



Práctica 7: Cadenas

Objetivos.

Usar las estructuras de datos basadas en colecciones del mismo tipo, los tipos de datos estructurados homogéneos. En concreto: las cadenas de caracteres.

Operaciones básicas sobre variables de tipo string.

Operador de entrada	<pre>string s; cin >> s;</pre>	Cambia el valor de la cadena s por el valor introducido. Corresponde a una palabra. Solo se almacenan los caracteres introducidos hasta un espacio en blanco, tabulador o salto de línea.
Operador de salida	<pre>string s; cout << s;</pre>	Escribe el contenido de la cadena s en la pantalla.
Entrada de línea	<pre>string s; getline (cin, s);</pre>	Asigna la cadena introducida por teclado (hasta antes del carácter de final de línea) a la variable s .
Asignación	<pre>s = s2; s = "abc";</pre>	Asignación entre variables de tipo string. Asignación de una cadena constante a un string.
Acceso, Operador []	<pre>char c; string s = "abc"; s[1] = 'c'; c = s[1];</pre>	s cambia a "acc" c contiene 'c'
Longitud	<pre>int i; i = s.length();</pre>	i almacena el número de caracteres de s
Vacío	<pre>int i = 0; if (s.empty()) i++; if (s == "") i++;</pre>	En ambos ejemplos se incrementa en 1 el valor de i si la cadena está vacía
Concatenación	<pre>s2 = s2 + "xa"; s2 += "xa";</pre>	En ambos ejemplos se añade "xa" al final de s2
Substring	<pre>s = s2.substr(1,4); s = s2.substr(1,50);</pre>	Guarda en s la subcadena de s2 que comienza en el carácter con índice 1 y tiene una longitud de 4. Si la longitud indicada es mayor que los caracteres restantes, se detiene en el último carácter
Sustitución de substrings	<pre>s.replace(4,3,"xxx");</pre>	Sustituye los 3 caracteres de s comenzando por la posición 4 con la cadena "xxx"
Eliminación de substrings	<pre>s.erase(4,5); s.erase(4);</pre>	Borra los 5 caracteres de s desde la posición 4. Borra desde la posición 4 hasta el final.
Búsqueda de Patrones	<pre>s.find("ab",4); s.find("ab");</pre>	Devuelve la posición de la cadena "ab" en la cadena s comenzando la búsqueda en la posición 4. Comienza desde el principio de la cadena a buscar.



BLOQUE DE EJERCICIOS

Ejercicio 1 (ejercicio1.cpp): Implementa una función **ProcesarTexto** que dado un texto busque una palabra y la sustituya por otra. La función tiene que realizar tantas sustituciones como veces aparezca la palabra que se busca. La función tiene como parámetros de entrada: el texto a procesar, la palabra a buscar y la palabra con la que se reemplaza. La función devuelve la cantidad de sustituciones que se han realizado y el texto modificado.

Escribe un programa en C++ que:

Primero, se lean palabras de una en una desde el teclado y se vayan concatenado en un texto, separadas por un espacio en blanco, hasta que el usuario introduzca "fin" (no se incluye).

Segundo, se pida al usuario que introduzca una palabra a buscar y una palabra con la cual reemplazarla.

Finalmente, haciendo uso de la función anterior, se procese el texto, se informe de cuántas sustituciones se han realizado y se muestre el texto modificado.

Ejercicio 2 (ejercicio2.cpp): Implementa una función **Traduce** que se le pase una palabra en castellano y devuelva la palabra correspondiente en inglés. Sólo se sabe traducir 10 palabras. Define el tipo de dato matriz de strings de tamaño 10x2. Úsalo para guardar las palabras en castellano y su traducción en inglés. En la primera columna se almacenan las palabras en castellano y en la segunda en inglés.

```
const unsigned int NFIL = 10;
const unsigned int NCOL = 2;
typedef string Matriz[NFIL][NCOL];

string Traduce (const Matriz, string);
```

Escribe un programa en C++ que lea una frase en castellano, para cada palabra que compone la frase se llama a la función anterior y se muestra la traducción en inglés. Si no existe la palabra, ésta se traduce por un guion '-'. Por ejemplo:

"Mi abrigo es azul" corresponde con "My coat is blue".

"Yo soy de España" corresponde con "I am - Spain".

Ejercicio 3 (ejercicio3.cpp): Implementa una función **Frecuencia** que tenga como parámetro de entrada un texto y devuelva como parámetro de salida el porcentaje de aparición de cada una de las letras del abecedario inglés sin contar símbolos ortográficos ni dígitos ni caracteres en blanco (cuestión 3 de prerequisites). No si distingue entre mayúsculas y minúsculas (función `tolower()`).



Por ejemplo, ante la entrada "Hola, me llamo Luis. Soy alumno de primer curso del Grado de Multimedia. Mi aula es la 1.2. Me gusta programar en C++." se tienen las siguientes frecuencias de aparición:

```
a: 0.127907    (11/86)
b: 0           (0/86)
c: 0.0232558   (2/86)
d: 0.0581395   (5/86)
. . .
y: 0.0116279   (1/86)
z: 0           (0/86)
```

```
const unsigned short TAM = 26;
typedef float VectorF [TAM];
void Frecuencia (string, VectorF);
```

Las dos letras con mayor frecuencia de aparición en los textos en castellano son la 'e' (12.2%) y la 'a' (11.5%) y en los textos en inglés son la 'e' (12%) y la 't' (9.1%). Realiza un programa en C++ que pida por teclado un texto y determine si tiene más probabilidad de estar escrito en castellano que en inglés. Para construir un texto, puedes aplicar el método del ejercicio 1.

Relative frequencies of letters in other languages.

https://en.wikipedia.org/wiki/Letter_frequency

Ejercicio 4 (ejercicio4.cpp): En el código Morse se usan rayas y puntos para representar los dígitos del 0 al 9. Cada cifra se codifica como una cadena de cinco caracteres:

0: ----	5:
1: .----	6: -....
2: ..---	7: --...
3: ...--	8: ---..
4:-	9: ----.

Implementa una función **Descodifica** que dada una cadena de caracteres en código Morse devuelva el **número entero** equivalente en base decimal. Por ejemplo: la cadena ".....--- ---- ." corresponde con el número entero 529.

Implementa un programa en C++ que lea una cadena desde el teclado y muestre por pantalla el correspondiente número entero haciendo uso de la función anterior.

Ayuda: empieza a procesar la cadena por el final. Inicializa previamente un vector de diez strings con los códigos Morse de los dígitos y realiza una búsqueda. Para obtener el número decimal es necesario conocer la posición del dígito y usar la función potencia con base diez.



IMPORTANTE:

- Todos los programas deben seguir la Guía de Estilo e incluir los comandos de Doxygen para generar la documentación.
- Subir al Aula Virtual los archivos .cpp de uno en uno (enunciado y adicionales). Sólo los sube un miembro de la pareja.
- Fecha de entrega: durante la sesión de vuestro grupo de lab.