

Clase 07: String

- Funciones para manejo de cadenas y caracteres
- Ejercicios

Funciones para manejo de cadenas y caracteres

Se encuentran en las librerías *string.h* (tabla 5.2), *ctype.h* (tabla 5.3) y *stdlib.h* (tabla 5.4). Varias de las macros y funciones declaradas en `<ctype>` realizan pruebas y conversiones de caracteres. En la tabla 5.3, las macros regresan un valor *int* y las funciones un valor *char*.

Tabla 5.2 Librería *string.h*

Nombre	Descripción
<i>strcat</i> (<i>cad1</i> , <i>cad2</i>)	Concatena (agrega) <i>cad2</i> al final de <i>cad1</i> (incluyendo el carácter nulo); el carácter inicial de <i>cad2</i> sobrescribe el carácter nulo al final de <i>cad1</i> . Devuelve el valor de <i>cad1</i> .
<i>strcmp</i> (<i>cad1</i> , <i>cad2</i>)	Compara los elementos de dos cadenas <i>cad1</i> y <i>cad2</i> . Si son iguales, devuelve 0. Si el de <i>cad1</i> es mayor que <i>cad2</i> , devuelve un valor mayor que cero; en caso contrario, devuelve un valor menor que cero.
<i>strcpy</i> (<i>cad1</i> , <i>cad2</i>)	Copia la cadena <i>cad2</i> , incluyendo el nulo, en el arreglo <i>cad1</i> . Devuelve el valor de <i>cad1</i> .
<i>strlen</i> (<i>cad</i>)	Devuelve el número de caracteres de la cadena <i>cad</i> , sin incluir el nulo de terminación.
<i>strncat</i> (<i>cad1</i> , <i>cad2</i> , <i>n</i>)	Concatena <i>n</i> caracteres de <i>cad2</i> a <i>cad1</i> . Devuelve el valor de <i>cad1</i> .
<i>strncmp</i> (<i>cad1</i> , <i>cad2</i> , <i>n</i>)	Es igual que <i>strcmp</i> pero con <i>n</i> caracteres, compara los <i>n</i> primeros caracteres de una cadena.
<i>strncpy</i> (<i>cad1</i> , <i>cad2</i> , <i>n</i>)	Copia los primeros <i>n</i> caracteres de <i>cad2</i> a <i>cad1</i> . Devuelve el valor de <i>cad1</i> .
<i>strchr</i> (<i>cad</i> , <i>c</i>)	Localiza la primera ocurrencia del carácter <i>c</i> en la cadena <i>cad</i> ; si no se encuentra, devuelve un puntero nulo.
<i>strstr</i> (<i>cad1</i> , <i>cad2</i>)	Busca en <i>cad1</i> la subcadena <i>cad2</i> , si la encuentra devuelve un puntero hacia la primera ocurrencia de <i>cad2</i> en <i>cad1</i> , si no regresa <i>null</i> .
<i>strupr</i> (<i>cad</i>)	Convierte a mayúsculas la cadena <i>cad</i> .
<i>strlwr</i> (<i>cad</i>)	Convierte a minúsculas la cadena <i>cad</i> .

Tabla 5.3 Librería *ctype.h*

Nombre	Descripción	Ejemplo	Valor que regresa
Macros			
<i>isalpha(c)</i>	Comprueba si un carácter es <i>alfabético</i>	isalpha('a') isalpha('7') isalpha('A')	2 0 1
<i>isdigit(c)</i>	Comprueba si un carácter es un <i>dígito decimal</i> .	isdigit('a') isdigit('7')	4 0
<i>islower(c)</i>	Comprueba si un carácter es de tipo <i>minúscula</i> . Devuelve un valor distinto de cero si <i>c</i> es cualquiera de las letras minúsculas [a-z]	islower('a') islower('Z') islower('7')	2 0 0
<i>ispunct(c)</i>	Comprueba si un carácter es <i>signo de puntuación</i> .	ispunct('.') ispunct('?')	16 16
<i>isspace(c)</i>	Comprueba si un carácter es un <i>espacio</i>	isspace(' ')	8
<i>isupper(c)</i>	Comprueba si un carácter es de tipo <i>mayúscula</i> . Devuelve un valor distinto de cero si <i>c</i> es cualquiera de las letras mayúsculas [A-Z].	isupper('a') isupper('Z')	0 1
<i>toascii(c)</i>	Convierte <i>c</i> a su correspondiente código ASCII; regresa un valor dentro del rango entre 0 y 127.	toascii('a') toascii('A')	97 65
Funciones			
<i>tolower(c)</i>	Devuelve la correspondiente letra minúscula si existe y si <i>isupper(c)</i> es distinto de cero; en caso contrario, devuelve <i>c</i> .	tolower('A') tolower('7')	a 7
<i>toupper(c)</i>	Devuelve la correspondiente letra mayúscula si existe y si <i>islower(c)</i> es distinto de cero; en caso contrario, devuelve <i>c</i> .	toupper('a') toupper('7')	A 7

