

Ejercicios Prácticos de Manejo de Cadenas en "C"

Para la solución de los siguientes ejercicios, no debes imprimir resultados dentro de las funciones que escribas. Los resultados deben ser devueltos mediante el return de la función.

Luego de escribir cada función, probala, invocándola desde el bloque principal del programa, pasándole distintos valores, ó los casos de prueba indicados en cada ejercicio.

1. Escribir una función que reciba como parámetro una cadena de caracteres y devuelva la cantidad de dígitos numéricos que hay en la cadena.
2. Escribir una función que reciba como parámetro una cadena de caracteres y devuelva la cantidad de caracteres no alfabéticos que hay en la cadena. No tener en cuenta la ñ, ni las vocales acentuadas.
3. Escribir una función en C, que reciba como parámetro una cadena de tipo t_cadena, y devuelva la cantidad de palabras que hay en la cadena. Considerar que una palabra está separada de otra, por uno ó más blancos. No recorrer la cadena más de una vez. Declare un tipo t_cadena acorde. Escribir el programa que incluya a la función y las invocaciones con los siguientes caso de prueba:

```
cant_palabras("El sol es amarillo.") devuelve 4
cant_palabras(" Hoy es 30 de Junio de 2021 ") devuelve 7
cant_palabras("30/06/2021") devuelve 1
cant_palabras("El precio es $2.000.- ") devuelve 4
cant_palabras(" ") devuelve 0
```

4. Escribir una función en C que reciba como primer parámetro, una cadena de tipo t_cadena, y devuelva como segundo parámetro, la misma cadena pero invertida, sin considerar los blancos. Sólo se deben invertir las posiciones con caracteres ciertos, y no aquellas posiciones potenciales a ser utilizadas pero que podrían contener caracteres inciertos. Recorrer la cadena a lo sumo una vez. Declare un tipo t_cadena acorde. Escribir el programa que incluya a la función y las invocaciones con los siguientes casos y las impresiones para comprobar que devuelven lo correcto.

Casos:

```
invertir_cadena("Hola, me llamo Ana.", cad_invertida) ==> cad_invertida: ".anAomalllem,aloH"
invertir_cadena("234561", cad_invertida) ==> cad_invertida: "165432"
invertir_cadena(" ", cad_invertida) ==> cad_invertida: ""
invertir_cadena("Somos o No somos", cad_invertida) ==> cad_invertida: "somosoNsomoS"
```

5. Escribir una función para validar una nueva clave de acceso.

La función deberá recibir una cadena de caracteres, que contendrá la clave candidata, que ya fue ingresada previamente por el usuario.

Devolverá true o false, dependiendo de si cumple o no, con las siguientes condiciones:

- La clave debe estar formada únicamente por, entre 6 y 12 caracteres numéricos
- La cantidad de dígitos pares debe ser mayor a la de los impares.

A los sumo debe recorrer una vez la cadena.

Evite realizar ciclos innecesarios.

Compruebe el correcto funcionamiento, incluyendo los siguientes casos de prueba:

```
validar("j20893") devuelve false
validar("20893a") devuelve false
validar("208X930") devuelve false
validar("20201") devuelve false
validar("23445776") devuelve false
validar("089010") devuelve true
validar("02784532132567") devuelve false
validar("027845320011") devuelve true
```

6. Escribir una función para validar una nueva clave de acceso.

La función deberá recibir una cadena de caracteres, que contendrá la clave candidata, ingresada previamente por el usuario.

Devolverá true o false, dependiendo de si cumple o no con las siguientes condiciones:

- La clave debe estar formada únicamente por, entre 4 y 8 caracteres numéricos
- Los caracteres no pueden ser todos iguales

Ejemplos:

```
validar("j2020") devuelve false
validar("2021a") devuelve false
validar("20X21") devuelve false
validar("2220") devuelve true
validar("23445776") devuelve true
validar("089") devuelve false
validar("027845321") devuelve false
validar("02784532") devuelve true
validar("33333") devuelve false
```

7. Escribir una función en C, que reciba una cadena que representa una palabra y devuelva si la misma es o no un palíndromo.
Una palabra es un palíndromo, si se lee igual en ambos sentidos.

