

SEMANA #6 CADENAS DE CARACTERES

Una cadena de caracteres, como su nombre lo dice, no es más que una cadena de diversos tipos de caracteres que pueden ser letras, números, símbolos, etc. Hay que considerar que el tipo de dato que estamos usando es **char**. Para el uso de cadenas de caracteres utilizaremos la biblioteca **<string.h>** para poder ingresar sin problemas más de una letra.

Para guardar y leer cadenas utilizaremos dos variables:

La primera es **gets**, aunque nos permite leer y guardar cadenas, no es muy viable trabajar con ella esto debido a que si definimos nuestra variable [20], es decir, contendrá 20 caracteres únicamente, el gets permite almacenar más cantidad de caracteres de los definidos. Esto causa que se utilice mayor cantidad de memoria de la requerida y puede causar que el buffer de la función se llene muy pronto.

Dicho esto, para leer *strings* nos aseguraremos de usar la función **cin.getline**.

Esta función permite guardar y leer cadenas de caracteres bajo la librería **<string>**, y aplicara el número asignado de caracteres definido en la función.

```
//CADENAS DE CARACTERES
#include<iostream>
#include<stdlib.h>
using namespace std;

int main() {
    char palabra[] = "Alberto";
    char palabra2[] { 'a', 'l', 'b', 'e', 'r', 't', 'o' };
    char nombre[20];

    cout << "Digite su nombre:";
    cin.getline(nombre, 20, '\n');

    cout << nombre << endl;

    system("Pause");
    return 0;
}
```

Las cadenas de caracteres tienen varias funciones que interactúan entre ellas:

FUNCION *strlen()*

Esta función nos muestra la longitud que tiene la cadena asignada. Nos va a mostrar un número entero, por consiguiente, debemos declarar la longitud como una variable de tipo int.

Veamos este ejemplo:

```
//LONGITUD DE CADENA FUNCION strlen()
#include<iostream>
#include<stdlib.h>
using namespace std;

int main() {

    char palabra[] = "Hola que tal";
    int longitud = 0;
    longitud = strlen(palabra);

    cout << "Numero de elementos de la cadena es" << longitud<<endl;

    system("Pause");
    return 0;
}
```

FUNCION *strcpy_s()*:

Nos permite copiar el contenido de una cadena a otra cadena:

```
//COPIAR UNA CADENA FUNCIÓN strcpy_s()
#include<iostream>
#include<stdlib.h>
using namespace std;

int main() {

    char nombre[] = "Hola que tal";
    char nombre2[20];

    strcpy_s(nombre2, nombre);

    cout << nombre2 << endl;

    system("pause");
    return 0;
}
```

FUNCION strcmp():

Esta función nos permite comparar una cadena con otra, es decir identificar alfabéticamente cual cadena es primero

```
// COMPARAR CADENAS POR ORDEN ALFABETICO FUNCION strcmp()
#include<iostream>
#include<stdlib.h>
#include<string.h>
using namespace std;

int main() {
    char palabra1[] = "Becerro";
    char palabra2[] = "Avión";
    //comparamos
    if (strcmp(palabra1, palabra2) > 0) {
        cout << palabra1 << " esta despues alfabeticamente"<<endl;
    }
    system("pause");
    return 0;
}
```

FUNCION strcat_s():

Esta función nos permite añadir o concatenar una cadena con otra, es decir unir una cadena con otra.

```
//CONCATENAR CADENAS FUNCIÓN strcat_s()
#include<iostream>
#include<stdlib.h>
#include<string.h>
using namespace std;

int main() {
    char nombre[30] = "Hola que tal";
    char nombre2[30]=" como estas";

    strcat_s(nombre,nombre2);

    cout << nombre << endl;

    system("pause");
    return 0;
}
```

FUNCION `_strrev()`

Esta función nos invierte toda una cadena de caracteres

```
//INVERTIR CADENAS FUNCIÓN _strrev()

#include<iostream>
#include<stdlib.h>
#include<string.h>
using namespace std;

int main() {

    char nombre[30];
    cout << "Digite una cadena de caracteres:" << endl;
    cin.getline(nombre, 30, '\n');

    _strrev(nombre);

    cout << nombre << endl;

    system("pause");
    return 0;
}
```

