Guion de prácticas 2

Manipulación de Cadenas

DECSAI

Metodología de la Programación

Grado en Ingeniería Informática

Prof. David A. Pelta

Introducción

En este guion se pondrán en práctica los conceptos asociados a las cadenas tipo 'C' (`cstring`) a través de la implementación de funciones que resuelven tareas específicas. Dado que los datos de tipo `cstring` se tratan como un array de caracteres controlado por un valor centinela, estas tareas se pueden entender también como ejercicios para resolver problemas de manipulación de arrays.

1. Descripción del problema

Sean `C1,C2` dos cadenas de texto (`cstrings`). Se pide implementar un módulo `proceso` con las siguientes funciones:

- `longitud`, recibe una cadena `C1` y devuelve su longitud.
- `sonIguales`, devuelve verdadero si dos cadenas `C1, C2` son exactamente iguales. Devuelve falso en caso contrario. La función es "sensible" a mayúsculas y minúsculas.
- `carsUnicos`, devuelve cuantos caracteres únicos tiene una cadena `C1`. Por ejemplo, si `C1 = {ABCDA}`, el resultado es 3.
- `comprime`, recibe una cadena `C1` y devuelve otra `C2` "comprimida". Por ejemplo, si `C1 = {aaaaBBeeekhh}`, entonces `C2 = {a4B2e3k1h2}`. Si `C1 = {ABC}`, entonces `C2 = {A1B1C1}`.
- `descomprime`, realiza la operación inversa de la función anterior. Recibe una cadena `C1` y devuelve otra `C2` expandida o "descomprimida". Por ejemplo, si `C1 = {a3B2e3k1h2}`, entonces`C2 = {aaaBBeeekhh}`. Si `C1 = {*1}`, entonces`C2 = {*}`.
- `anagrama`, recibe dos cadenas `C1, C2` y comprueba si `C2` es un anagrama de `C1` (tiene los mismos caracteres pero en un orden diferente.). Por ejemplo, `C1 = {ballena}` y `C2 = {llenaba}` son anagramas. Lo mismo ocurre con `C1 = {acuerdo}` y `C2 = {ecuador}`. Por el contrario `C1 = {casa}` y `C2 = {saco}` no son anagramas. Asuma que las cadenas tienen la misma longitud.

Durante la sesión de prácticas que corresponda, se darán indicaciones adicionales para la resolución de los problemas planteados. Tenga en cuenta las siguientes recomendaciones.

Recuerde que los datos de tipo `cstring` se representan como un array de caracteres controlado por un valor centinela. Por tanto, debe evitar (siempre que sea posible) calcular la longitud de las cadenas y luego realizar algún procesamiento que requiera recorrerlas otra vez.

Para las funciones `comprime, descomprime` tendrá que manejar dos índices: uno para el símbolo actual y otro para el símbolo previo de la

cadena. En algún momento, deberá transformar un dígito numérico en el correspondiente valor de tipo `char` y viceversa. Recuerde que los valores de tipo char tienen asociado un valor entero en el código ASCII. En este código, el carácter '0' tiene asociado el valor 48, el '1' el 49, y así hasta que al '9' le corresponde el 57.

Finalmente, para la función `anagrama`, considere dos situaciones. Inicialmente suponga que en `C1` no existen caracteres repetidos e implemente una solución simple para el problema: para cada símbolo de `C1`, comprobar si aparece en `C2`.

Posteriormente, considere que puede haber símbolos repetidos. En este caso, la solución anterior no funcionará. El enfoque a considerar requiere "marcar" de alguna manera los símbolos de `C2` ya procesados. En principio podríamos utilizar un símbolo especial como marcador (por ejemplo, `'*'`) pero no se ha establecido ninguna restricción sobre los posibles símbolos en las cadenas de entrada.

La solución que se sugiere es "marcar" las posiciones visitadas. Para ello, puede utilizar un array de valores booleanos. Así, dado un símbolo `c` de `C1`, lo buscaremos en `C2`. Si lo encontramos, marcamos como `true` la posición correspondiente en el array. La búsqueda de `c` en `C2` solo se realiza sobre las posiciones aún no marcadas (están en `false`).

2. Tareas

Dispone en PRADO de la parte pública del módulo `proceso` (`fichero proceso.h`) y un programa de prueba (`prueba.cpp`) QUE NO DEBE MODIFICAR. Puede añadir todas las funciones adicionales que crea convenientes. Justifique si deben ser públicas o privadas.

Observe la estructura de los comentarios en el fichero `proceso.h`. Dicha estructura se utilizará más adelante para generar automáticamente la documentación del código en formato `html`, mediante el programa `Doxygen`.

LAS INSTRUCCIONES PARA LA ENTREGA SE PUBLICARÁN PRÓXIMAMENTE.