



Laboratorio 06

Cadena de caracteres y punteros

1. Escribe una función que reciba una cadena como puntero y cuente el número de vocales.
2. Escribe una función que reciba una cadena y cambie todas las letras minúsculas por mayúsculas usando punteros.
3. Implementa una función que invierta una cadena de caracteres utilizando punteros.
4. Crea una función que reciba dos cadenas y determine si son anagramas utilizando únicamente punteros. Un **anagrama** es cuando dos palabras contienen las mismas letras en diferente orden. Es decir, ambas palabras deben tener el mismo número de caracteres y las mismas letras, aunque posiblemente en un orden distinto.
 - **Entrada:** Dos cadenas de caracteres.
 - **Proceso:** Comprobar que las cadenas tienen la misma longitud, luego ordenar ambas cadenas alfabéticamente (puedes usar un algoritmo de ordenamiento como burbuja con punteros) y finalmente comparar letra por letra usando punteros.
 - **Salida:** Un mensaje que indique si las cadenas son anagramas o no.

Ejemplo:

Ingrese primera cadena: roma

Ingrese segunda cadena: amor

Salida: Las cadenas son anagramas.

Ingrese primera cadena: casa

Ingrese segunda cadena: saco

Salida: Las cadenas no son anagramas.

5. Escribe una función que reciba una cadena y elimine todos los espacios en blanco, modificando la cadena original mediante punteros.
6. Escribe una función que ordene alfabéticamente los caracteres de una cadena usando el algoritmo de burbuja (bubble sort) y punteros.

7. Crea una función que determine si una cadena es un palíndromo utilizando punteros. Un **palíndromo** es una palabra o frase que se lee igual de izquierda a derecha que de derecha a izquierda. Por simplicidad, ignora los espacios y considera las mayúsculas/minúsculas como iguales.

- **Entrada:** Una cadena de caracteres.
- **Proceso:**
 - Ignorar espacios.
 - Convertir todas las letras a minúsculas (si es necesario).
 - Comparar usando dos punteros: uno que avance desde el inicio, y otro que retroceda desde el final.
 - Si todas las letras coinciden, es un palíndromo.
- **Salida:** Un mensaje indicando si la cadena es un palíndromo o no.

Ejemplo 1:

Ingrese una cadena: oso

Salida: La cadena es un palíndromo.

Ejemplo 2:

Ingrese una cadena: anita lava la tina

Salida: La cadena es un palíndromo.

Ejemplo 3:

Ingrese una cadena: computadora

Salida: La cadena no es un palíndromo.

8. Implementa una función que compare dos cadenas caracter por caracter usando punteros (sin usar la función strcmp).
9. Desarrolla una función que reciba una cadena y una letra, y elimine todas las apariciones de esa letra en la cadena usando punteros.
10. Implementa una función que reciba un arreglo de cadenas y ordene alfabéticamente las cadenas usando el algoritmo de selección.
11. Programa una función que reciba una cadena y retorne una nueva cadena que sea la original pero duplicando cada vocal.
12. Crea una función que reciba una cadena y reemplace cada dígito numérico por un asterisco (*), usando punteros.
13. Escribe una función que cuente cuántas palabras contiene una cadena (considerando palabras separadas por espacios).

14. Desarrolla una función que reciba dos cadenas y concatene la segunda al final de la primera utilizando punteros.
15. Implementa una función que, dado un arreglo de cadenas, las ordene de mayor a menor longitud utilizando el algoritmo de inserción.
16. Escriba una función en C++ que reciba una cadena y convierta todas las vocales en símbolos específicos. Los símbolos a utilizar para cada vocal serán los siguientes:

- $a \rightarrow @$
- $e \rightarrow \#$
- $i \rightarrow \$$
- $o \rightarrow \%$
- $u \rightarrow \wedge$

La función debe recorrer la cadena y, cada vez que encuentre una vocal, sustituirla por el símbolo correspondiente. Tenga en cuenta que la función debe ser sensible tanto a mayúsculas como a minúsculas, por lo que se deben hacer las conversiones adecuadamente para ambas formas de las vocales.

Por ejemplo, si la entrada es la siguiente cadena:

"Hola, ¿cómo estás?"

El resultado debe ser:

"H@l#, ¿c%m# st#\$s?"

La firma de la función debe ser la siguiente:

```
void convertirVocales(char cad[]);
```

Donde cad es la cadena que será modificada.

El programa principal debe invocar la función y mostrar la cadena convertida.

17. Crea una función que reciba un arreglo de palabras y ordene las palabras que empiezan con vocal primero, luego las demás, manteniendo el orden interno.
18. Implementa una función recursiva que cuente el número de letras mayúsculas en una cadena utilizando punteros.
19. Programa una función que reciba una cadena y retorne una nueva cadena que contenga únicamente las letras (eliminando números y símbolos).
20. Escribe una función que ordene un arreglo de cadenas de acuerdo a la última letra de cada cadena.
21. Escriba un programa para encontrar la palabra más grande y más pequeña en una cadena.
22. Escriba un programa para reemplazar los espacios de una cadena con un carácter específico.

23. Escribe un programa en C++ que realice lo siguiente:

- Lea un número entero n , correspondiente al número de líneas que tendrá un párrafo.
- Posteriormente, lea el texto línea por línea. Cada línea puede tener como máximo 40 caracteres.
- Para cada línea leída, cuente el número de vocales (a, e, i, o, u) que contiene.
- Finalmente, imprima la cantidad de vocales encontrada en cada línea.

