

Los ejercicios en adelante requieren de cumplir los Requerimientos de Valor Agregado en Código Fuente (hasta el requerimiento HH):

- Es indispensable en la entrega de cada programa, que el código fuente sea completamente en idioma Español (a excepción de lo correspondiente a la API del lenguaje).

IMPLEMENTAR LOS SIGUIENTES PROGRAMAS EN LENGUAJE C ANSI

1. Implementar una función que haga lo mismo que `strlen()` de `string.h`, recibiendo igual un arreglo de `char` y usar dicha función desde `main()`. El programa no incluya más librerías que `stdio.h` y `stdlib.h`. Ingresar desde consola la cadena a usar con la función diseñada y, al final, imprimir el tamaño de la cadena leída.
2. Implementar una función que haga lo mismo que `strcpy()` de `string.h`, recibiendo igual dos arreglos de `char` y usar dicha función desde `main()`. El programa no incluya más librerías que `stdio.h` y `stdlib.h`. Ingresar desde consola la primer cadena a usar con la función diseñada y, al final, imprimir la segunda cadena, luego de pasarla por la función.
3. Implementar una función que haga lo mismo que `strcat()` de `string.h`, recibiendo igual dos arreglos de `char` y usar dicha función desde `main()`. El programa no incluya más librerías que `stdio.h` y `stdlib.h`. Ingresar desde consola una cadena y pasarla a la función diseñada. Al final, imprimir el valor devuelto por la función.
4. Implementar una función que haga lo mismo que `strcmp()` de `string.h`, recibiendo igual dos arreglos de `char` y usar dicha función desde `main()`. El programa no incluya más librerías que `stdio.h` y `stdlib.h`. Ingresar desde consola las dos cadenas a usar con la función diseñada y, al final, imprimir verdadero o falso, de acuerdo con el valor retornado por la función.
5. Implementar un programa que lea de la consola letras para un arreglo de caracteres, luego convertir a minúsculas todos los caracteres y finalmente imprimir el arreglo de caracteres (ya en minúsculas). Tomar en cuenta que el código ASCII de la "A" es 65, el de la "Z" es 90. El de la "a" es 97 y el de la "z" es 122. Ignorar para la conversión los caracteres acentuados y la "ñ".
6. Implementar un programa que lea de la consola letras para un arreglo de caracteres, luego convertir a mayúsculas todos los caracteres y finalmente imprimir el arreglo de caracteres (ya en mayúsculas). Tomar en cuenta que el código ASCII de la "A" es 65, el de la "Z" es 90. El de la "a" es 97 y el de la "z" es 122. Ignorar para la conversión los caracteres acentuados y la "ñ".
7. Implementar una función que haga lo mismo que `strchr()` de `string.h`, recibiendo igual dos arreglos de `char` y usar dicha función desde `main()`. El programa no incluya más librerías que `stdio.h` y `stdlib.h`. Ingresar desde consola una cadena de caracteres y un carácter a buscar en la cadena ingresada; pasar a la función diseñada los dos valores leídos. Al final, imprimir el valor devuelto por la función, es decir, la posición de la primera ocurrencia del carácter buscado o en su defecto un -1 si no es encontrado.

8. Implementar un programa que realice lo siguiente:
 - a. En una subrutina, inicializar desde consola un arreglo de 25 enteros, colocando algunos elementos repetidos. **Proveer desde consola todos los elementos para el arreglo.**
 - b. Al salir de dicha subrutina se imprima el contenido del arreglo (el separador de elementos mostrados en pantalla sea un espacio).
 - c. Otra subrutina reciba como parámetros por referencia a dicho arreglo y la cantidad de elementos.
 - d. Al salir de esta última subrutina, los elementos repetidos hayan sido eliminados y el parámetro de cantidad de elementos haya sido actualizado.
 - e. Imprimir el contenido actual del arreglo y la cantidad de elementos.

* Por ejemplo: si en la lista se encuentran los elementos 1,8,7,9,2,7,3,7,3,3 la lista quede como 1,8,7,9,2,3.

Los ejercicios en adelante requieren de cumplir los Requerimientos de Valor Agregado en Código Fuente (hasta el requerimiento KK):

En adelante, entiéndase por “programa ABC”, aquél que gestione un listado de registros y cuente con las opciones de menú: “Alta”, “Baja”, “Consulta” (de un registro), “Cambio” (de datos de un registro), “Listado” (de todos los registros) y “Búsqueda” (de un registro particular o varios que coincidan con un determinado patrón de búsqueda).

9. Modificar el programa ABC del ejercicio 48 de “Ejercicios PRO.pdf”, para cumplir con lo todo lo solicitado, pero que al principio del programa se solicite al usuario la cantidad máxima de registros que podrán ser capturados. El programa funcione igual que la versión anterior del programa, pero soportando la captura de tantos registros como se haya especificado por el usuario. Declarar y definir el arreglo de registros en main(), y hacer paso de parámetros por referencia, pasando el arreglo de registros como parámetro desde main() hacia las subrutinas encargadas del ABC. Este programa no requiera de variables globales.
10. Modificar el programa ABC del ejercicio 51 de “Ejercicios PRO.pdf”, para cumplir con todo lo solicitado pero, declarar y sin definir una variable global de tipo apuntador a registro, incluir una subrutina de inicialización que se ejecute al principio del programa y ahí se solicite al (la) usuari@ la cantidad máxima de registros que podrán ser capturados; luego de lo anterior sea creado el arreglo de registros usando “malloc”; posterior a ejecutar la subrutina de inicialización, el programa funcione igual que la versión anterior del programa. Al finalizar el programa, toda la memoria solicitada sea liberada justo antes de que termine la subrutina “main”.

IMPLEMENTAR LOS SIGUIENTES PROGRAMAS EN LENGUAJE C++ ANSI

11. Modificar el ejercicio anterior para que funcione tal cual se describió, pero, solicitar la memoria para el arreglo de registros usando new y delete en lugar de malloc y free. Lo anterior usando el tipificado de dato como struct y no class.
12. Implementar un programa, donde una subrutina reciba **una cadena por referencia** y busque en dicha cadena un carácter indicado por el usuario y, si se encuentra, removerlo de la cadena. Remover de la cadena tantas ocurrencias como se encuentren del citado carácter. Imprimir la cadena con el char removido. El programa incluya dos variantes en carpetas diferentes: en la “variante1”, la cadena sea un arreglo de caracteres y en la “variante2” sea un objeto string.

13. Implementar una subrutina que convierta un int en un arreglo de caracteres (no un objeto string); la firma de la subrutina contenga un int, el puntero a un arreglo de caracteres y el tamaño del arreglo de caracteres. El arreglo de caracteres sea creado antes de llamar a dicha subrutina y se devuelva verdadero si y sólo si la cantidad de caracteres a escribir en el arreglo pudieron ser almacenados en el arreglo. Al salir de la subrutina, imprimir el contenido del arreglo de caracteres. Por ejemplo: si la subrutina recibe un entero "2017" y el arreglo cuenta al menos con 5 celdas, el arreglo de caracteres contenga en la celda 0 el código ASCII del '2', en la celda 1 el ASCII del '0', etc. donde finalmente la celda 4 contenga el carácter nulo. Evitar usar la clase stringstream y cualquier otra clase o subrutina de librería.