TP 2 LPOO CADENA DE CARACTERES

Apellido y nombre: Ridolfi Ramiro

**LINK DEL REPOSITORIO: https://github.com/Ridooo11/tp-2-enet35**

1. Pedir al usuario que ingrese su nombre y luego imprimirlo en pantalla.

#include <stdio.h>

#include <string.h>

int main() {

const int MAXIMO = 50;

char nombre[MAXIMO];

printf("Ingrese su nombre: ");

fgets(nombre, sizeof(nombre), stdin);

printf("Su nombre es: %s", nombre);

}



2. Realiza un programa que cuente cuántas vocales tiene una cadena de

caracteres ingresada por el usuario.

#include <stdio.h>

#include <string.h>

int main() {

char cadena[50];

const int CANT\_VOCALES = 5;

char vocales[5] = { 'a', 'e', 'i', 'o', 'u' };

printf("Ingrese una cadena de texto: ");

fgets(cadena, sizeof(cadena), stdin);

if (cadena[strlen(cadena) - 1] == '\n') {

cadena[strlen(cadena) - 1] = '\0';

}

int i, j;

int encontrado = 0;

for (i = 0; i < strlen(cadena); i++) {

for (j = 0; j < CANT\_VOCALES; j++) {

if (cadena[i] == vocales[j]) {

printf("Se encontro la vocal %c en la cadena\n", cadena[i]);

encontrado = 1;

break;

}

}

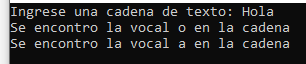
}

if (!encontrado) {

printf("No se ha encontrado ninguna vocal en la cadena: %s\n", cadena);

}

}





3. Realiza un programa que determine si una cadena de caracteres ingresada

por el usuario es palíndromo (se lee igual de izquierda a derecha que de

derecha a izquierda).

#include <stdio.h>

#include <string.h>

int main() {

char cadena[50];

char cadenaInvertida[50];

printf("Ingrese una cadena de texto: ");

fgets(cadena, sizeof(cadena), stdin);

int longitud = strlen(cadena);

int palindromo = 0;

if (cadena[longitud - 1] == '\n') {

cadena[longitud - 1] = '\0';

}

int i, j;

for (i = 0, j = longitud - 2; i < longitud; i++, j--) {

cadenaInvertida[j] = cadena[i];

}

cadenaInvertida[longitud] = '\0';

palindromo = strcmp(cadena, cadenaInvertida);

if (!palindromo) {

printf("La cadena de texto '%s' es palindroma", cadena);

} else {

printf("La cadena de texto '%s' no es palindroma", cadena);

}

}





4. Realiza un programa que reemplace todas las apariciones de un carácter

en una cadena de caracteres ingresada por el usuario por otro carácter

también ingresado por el usuario.

#include <stdio.h>

#include <string.h>

int main() {

char cadena[100];

char caracter, caracterNuevo;

int i;

int caracterExiste = 0;

printf("Ingrese una cadena de texto: ");

fgets(cadena, sizeof(cadena), stdin);

int longitud = strlen(cadena);

do {

printf("Ingrese el caracter que desea reemplazar: ");

scanf(" %c", &caracter);

for (i = 0; i < longitud; i++) {

if (cadena[i] == caracter) {

caracterExiste = 1;

break;

}

}

if (!caracterExiste) {

printf("\nEl caracter que desea reemplazar no existe en el string.\n\n");

}

} while(caracterExiste == 0);

if (caracterExiste) {

printf("Ingrese el nuevo caracter: ");

scanf(" %c", &caracterNuevo);

if (cadena[longitud - 1] == '\n') {

cadena[longitud - 1] = '\0';

}

for (i = 0; i < longitud; i++) {

if (cadena[i] == caracter) {

cadena[i] = caracterNuevo;

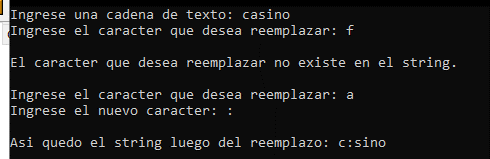
}

}

printf("\nAsi quedo el string luego del reemplazo: %s", cadena);

}

}



5. Realiza un programa que reemplace todas las apariciones de un carácter

en una cadena de caracteres ingresada por el usuario por otro carácter

también ingresado por el usuario.