Tabla 5.4 Funciones para la conversión de tipos, librería *stdlib.h*

<i>Nombre</i>	<i>Descripción</i>	<i>Tipo de valor de regreso</i>	<i>Ejemplo</i>	<i>Valor que regresa</i>
atof(cad)	Convierte los caracteres de la cadena <i>cad</i> a la representación interna de tipo <i>double</i> y devuelve ese valor.	double	atof("25.5")	25.500000
atoi(cad)	Convierte los caracteres de la cadena <i>cad</i> a la representación interna de tipo <i>int</i> y devuelve ese valor.	int	atoi("25.5")	25

Nota: En las funciones *atof* y *atoi*, si el primer carácter de la cadena no es un número devuelven 0.

Ejercicio 1

Ejercicio 2. Leer el Password, si es correcto imprimir "Correcto, puedes continuar" y si no imprimir "Password equivocado".

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

main()
{
    char cad[80];
    puts("Dame tu Password :"); gets(cad);
    if((strcmp(cad, "acceso1234")) == 0)
        puts("Correcto, puedes continuar");
    else puts("Password equivocado");
    getch(); return 0;
}
```

Una posible salida en pantalla,
después de ejecutar el programa
sería:

Dame tu Password :

acceso1234

Correcto, puedes continuar

Ejercicio 2

Ejercicio 3. Generar el código de César. El cifrado César mueve cada letra un determinado número de espacios en el alfabeto.

```
#include <ctype.h>
#include <string.h>
#include <conio.h>
#include <stdio.h>
main()
{
    char cadena[80], codigo[80], alfab[27];
    int i,j, espacio;
    for (i=0,j= 65;i<26;i++,j++)
        alfab[i] = j;
    printf("Dame el Texto: "); gets(cadena);
    printf("Dame el número de espacios: ");
    scanf("%d",&espacio);
    for (i=0;i<strlen(cadena);i++)
    {
        cadena[i]=toupper(cadena[i]);
        for (j=0;j<26;j++)
            if (cadena[i]==alfab[j]) break;
        j=j+espacio;
        if (j>25) j=j-26;
        codigo[i]=alfab[j];
    }
    codigo[strlen(cadena)]= '\0';
    puts (codigo); getch(); return 0;
}
```

Una posible salida en pantalla,
después de ejecutar el programa
sería:

Dame el Texto: ABCD

Dame el número de espacios: 2
CDEF

Ejercicio 3

Ejercicio 4. Leer una cadena e imprimir el total de: consonantes, vocales, dígitos, caracteres especiales, mayúsculas, minúsculas, espacios en blanco y caracteres de la cadena.

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

main ()
{
    char cadena[26];
    int consonan=0,vocales=0,digitos=0,punt=0,may=0,min=0,espacios=0,i, total;
    printf ("Introduce una oración (máx. 25 caracteres): ");
    gets(cadena); total=strlen(cadena);
    for(i=0;i<total;i++)
    {
        if (isdigit(cadena[i])) digitos++; // Cuenta dígitos
        if (ispunct(cadena[i])) punt++; // Cuenta signos de puntuación
        if (isspace(cadena[i])) espacios++; // Cuenta espacios en blanco
        if (isalpha(cadena[i])) // Letras
        {
            if(isupper(cadena[i])) may++; // Cuenta mayúsculas
            else min++; // Cuenta minúsculas
            cadena[i]=tolower(cadena[i]); //Pasamos letras a minúsculas para contarlas
        }
    }
}
```

```

switch(cadena[i])
{
    case ('a'):case ('e'):case('i'):case('o'):case('u'): vocales++;// cuenta vocales
    break;
    default:consonan++;
}
}
}
printf ("En la frase hay:\n");
printf ("Consonantes:%d\n", consonan);
printf ("Vocales :%d\n" , vocales);
printf ("Dígitos :%d\n" , digitos);
printf ("Signos de puntuación:%d\n ",punt);
printf ("Mayúsculas :%d \n",may);
printf ("Minúsculas :%d \n",min);
printf ("Espacios en blanco:%d\n",espacios);
printf ("Total:%d\n",total);getch();return 0;
}

```

Una posible salida en pantalla después de ejecutar el programa, sería:

Introduce una oración (max. 25 caracteres):

Revolución # 1500

En la frase hay;

Consonantes: 5

Vocales: 4

Dígitos: 4

Signos de puntuación: 1 //El carácter #

Mayúsculas: 1

Minúsculas: 9

Espacios en blanco: 2

Total: 17

Ejercicio 4

Ejercicio 5. Comprobar si un texto introducido por teclado es o no un palíndromo.

