UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

Ingeniería en Computación

Actividad 7

Programación Estructurada

***

ALUMNO: Arredondo Urbalejo Isai Alexis

MATRÍCULA: 368747

GRUPO: 932

PROFESOR: Pedro Nunez Yepiz

Introducción

Una de las estructuras de datos más esenciales en este contexto son las cadenas de caracteres o, simplemente, cadenas. Las cadenas son un elemento omnipresente en cualquier programa, ya sea para procesar texto, manipular datos o interactuar con el usuario.

En C++, una cadena se refiere a una secuencia de caracteres que se utiliza para representar texto o datos de caracteres en un programa. Estas secuencias de caracteres se almacenan como arrays de caracteres (`char`) y están terminadas con un carácter especial nulo `'\0'`. Este carácter nulo indica el final de la cadena y es esencial para que las funciones de manipulación de cadenas en C++ funcionen correctamente.

Competencia

El objetivo de este informe es proporcionar una comprensión sólida de cómo trabajar con cadenas en C++, aprovechando al máximo las capacidades de este lenguaje de programación, y resolviendo desafíos comunes que los programadores enfrentan al manipular datos de texto.

Fundamentos

1. Arrays de Caracteres

Los arrays de caracteres son la forma más primitiva de representar cadenas en C++. Consisten en una secuencia de caracteres seguida por un carácter nulo (`'\0'`) que marca el final de la cadena. Algunas consideraciones importantes incluyen:

- Declaración e Inicialización: Los arrays de caracteres se declaran como `char` seguidos por el nombre de la variable y se inicializan con comillas dobles.

- Manipulación: La manipulación de arrays de caracteres puede ser propensa a errores, ya que es necesario realizar un seguimiento cuidadoso de los límites y del carácter nulo.

3. Clase `std::string`

La clase `std::string` es una parte fundamental de la Biblioteca Estándar de C++ (STL) y proporciona una forma más segura y conveniente de trabajar con cadenas. Algunos aspectos clave son:

- Declaración y Uso: Para utilizar `std::string`, debes incluir la cabecera `<string>`. Luego, puedes declarar una cadena utilizando `std::string` seguido por el nombre de la variable.

- Ventajas: `std::string` maneja automáticamente la gestión de memoria, evita desbordamientos de búfer y proporciona una amplia gama de funciones y operadores para realizar operaciones comunes de manipulación de cadenas.

4. Operaciones Comunes con Cadenas

Independientemente de si se utilizan arrays de caracteres o `std::string`, existen operaciones comunes que se pueden realizar con cadenas en C++:

- Concatenación: Combina dos o más cadenas en una sola.

- Búsqueda: Encuentra la posición de una subcadena dentro de una cadena.

- Inserción y Eliminación: Agrega o quita caracteres en una posición específica de la cadena.

- Comparación: Compara dos cadenas para determinar su igualdad o relación lexicográfica.

Guzman, H. C. (s. f.). ***Cadenas de caracteres | Apuntes Lenguaje C++ | Hektor Profe*****.**
https://docs.hektorprofe.net/cpp/08-cadenas-caracteres/

Paszniuk, R. (s. f.-a). ***ARREGLOS y CADENAS en C++ – programación*****.**
https://www.programacion.com.py/escritorio/c/arreglos-y-cadenas-en-c

Programación en C++/Arrays y cadenas de texto - Wikilibros**. (s. f.).**
https://es.wikibooks.org/wiki/Programaci%C3%B3n_en_C%2B%2B/Arrays_y_cadenas_de_texto

Procedimiento

Parte 1

Realizar un programa que contenga funciones que realice lo siguiente.

1.- Leer una cadena y desplegarla de la siguiente manera:
(Realizar una función para cada salida)

cadena: Ensenada

SALIDA 1	SALIDA 2	SALIDA 3
ENSENADA	ADANESNE	E N S E N A D A
SALIDA 4	**SALIDA 5**	**SALIDA 6**
A D A N E S N E	ENSENADA ENSENAD ENSENA ENSEN ENSE ENS EN E	ADANESNE ADANESNE ADANES ADANE ADAN ADA AD A
SALIDA 7	**SALIDA 8**	**SALIDA 9**

ENSENADA NSENADA SENADA ENADA NADA ADA DA A	ADANESNE DANESNE ANESNE NESNE ESNE SNE NE E	NSND SALIDA 10 EEAA

LIGAS DE INTERÉS Y AYUDA

http://www.uco.es/grupos/eatco/informatica/metodologia/cadenasyarrays.pdf

http://platea.pntic.mec.es/vgonzale/cyr_0204/cyr_01/control/lenqua_C/cadenas.htm

http://c.conclase.net/curso/?cap=008

https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=934:funciones-para-cadenas-en-c-longitud-sizeof-strlngh-y-strcpy-strlen-strcat-strcmp-ejemplos-cu00535f&catid=82&Itemid=210

https://es.slideshare.net/angenio2/programacin-1-cadenas-en-c

https://pablohaya.com/2013/10/12/diferencia-entre-scanf-gets-y-fgets/

Parte 2

REALIZA LAS SIGUIENTES FUNCIONES:

1.- Función que reciba como parámetro una cadena y la convierta a MAYÚSCULAS
2.- Función Que reciba como parámetro una cadena y la convierta a MINÚSCULAS
3.- Función que reciba como parámetro una cadena y la convierta a CAPITAL
4.-Función que reciba como parámetro una cadena y retorne la cantidad de caracteres que tiene la cadena.
5.-Función que reciba como parámetro una cadena y retorna una cadena con sus caracteres acomodados de forma inversa (al revés)
6.-Función que reciba como parámetro una cadena y genere una nueva cadena basada en la origina pero sin espacios.

7.-Función que sirva para leer una cadena y solo permite caracteres alfabéticos (A...Z) y el espacio, donde una cadena no puede comenzar o terminar con espacio, no debe tener dos espacios seguidos. retornar la cadena ya sea como parámetro o variable.
8.-Función que reciba como parámetro una cadena, y utilizando las funciones anteriores, imprima en MAYÚSCULAS, MINÚSCULAS , CAPITAL, SIN ESPACIOS, AL REVÉS la cadena original.
9.-Función que reciba como parámetro una cadena, y desplegar la leyenda si la cadena es un palíndromo SI o NO
(VALIDADA AL 100% NO NÚMEROS, NO DOBLES ESPACIOS Y SOLO MAYÚSCULAS EN LA CADENA)

Resultados y Conclusiones

Como se vio en el reporte las cadenas son muy importantes en la programación. Las operaciones comunes con cadenas, como la concatenación, la búsqueda, la inserción y la eliminación nos dan mucha movilidad a la hora de hacer los programas.