Notas importantes:

- Después de leer el número de líneas, debes limpiar el buffer de entrada usando `cin.ignore()`.
- Para leer cada línea de texto se recomienda usar la función `getline(cin, cadena)`.
- Puedes almacenar cada línea en un arreglo de cadenas de caracteres o trabajar con cada línea individualmente.

Ejemplo de ejecución esperada:

```
Ingrese el número de líneas a escribir: 4
Ingrese 4 líneas de texto (máximo 40 caracteres por línea):
una cucaracha, cha, cha, cha
baila con un grillo, cri, cri, cri
me fui a la laguna
hola mundo
```

```
La 1° línea tiene 9 vocales.
La 2° línea tiene 10 vocales.
La 3° línea tiene 8 vocales.
La 4° línea tiene 4 vocales.
```

24. Escribe un programa en C++ que realice lo siguiente:

- Lea una cadena completa ingresada por el usuario.
- Busque un conjunto de palabras dentro de la cadena.
- Por cada palabra encontrada, imprima un mensaje indicando que fue encontrada.

Ejemplo:

```
Palabras buscadas: {"juan","pedro","maría","rosa"}
```

```
Ingrese la cadena: juan busca a rosa y también a maría
```

```
Salida:
encontré a juan
encontré a rosa
encontré a maría
```

25. Todos los procesadores de textos tienen la opción de "buscar y reemplazar". Escriba un programa en C++ que implemente la función:

```
void reemplazar(char *linea, char *strbusc, char *streemp);
```

El programa debe realizar lo siguiente:

- Leer una cadena de caracteres `strbusc` que será la cadena a buscar.
- Leer una cadena de caracteres `streemp` que será la cadena que la reemplazará.
- Leer una línea de texto `linea` (máximo 100 caracteres).
- Buscar en la línea todas las apariciones de la cadena `strbusc` y reemplazarlas por `streemp`.

Sugerencia: Utilizar las funciones: `strlen`, `strcpy` y `strcat`.

Ejemplo de ejecución:

Ingrese la cadena a buscar: Mary

Ingrese la cadena a reemplazar: Naty

Ingrese el texto: Amo a Mary. Mañana conoceré a los padres de Mary y el domingo iré con Mary al cine.

Texto reemplazado:

Amo a Naty. Mañana conoceré a los padres de Naty y el domingo iré con Naty al cine.

Modifique el ejercicio anterior, considerando ahora el prototipo

```
void reemplazar(std::string &linea, const std::string &buscar,  
               const std::string &reemplazo);
```

Para realizar el reemplazo, puedes usar las funciones de la clase `std::string`, como `find`, `replace`, y `substr`.

26. Defina las siguientes constantes simbólicas:

```
#define FIL 10  
#define COL 100  
#define ARTICULOS 3
```

y las siguientes matrices:

```
char recital[FIL][COL];  
char derechos[ARTICULOS][COL] = {  
    "Todos los seres humanos nacen libres e iguales en dignidad y derechos.",  
    "Todo individuo tiene derecho a la vida, a la libertad y a la  
    de su persona.",  
    "Nadie podrá ser arbitrariamente detenido, preso ni desterrado"  
};
```

Luego, implemente la función:

```
void generar(char (*des)[COL], char (*ori)[COL], int items_des, int items_ori);
```

de manera que, al ejecutarse `generar(recital, derechos, FIL, ARTICULOS)`; se copie en cada fila de la matriz `recital` de forma aleatoria una de las tres cadenas de la matriz `derechos`.

Finalmente, implemente la función:

```
void mostrar(char (*ptr)[COL], int items);
```

de manera que, al ejecutarse `mostrar(recital, FIL)`; se imprima un recital de derechos fundamentales, como el siguiente:

- 0: Todo individuo tiene derecho a la vida, a la libertad y a la seguridad de su persona.
- 1: Nadie podrá ser arbitrariamente detenido, preso ni desterrado
- 2: Todo individuo tiene derecho a la vida, a la libertad y a la seguridad de su persona.
- 3: Nadie podrá ser arbitrariamente detenido, preso ni desterrado
- 4: Todo individuo tiene derecho a la vida, a la libertad y a la seguridad de su persona.
- 5: Nadie podrá ser arbitrariamente detenido, preso ni desterrado
- 6: Nadie podrá ser arbitrariamente detenido, preso ni desterrado
- 7: Todo individuo tiene derecho a la vida, a la libertad y a la seguridad de su persona.
- 8: Todo individuo tiene derecho a la vida, a la libertad y a la seguridad de su persona.
- 9: Todos los seres humanos nacen libres e iguales en dignidad y derechos.

27. Escriba un programa en C++ para eliminar un conjunto de palabras de un texto. Para ello, implemente la función:

```
string eliminar(char cad[], int tam);
```

Desde la función `main()` invóquela como:

```
string rpta = eliminar(linea, l);
```

donde `l` es la longitud de caracteres de la línea.

Considere, por ejemplo, la siguiente línea de texto:

```
"hola mi amigo, veo que estas caminando como C3P-0"
```

y el conjunto de palabras a eliminar: "amigo", "como", "estas" y "hola".

Para el texto original:

```
"hola mi amigo, veo que estas caminando como C3P-0"
```

El texto final debería ser:

```
"mi, veo que caminando C3P-0"
```