FUNCION `_strupr_s()`

Esta función nos invierte toda una cadena de caracteres de minúscula a mayúscula.

```
// CADENAS A MAYUSCULAS FUNCIÓN _strupr_s()

#include<iostream>
#include<stdlib.h>
#include<string.h>
using namespace std;

int main() {

    char nombre[30];
    cout << "Digite una cadena de caracteres en minuscula:" << endl;
    cin.getline(nombre, 30, '\n');

    _strupr_s(nombre);

    cout << nombre << endl;

    system("pause");
    return 0;
}
```

FUNCION `_strlwr_s()`

Esta función nos invierte toda una cadena a minúscula

```
// CADENAS A MINUSCULAS FUNCIÓN _strlwr_s()

#include<iostream>
#include<stdlib.h>
#include<string.h>
using namespace std;

int main() {

    char nombre[30];
    cout << "Digite una cadena de caracteres en mayuscula:" << endl;
    cin.getline(nombre, 30, '\n');

    _strlwr_s(nombre);

    cout << nombre << endl;

    system("pause");
    return 0;
}
```

FUNCION `atoi()` y `atof()`

Estas funciones nos invierten toda una cadena de caracteres a números.

NOTA: estas dos funciones trabajan bajo la librería `<<stdlib.h>`

```
// CONVERTIR A VALOR ENTERO atoi()

#include<iostream>
#include<stdlib.h>
#include<string.h>
using namespace std;

int main() {

    char cadena[]="123";
    int numero;

    numero = atoi(cadena);

    cout << numero << endl;

    system("pause");
    return 0;
}
```

```
// CONVERTIR A VALOR FLOTANTE  atof()

#include<iostream>
#include<stdlib.h>
#include<string.h>
using namespace std;

int main() {

    char cadena[]="123.456";
    float numero;

    numero = atof(cadena);

    cout << numero << endl;

    system("pause");
    return 0;
}
```

Ejercicio:

Hacer un programa que determine si una palabra es palíndroma

```
// Hacer un programa que determine si una palabra es PALINDROMA

#include<iostream>
#include<stdlib.h>
#include<string.h>
using namespace std;

int main() {
    char palabra[20];
    char palabra2[20];
    cout << "Digite una palabra: \n";
    cin.getline(palabra, 20, '\n');
    //Primero copiamos la cadena
    strcpy_s(palabra2, palabra);

    //Segundo invertimos la cadena
    _strrev(palabra2);

    //Tercero comparamos
    if (strcmp(palabra, palabra2) == 0) {
        cout << "La palabra es PALINDROMA";
    }
    else {
        cout << "La palabra NO es palindroma";
    }
    system("pause");
    return 0;
}
```

Realice un programa que lea una cadena de caracteres de la entrada estándar y muestre en la salida estándar cuántas ocurrencias de cada vocal existen en la cadena.

```
/*Realice un programa que lea una cadena de caracteres de
la entrada estándar y muestre en la salida estándar cuántas
ocurrencias de cada vocal existen en la cadena.*/

#include<iostream>
#include<stdlib.h>
#include<string.h>
using namespace std;

int main() {
    char frase[30];
    int vocal_a = 0, vocal_e = 0, vocal_i = 0, vocal_o = 0, vocal_u = 0;

    cout << "Digite una frase: \n";
    cin.getline(frase, 30, '\n');

    for (int i = 0; i < 30; i++) {
        switch (frase[i]) {
            case 'a':vocal_a++;break;
            case 'e':vocal_e++;break;
            case 'i':vocal_i++;break;
            case 'o':vocal_o++;break;
            case 'u':vocal_u++;break;
        }
    }

    cout << "\n Contador de vocales\n";
    cout << "Vocal a:" << vocal_a << endl;
    cout << "Vocal e:" << vocal_e << endl;
    cout << "Vocal i:" << vocal_i << endl;
    cout << "Vocal o:" << vocal_o << endl;
    cout << "Vocal u:" << vocal_u << endl;

    system("pause");
    return 0;
}
```

PROGRAMACION ESTRUCTURADA EN C

(IN-3020; 3 créditos) Ing. Alberto Espinoza Zamora



PROGRAMACION ESTRUCTURADA EN C

(IN-3020; 3 créditos) Ing. Alberto Espinoza Zamora

