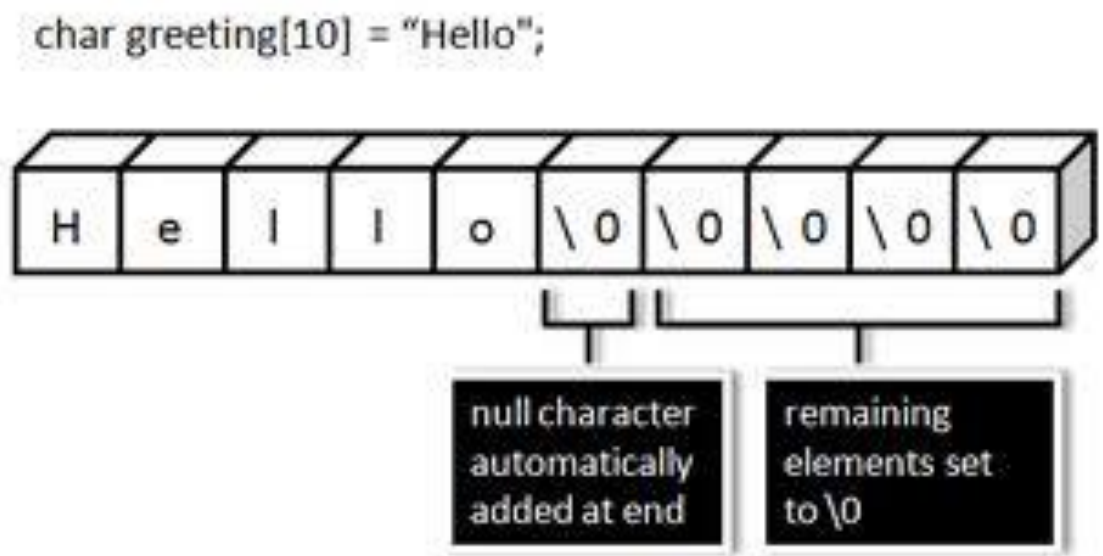
char nombre_arr[tam]="cadena";

- Por ejemplo, el siguiente fragmento inicializa cadena con ``hola'':

char cadena[5]="hola";

  - El código anterior es equivalente a:

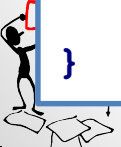
char cadena[5]={'h','o','l','a','\0'};



- Para asignar la entrada estándar a una cadena se puede usar la función **scanf** con la opción **%s** (cadenas sin espacios) **%[^\n]** (cadenas con espacios hasta el fin de línea) *(observar que no se requiere usar el operador & en el scanf paras el argumento, ya que el nombre del arreglo es un apuntador estático al arreglo),* de igual forma para mostrarlo en la salida estándar.

```
#include <stdio.h>
int main(void)
{
    char nombres[15], apellido[30];

    printf("Introduce tu nombre(s): ");
    scanf("%[^\n]",nombres); //Leer caracteres hasta fin de línea
    printf("Introduce tu apellido: ");
    scanf("%s",apellido);
    printf("Usted es %s %s\n",nombres,apellido); //Leer caracteres
hasta el primer espacio
    return 0;
}
```



- El %s puede delimitar los caracteres a considerar como máximos a tomar para almacenar en una cadena. %24s indica que 24 es la máxima cantidad de caracteres que esperemos se ingresen (**uno menos que el tamaño del arreglo**). Si se intenta ingresar más de 24 sencillamente no los almacenará. Esta es una buena costumbre cada vez que se utiliza scanf() para capturar cadenas.

```c
#include <stdio.h>
int main(void)
{
        char a[25];
        printf("Ingresa tu nombre: ");
        scanf("%24s",a); // 24 caracteres a tomar
        printf("Tu nombre es: %s\n",a);
    return 0;
}
```

- El lenguaje C no maneja cadenas de caracteres, como se hace con enteros o flotantes, por lo que lo siguiente no es válido:

```c
#include <stdio.h>
int main(void)
{
    char nombre[40], apellidos[40], completo[80];

    nombre="José María";     // Asignaciones ilegales
    apellidos="Morelos y Pavón";
    completo="Gral."+ nombre + apellidos; // Operación Ilegal

    return 0;
}
```

Manejo de cadenas en C <string.h>

- **string.h** es un archivo de la Biblioteca estándar que contiene la definición de macros, constantes, funciones y tipos de utilidad para trabajar con cadenas de caracteres.

