

**Objetivo de la práctica:**

- Aprender a utilizar el tipo cadena (o **string**) y las funciones definidas sobre él.
- Declarar y utilizar el tipo de dato estructurado registro (estructura o **struct**)

NOTA 1: Durante la práctica todos los ejercicios deberán ser guardados **temporalmente** en el directorio `\tmp`. Una vez finalizada la misma y transferidos los ficheros a un disquete, se deberá eliminar dicho directorio.

Conceptos básicos: tipo string

Las cadenas de caracteres (strings) permiten la manipulación de textos. Un string se puede considerar un vector de caracteres con unas operaciones añadidas. En C++, las cadenas de caracteres se representan mediante el tipo *string*. Para su uso es preciso utilizar `#include <string>`, no `<string.h>`.

Operaciones básicas definidas para *string*:

- Creación de variables:
`string palabra, frase;`
- Asignación:
`frase = palabra;`
`frase = "hola";`
- Acceso a los caracteres (como con vectores):
`palabra[0]`
- Comparación lexicográfica (`==`, `!=`, `<`, `>`):
`frase == palabra`
`frase > palabra`
- Lectura/escritura:
`cin >> palabra;`
`getline (cin, frase); //lee frase de cin hasta encontrar el fin de línea`
`cout << frase << endl;`
- Manipulación de textos (suponer `unsigned int i`):
`// indica si la frase es vacía`
`vacía = frase.empty();`
`// nº de caracteres de palabra`
`i = palabra.length();`
`// inserta palabra en la posición 3 de frase`
`frase.insert(3, palabra);`
`// concatena (une) palabra y "hola" y almacena el resultado en frase`
`frase = palabra + "hola";`
`// concatena (añade al final) palabra a frase`
`frase += palabra;`
`frase.append(palabra);`
`// borra 7 caracteres de frase desde la posición 3`
`frase.erase (3,7);`
`// sustituye (reemplaza) 6 caracteres de frase, empezando en la posición 1, por la cadena palabra`
`frase.replace (1, 6, palabra);`
`//busca palabra como una subcadena dentro de frase, devuelve la posición donde la encuentra`
`i = frase.find(palabra);`
`//devuelve la subcadena formado por 3 caracteres desde la posición 5 de la frase`
`palabra = frase.substr(5,3);`



Conceptos básicos: Registro

Una estructura es una colección de elementos de información, denominados campos, que representan información relevante sobre una entidad. Cada uno de estos campos puede pertenecer a un tipo de dato diferente y el acceso se realiza mediante el nombre del campo (y no mediante un índice como en los vectores). Ejemplo:

```
// información de interés sobre un empleado de una empresa
struct empleado
{
    string nombre;
    long int salario; // salario anual en euros
    string num_telefono;
};
```

Operaciones básicas

- Creación de una variable estructurada.
`empleado jose, antonio;`
- Asignación de estructuras.
`jose = antonio;`
- Acceso a los datos contenidos en la estructura (operador 'punto').
`jose.nombre = "Jose";`
- La lectura y escritura de la información de la estructura se debe hacer campo a campo.
`cin >> jose.nombre;`
`cin >> jose.salario;`
`cin >> jose.telefono;`
`cout << antonio.nombre << antonio.telefono << endl;`

Una estructura puede tener como campos otras estructuras. Ejemplo:

```
struct departamento
{
    string nombre;
    int numero_empleados;
    empleado jefe;
};
```

También es posible definir *arrays* de estructuras. Ejemplo:

```
typedef empleado plantilla [30];
```



PROBLEMAS

1. Realiza un programa que lea una frase por teclado e indique si la frase es un palíndromo o no lo es. Ejemplo de palíndromos:

Amigo, no gima. (Julio Cortazar)

Dábale arroz a la zorra el abad.

Amo la pacífica paloma.

A man, a plan, a canal: Panamá. (Lema electoral de Roosevelt, 1904.)

2. Escribe un programa que calcula y visualiza los palíndromos en una pasada de los números introducidos por teclado, si los tiene. El proceso consiste en leer un número entero por teclado, calcular si el número tiene un palíndromo en una pasada e indicarlo mediante un mensaje. Ejemplos:

Entrada → Introduce un número(0 Salir): 216

Proceso → Obtiene el traspuesto de 216, que es 612. Suma 216 + 612 = 828. Comprueba que es palíndromo.

Salida → Su palíndromo en un paso es: 828

Entrada → Introduce un número(0 Salir): 154

Proceso → Obtiene el traspuesto de 154, que es 451. Suma 154 + 451 = 605. Comprueba que no es palíndromo.

Salida → El 154 no tiene un palíndromo en una pasada.

Se requiere implementar una función que convierta un string a entero. La función devolverá un bool indicando si la conversión se ha realizado con éxito. Para convertir un carácter a entero se le resta el código del carácter '0'. También se necesita implementar una función que convierta un entero a string.

3. Realiza un programa que acepta por teclado una fecha de este siglo como una cadena, por ejemplo 01 12 03, la almacena en una estructura el día, mes y año e indica si la fecha es válida con el formato "dd de mmmm de aaa".

Nota: Un año es bisiesto si es divisible entre 400 o bien si es divisible entre 4 pero no entre 100.

4. Haz un programa que almacene información sobre vuelos de compañías aéreas. Los datos almacenados son: el número de vuelo, la compañía, la ciudad origen, la ciudad destino, el número de plazas totales y el número de plazas libres. Un ejemplo puede ser:

<pre>1 Iberia Valencia Madrid 250 10 2 Iberia Madrid Buenos-Aires 300 50 3 AirFrance Valencia Paris 250 70</pre>
--

El programa deberá permitir realizar desde menú las siguientes operaciones:

- a. Añadir vuelo
- b. Borrar vuelo
- c. Modificar el número de plazas libres.
- d. Ver todos vuelos
- e. Ver vuelos con plazas libres a un cierto destino.



Alternativos:

5. Escribe un programa que cuente el número de palabras que aparecen en una sola línea de entrada. Se entiende por palabra una secuencia no vacía de letras mayúsculas o minúsculas delimitada por el principio de la línea, el final de ella o cualquier carácter no alfabético.

Por ejemplo, si al ejecutar el programa el usuario introduce la línea que reproduce las primeras frases de La Regenta,

La heroica ciudad dormía la siesta. El viento sur, caliente y <retorno>

el programa debe contabilizar 11 palabras.

6. Ampliación de fechas:
 - a. Escribe una función para validar las fechas.
 - b. Crear una función para comparar fechas y que devuelva 1 si la primera es mayor, -1 si la segunda es mayor y 0 si son iguales.
 - c. Crear otra que devuelva los días entre dos fechas.
7. Se desea realizar un programa que solicite por teclado y almacene la temperatura y la pluviosidad de las provincias valencianas durante los 12 meses del año y calcule la media anual de cada provincia, la media provincial de cada mes y la media total de la comunidad.
8. Realiza un programa que lea una cadena por teclado palabra a palabra (se introduce "0" para terminar), sustituya las ocurrencias de una palabra que aparecen en la cadena y finalmente indica las posiciones en las que una palabra aparece en la cadena.