6. Realiza un programa que determine si una cadena de caracteres ingresada

por el usuario es un anagrama de otra cadena también ingresada por el

usuario. Un anagrama es una palabra o frase formada por las mismas

letras de otra palabra o frase, pero en un orden diferente, como por

ejemplo roma y amor

#include <stdio.h>

#include <string.h>

int main() {

char cadena1[100], cadena2[100];

int contador[256] = {0};

int i;

printf("Ingresa la primera cadena: ");

scanf("%s", cadena1);

printf("Ingresa la segunda cadena: ");

scanf("%s", cadena2);

if (strlen(cadena1) != strlen(cadena2)) {

printf("Las cadenas no son anagramas\n");

} else {

for (i = 0; cadena1[i] && cadena2[i]; i++) {

contador[cadena1[i]]++;

contador[cadena2[i]]--;

}

for (i = 0; i < 256; i++) {

if (contador[i] != 0) {

printf("Las cadenas no son anagramas\n");

return 0;

}

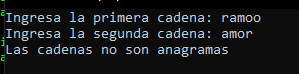
}

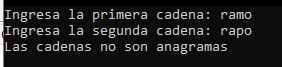
printf("Las cadenas son anagramas\n");

}

}







7. Realizar un programa que ingrese 5 palabras y devuelta la cadena mas

larga entre ellas

#include <stdio.h>

#include <string.h>

int main() {

int i, j, l, k;

const int MAX\_CADENAS = 5;

const int MAX\_LONGITUD = 100;

char cadenas[MAX\_CADENAS][MAX\_LONGITUD];

char subcadena\_mas\_larga[MAX\_LONGITUD];

subcadena\_mas\_larga[0] = '\0'; // se inicializa el string vacio. No se puede hacer en la delcaracion.

int frecuencia\_mas\_larga = 0;

printf("Ingrese las cinco cadenas de texto:\n");

for (i = 0; i < 5; i++) {

fgets(cadenas[i], sizeof(cadenas[i]), stdin);

cadenas[i][strcspn(cadenas[i], "\n")] = '\0'; // se borra el salto de linea

}

for (i = 0; i < MAX\_CADENAS; i++) { // se recorren las cadenas

int longitud = strlen(cadenas[i]);

for (j = 0; j < longitud; j++) { // se recorren los caracteres de las cadenas

for (k = j + 2; k <= longitud; k++) { // se buscan las subcadenas de al menos 2 de longitud, por eso k = j + 2

char subcadena[100];

strncpy(subcadena, cadenas[i] + j, k - j); // k -j indica la longitud de la subcadena // cadenas[i] + j indica el punto de inicio donde se empieza a copiar la sub cadena

subcadena[k - j] = '\0'; // agrega un caracter nulo para terminar la subcadena

int frecuencia = 0;

for (l = 0; l < 5; l++) {

if (strstr(cadenas[l], subcadena) != NULL) { // se busca la ocurrencia de la subcadena en las cadenas originales

frecuencia++; // si existe se aumenta la frecuencia con la que la subcadena aparece

}

}

if (frecuencia > 1 && strlen(subcadena) > strlen(subcadena\_mas\_larga)) { // se verifica que la subcadena este en mas de una cadena y si la longitud de esa sub cadena es mayor a la subcadena mas larga, se actualiza la frecuencia y la subcadena

strcpy(subcadena\_mas\_larga, subcadena);

frecuencia\_mas\_larga = frecuencia;

}

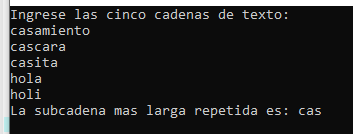
}

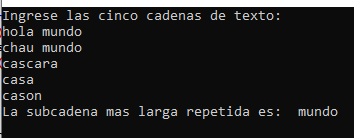
}

}

printf("La subcadena mas larga repetida es: %s\n", subcadena\_mas\_larga);