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <ctype.h>
main()
{
    char cad1[80], cad2[80];
    int i, c=0;
    printf("Dame una Cadena: "); gets(cad1);strupr(cad1);
    for(i=0; i<strlen(cad1); i++)
        if(!isspace(cad1[i]))
        {
            cad2[c]=cad1[i]; c++;
        }
    cad2[c]='\0';
    strcpy(cad1, cad2);
    //invertimos cadena
    for(i=0; i<strlen(cad1); i++)
    {
        c--; cad2[i]=cad1[c];
    }
    if (strcmp(cad1, cad2)==0) printf("Es palíndroma");
    else printf("No es palíndroma");
    getch(); return 0;
}
```

Una posible salida en pantalla después de ejecutar el programa sería:

Dame una Cadena: anita lava la tina

Es palíndroma

Ejercicio 5

Ejercicio 6. Leer un número arábigo e imprimir su correspondiente romano.

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

<i>i</i>	0	1	2	3	4	5	6	7	8	9	10	11	12
<i>romano</i>	M	CM	D	CD	C	XC	L	XL	X	IX	V	IV	I
<i>arabigo</i>	1000	900	500	400	100	90	50	40	10	9	5	4	1

```
#include<string.h>
main ()
{
int num,i, arabigo[13]={1000,900,500,400,100,90,50,40,10,9,5,4,1};
char romano[13][3]={ "M","CM","D","CD","C","XC","L","XL","X","IX","V","IV",
    "I"}, NumRom[15]="\0";
printf("Dame un número arábigo: "); scanf("%d",&num);
while (num!= 0)
{
for (i=0;i<13;i++)
if (num>=arabigo[i])
{
strcat(NumRom, romano[i]);
num=num-arabigo[i];
break;
}
}
printf("%s\n",NumRom);getch(); return 0;
}
```

Una posible salida en pantalla
después de ejecutar el programa
sería:

Dame un número arábigo: 1756
MDCCLVI

Ejercicio 6

Ejercicio 7. Cambiar las vocales de 3 cadenas por un determinado carácter.

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
main ()
{
    char cadena[3][70], c;
    int i, j;
    printf ("Dame un carácter : ");
    scanf ("%c",&c);
    for (i = 0; i < 3; i++)
    {
        printf ("Cadena %d: ", i+1);
        fflush(stdin);
        gets (cadena[i]);
    }

    puts ("\nCADENAS NUEVAS");
    for (i = 0; i < 3; i++)
    {
        for (j = 0; cadena[i][j]; j++)
            if (strchr("aeiouAEIOU",cadena[i][j])) cadena[i][j] = c;
        puts (cadena[i]);
    }
    getch(); return 0;
}
```

Una posible salida en pantalla después de ejecutar el programa, sería:

Dame un carácter : \$

Cadena 1: La casa

Cadena 2: CUCEI

Cadena 3: Programacion

CADENAS NUEVAS

L\$ c\$ s\$

C\$ C\$ s\$

Pr\$ gr\$ m\$ c\$ s\$ n

Ejercicio 7

Ejercicio 8. Con la función *strncpy* podemos crear una nueva cadena con *n* caracteres de la cadena anterior, por ejemplo.

```
#include <stdio.h>
#include <string.h>
#include <conio.h>
main()
{
    char cad_ant[30],
    cad_nva[30]="programación";
    int n;
    printf("Dame una cadena: ");
    gets (cad_ant);
    printf("Dame el total de caracteres que quieres copiar, comenzando del primero: ");
    scanf("%d",&n);
    strncpy(cad_nva, cad_ant, n);
    printf ("La nueva cadena es: %s",cad_nva);
    getch(); return 0;
}
```

Una posible salida en pantalla después de ejecutar el programa sería:

Dame una cadena: CUCEI

Dame el total de caracteres que quieres copiar,
comenzando del primero: 3

La nueva cadena es: CUCgramación

Ejercicio 8

Ejercicio 9. Validar que el valor capturado sea un número entero.

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
main()
{
    char cad[]="\0"; int num;
    printf("Dame un número diferente de cero: ");
    gets(cad);
    num= atoi(cad);
    if (num!=0) printf("El número es: %d",num);
        else printf("Lo que capturaste NO es un número");
    getch(); return 0;
}
```

Una posible salida en pantalla después de ejecutar el programa sería:

Dame un número diferente de cero: 1235t5

El número es: 1235