- Las funciones declaradas en string.h funcionan en cualquier plataforma que soporte ANSI C. Sin embargo, existen algunos problemas de seguridad con estas funciones, como el desbordamiento de arreglos, que hacen que algunos programadores prefieran opciones más seguras frente a la portabilidad que estas funciones ofrecen. Además, las funciones para cadenas de caracteres sólo trabajan con conjuntos de caracteres ASCII o extensiones ASCII compatibles.

- Para poder utilizar las funciones del manejo de cadenas, es necesario incluir la biblioteca:

#include <string.h>

- Además, es importante preservar el carácter de terminación NULL ‘\0’, ya que con éste es como C define y maneja las longitudes de las cadenas.

- Todas las funciones de la biblioteca estándar para el manejo de cadenas de C lo requieren para una operación satisfactoria.

Funciones para manipulación de arreglos de caracteres (Manipulación de cadenas)

- Las funciones de uso más común de dicho archivo de cabecera son:

Nombre de la función	Descripción
strcat()	`char* strcat (char* destino, const char* fuente)` Añade la cadena fuente al final de la destino. Devuelve la cadena destino.
strlen()	`size_t strlen( const char* s )` Devuelve la longitud de la cadena `s`
strncpy()	`char* strncpy( char* dest, const char* fuente, size_t n)` Copia n caracteres de la cadena fuente a la cadena destino

Nombre de la función	Descripción
strncat()	`char* strncat( char* s1, const char* s2, size_t n)` Añade los primeros n caracteres de S2 a S1. Devuelve s1 si `n>=strlen(s2)`, entonces `strncat` tiene el mismo efecto que `strcat`.
strchr()	`char* strchr( const char* s1, int ch)` Devuelve un apuntador a la primera ocurrencia de ch en s1, devuelve NULL si ch no se encuentra en s1.
strcmp()	`int strcmp( const char* s1, const char* s2)` Compara alfabeticamente la cadena s1 con s2 y devuelve un dato de: cero si s1 == s2 menor que cero si s1 < s2 mayor que cero si s1 > s2
strcspn()	`size_t strcspn( const char* s1, const char* s2)` Devuelve la longitud de la subcadena más larga de `s1` que comienza con el carácter `s1[0]` y no contiene ninguno de los caracteres de la cadena `s2`.

Consultar las funciones de la librería "string.h" de ANSI C

Prueba y manejo de caracteres en C <ctype.h.h>

- Una biblioteca relacionada #include <ctype.h> la cual contiene muchas funciones útiles para convertir y probar caracteres individuales.

- Las funciones más comunes para revisar caracteres tienen los siguientes prototipos:
  - ***int isalnum(int c):*** *Verdad si c es alfanumérico.*
  - ***int isalpha(int c):*** *Verdad si c es una letra.*
  - ***int isascii(int c):*** *Verdad si c es ASCII.*
  - ***int iscntrl(int c):*** *Verdad si c es un caracter de control.*
  - ***int isdigit(int c):*** *Verdad si c es un dígito decimal.*
  - ***int isgraph(int c):*** *Verdad si c es un caracter imprimible, exceptuando el espacio en blanco.*

- ***int islower(int c):*** *Verdad si c es una letra minúscula.*
- ***int isprint(int c):*** *Verdad si c es un caracter imprimible, incluyendo el espacio en blanco.*
- ***int ispunct(int c):*** *Verdad si c es un signo de puntuación.*
- ***int isspace(int c):*** *Verdad si c es un espacio*
- ***int isupper(int c):*** *Verdad si c es una letra mayúscula.*
- ***int isxdigit(int c):*** *Verdad si c es un dígito hexadecimal.*

- Las funciones para conversión de caracteres son:
  - ***int toascii(int c):*** *Convierte c a ASCII o un unsigned char de 7 bits, borrando los bits altos.*
  - ***int tolower(int c):*** *Convierte la letra c a minúsculas, si es posible.*
  - ***int toupper(int c):*** *Convierte la letra c a mayúsculas, si es posible.*