}



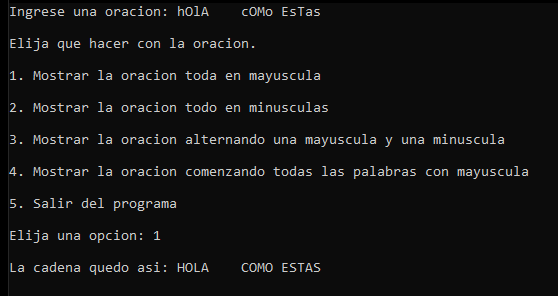


8. Realizar un programa en el cual el usuario primero ingrese una oración

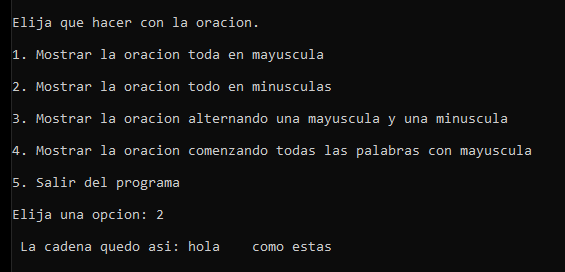
la cual puede estar en mayúscula o minúscula o de manera alternada

y luego pueda seleccionar que realice las siguientes acciones

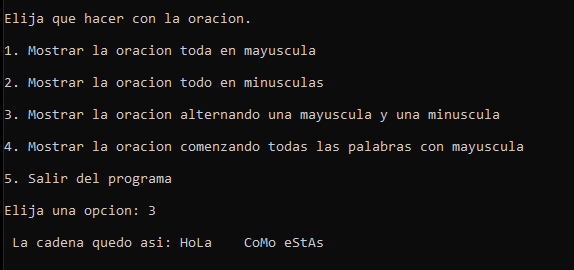
a) Mostrar la oración toda en mayúscula



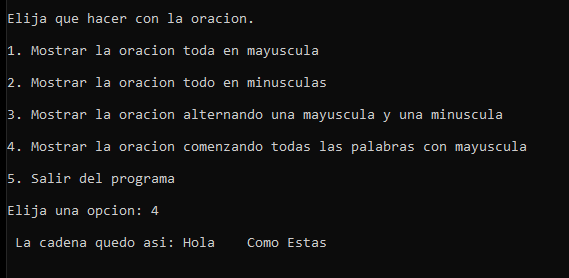
b) Mostrar la oración todo en minúsculas

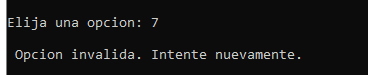


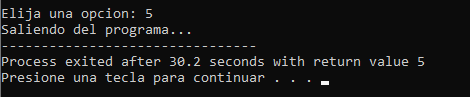
c) Mostrar la oración alternando una mayúscula y una minúscula



D) Mostrar la oración comenzando todas las palabras con mayúscula







#include <stdio.h>

#include <string.h>

#include <ctype.h>

int main() {

char cadena[100];

int opcion;

int i;

printf("Ingrese una oracion: ");

fgets(cadena, sizeof(cadena), stdin);

do {

printf("\nEliga que hacer con la oracion.\n\n");

printf("1. Mostrar la oracion toda en mayuscula\n\n");

printf("2. Mostrar la oracion todo en minusculas\n\n");

printf("3. Mostrar la oracion alternando una mayuscula y una minuscula\n\n");

printf("4. Mostrar la oracion comenzando todas las palabras con mayuscula\n\n");

printf("5. Salir del programa\n\n");

printf("Elija una opcion: ");

scanf("%d", &opcion);

switch(opcion) {

case 1:

for (i = 0; i < strlen(cadena); i++) {

cadena[i] = toupper(cadena[i]);

}

printf("\nLa cadena quedo asi: %s\n", cadena);

break;

case 2:

for (i = 0; i < strlen(cadena); i++) {

cadena[i] = tolower(cadena[i]);

}

printf("\nLa cadena quedo asi: %s\n", cadena);

break;

case 3:

for (i = 0; i < strlen(cadena); i++) {

if (i % 2 == 0) {

cadena[i] = toupper(cadena[i]);

} else {

cadena[i] = tolower(cadena[i]);

}

}

printf("\nLa cadena quedo asi: %s\n", cadena);

break;

case 4:

for (i = 0; i < strlen(cadena); i++) {

if (i == 0 || cadena[i - 1] == ' ') {

cadena[i] = toupper(cadena[i]);

} else {

cadena[i] = tolower(cadena[i]);

}

}

printf("\nLa cadena quedo asi: %s\n", cadena);

break;

case 5:

printf("\nSaliendo del programa...");

break;

default:

printf("\nOpcion invalida. Intente nuevamente.\n");

break;

}

} while(opcion != 5);

}