Probar la función con los siguientes casos de prueba:

- anilina (Es palíndromo)
- ojo (Es palíndromo)
- radar (Es palíndromo)
- reconocer (Es palíndromo)
- algoritmos (No es palíndromo)
- programas (No es palíndromo)

Evitar realizar ciclos innecesarios.

8. Escribir una función en C, que reciba una cadena que representa una palabra o una frase y devuelva si la misma es o no un palíndromo.
Un palíndromo, es una palabra o frase que se lee igual en ambos sentidos.

Probar la función con los siguientes casos de prueba:

- anilina (Es palíndromo)
- ojo (Es palíndromo)
- radar (Es palíndromo)
- reconocer (Es palíndromo)
- amar a roma (Es palíndromo)
- ana lava lana (Es palíndromo)
- a ti no, bonita (Es palíndromo)
- algoritmos y programas (No es palíndromo)
- lenguaje C (No es palíndromo)

Evitar realizar ciclos innecesarios.

9. Escribir una función que recibirá por parámetro, una palabra, que representa un sustantivo en singular.
La función deberá devolver, el plural de dicho sustantivo, aplicando las siguientes reglas:
- a.** Agregar una "s" al final, si la palabra termina en vocal sin acento.
 - b.** Agregar una "s" al final, si la palabra termina con una é (acentuada).
 - c.** Si la palabra termina en "z", la reemplazamos por "ces".
 - d.** Agregamos "es" al final, si la palabra termina en una consonante (a excepción de la "s", la "z", y la "x"), ó si la palabra termina con las vocales acentuadas: á, í, ó, ú.
 - e.** Si el sustantivo termina en "s" ó "x", entonces el plural es igual al singular, por lo tanto la función deberá devolver lo mismo que recibió.

Para cada uno de los siguientes ejercicios, deberás escribir un programa compuesto por una o más funciones. Recordá que lo adecuado es que una función realice sólo una tarea, por eso, antes de ponerte a programar cada una de las soluciones de los ejercicios, diseña la solución indicando cuáles serán las funciones que escribirás y compondrán tu programa.

10. Escribir un programa modular que solicite el ingreso de dos palabras y luego muestre las mismas según la longitud de cada una de ellas, mostrando primero la de menor longitud.
A igualdad de longitud, mostrar primero la menor de las dos, en términos de orden alfabético.
Las palabras tienen una longitud máxima de 15 caracteres.
11. Escribir un programa modular en C, que solicite el ingreso de no más de 20 palabras. C
Cada palabra no tendrá más de 20 caracteres. Informar:
 1. Mostrar las palabras ingresadas.
 - 2.Cuál es la palabra más larga.
 3. Solicitar el ingreso de una palabra, e indicar si la misma se encuentra entre las ingresadas.
12. Escribir un programa modular en C, que solicite el ingreso de 1 oración, de no más de 100 caracteres.
Luego informar la cantidad de vocales minúsculas, no acentuadas, que hay en la oración.
13. Escribir un programa modular en C, que solicite el ingreso de 3 oraciones, de no más de 50 caracteres cada una.
Luego informar:
 - 1.Cuál es la oración más larga.
 2. Si hay al menos 2 oraciones iguales.
 3. Solicitar el ingreso de una palabra o parte de una oración, e indicar si la misma se encuentra en las oraciones, y en cuales.
14. Escribir un programa que solicite el ingreso de un texto que será enviado mediante un telegrama. Luego de ingresado, se deberá informar la cantidad de palabras que lo componen y el importe a abonar por el solicitante.
El texto sólo puede contener, letras, números y los siguientes signos de puntuación: . , ; : ()
Para el cálculo de las palabras, considerar que una palabra estará separada de otra, por uno ó más blancos.
Para el cálculo del importe a abonar, deberá considerar que cada palabra pagará \$10 por cada 3 caracteres. Por las fracciones menores a los 3 caracteres, pagará \$8.