



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Ingeniero en computación

Ingeniero en Software y tecnologías emergentes

Materia: Programación Estructurada / Clave 36276

Alumno: Teresa Rivas Gómez

Matrícula: 372565

Maestro: Pedro Núñez Yépiz

Actividad No. : 7

Tema - Unidad : Cadenas

Ensenada, Baja California a 25 de Septiembre del 2023



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Introducción:

En esta actividad abarcamos el uso de las cadenas y los diferentes usos que le podemos dar, para que funciona y un poco de la historia de lo que es la cadena como el método ANCII y como hoy en día ya existen muchas librerías que te facilitan el trabajo a la hora de realizarlas, pero que en nuestro caso las estaremos ejecutando de la forma clásica.

Estas nos pueden ayudar mucho a la hora de realizar diversos programas, pueden optimizarlos mejor dependiendo el caso y nos harán de gran utilidad en nuestras próximas actividades.

Competencias:

Que nosotros sepamos usar las cadenas y sus componentes, saberlas declarar correctamente y sobre todo saber cuando podemos hacer uso de ellas y cuando no, lo que hacen dependiendo del ciclo en el que las pongamos y como determinar cual es un buen tamaño para nuestras cadenas dependiendo del programa que queramos realizar

Se espera que podamos tener un dominio de este tema y que con la practica podamos desarrollar programas complejos que utilicen todas estas funciones sin necesidad de recurrir a internet o libros, que utilicemos la razón y la lógica y que nuestros programas queden mejor estructurados.



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Documentación:

Para la teoría me estuve apoyando de diversas paginas en internet, de la inteligencia artificial y de los videos que publico en Blackboard para la introducción de las cadenas y como usarlas. También me ayudaron mucho todos los enlaces a las paginas que recomendó en la actividad 7 en blackboard.

Procedimiento

ACTIVIDAD 7

INSTRUCCIONES

- 1.- Realiza los ejercicios en C
- 2.- Una vez los ejercicios terminados y con los nombres correctos como se deben nombrar, realiza captura de pantalla del código y salida, pegar en un archivo de Word (**INCLUIR PORTADA**)
- 3.- Realiza el **reporte de práctica**, Realiza archivo PDF nombrarlo con tus iniciales_PE_RP04.PDF
- 4.-convierte el archivo word con todo y portada en archivo PDF (será el anexo del reporte de practica)
- 5.- Sube a blackboard los ejercicios hechos en C (**2 PROGRAMAS PARTE1 Y PARTE2**)
- 6.- Sube a Blackboard los archivos PDF (reporte de práctica, anexo y actividad)

ACTIVIDAD 7

PARTE 1



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

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 ENSENADA	SALIDA 2 ADANESNE	SALIDA 3 E N S E N A D A
SALIDA 4 A D A N E S N E	SALIDA 5 ENSENADA ENSENAD ENSENA ENSEN ENSE ENS EN E	SALIDA 6 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
--	--	-------------------------------

ACTIVIDAD 7

PARTE 2

REALIZA LAS SIGUIENTES FUNCIONES:

- 1.- Función que reciba como parámetro una cadena y la convierta a MAYUSCULAS
 - 2.- Función Que reciba como parámetro una cadena y la convierta a MINUSCULAS
 - 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 retorne una cadena con sus caracteres acomodados de forma inversa (al reves)
 - 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 permita 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 MAYUSCULAS, MINUSCULAS , CAPITAL, SIN ESPACIOS, ALREVES la cadena original.
 - 9.-Función que reciba como parámetro una cadena, y desplegar la leyenda si la cadena es un palíndromo SI o NO



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Resultados y Conclusiones:

Parte 1:

- Declaracion de funciones:

```
// Librerias

#include <stdio.h>

#include <stdlib.h>

// Menu Principal

int msges();

void menu();

//Funcion para la pedir de datos

void datos(char nombre[]);

// Validacion

int validar(char mensj[], int ri, int rf);

// Declarar funciones

//Ejercicios de la Parte 1

void salida1(char nombre[]);

void salida2(char nombre[]);

void salida3(char nombre[]);

void salida4(char nombre[]);

void salida5(char nombre[]);

void salida6(char nombre[]);

void salida7(char nombre[]);

void salida8(char nombre[]);

void salida9(char nombre[]);

void salida10(char nombre[]);
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

● Menu

```
// Funcion main

int main()

{

    menu();

    return 0;

}
```

● Mensajes

```
// Menu que se muestra al usuario

int msges()

{

    int op;

    system ("CLS");

    printf("\n MENU PARA LAS DIFERENTES SALIDAS DE LA CADENA \n");

    printf("\n OPCIONES: \n");

    printf("1.- Salida 1: MAYUSCULAS\n");

    printf("2.- Salida 2: MAYUSCULAS AL REVES\n");

    printf("3.- Salida 3: UNA LETRA POR RENGLON\n");

    printf("4.- Salida 4: UNA LETRA POR RENGLON AL REVES\n");

    printf("5.- Salida 5: LA PALABRA MENOS UNA LETRA\n");

    printf("6.- Salida 6: LA PALABRA MENOS UNA LETRA AL REVES\n");

    printf("7.- Salida 7: LA PALABRA MENOS UNA LETRA DEL INICIO\n");

    printf("8.- Salida 8: LA PALABRA MENOS UNA LETRA DEL INICIO AL REVES\n");

    printf("9.- Salida 9: SOLO CONSONANTES\n");

    printf("10.- Salida 10: SOLO VOCALES\n");

    printf("0.- SALIR  \n");

}
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
op = validar ("ESCOGE UNA OPCION: ", 0, 10);  
  
return op;  
  
}
```

● Casos depende del ejercicio

```
// Casos depende la seleccion del usuario
```

```
void menu()  
{  
  
    system ("CLS");  
  
    int op;  
  
    char nombre[50];  
  
    do {  
  
        op = msges();  
  
        switch(op)  
        {  
  
            case 1:  
  
                datos(nombre);  
  
                salida1(nombre);  
  
                break;  
  
            case 2:  
  
                datos(nombre);  
  
                salida2(nombre);  
  
                break;  
  
            case 3:  
  
                datos(nombre);  
  
                salida3(nombre);  
  
                break;  
  
            case 4:
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
        datos(nombre);

        salida4(nombre);

        break;

case 5:

        datos(nombre);

        salida5(nombre);

        break;

case 6:

        datos(nombre);

        salida6(nombre);

        break;

case 7:

        datos(nombre);

        salida7(nombre);

        break;

case 8:

        datos(nombre);

        salida8(nombre);

        break;

case 9:

        datos(nombre);

        salida9(nombre);

        break;

case 10:

        datos(nombre);

        salida10(nombre);

        break;

case 0:

        printf("Saliste del programa.\n");
```




Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
        break;

        default:

            printf("Opcion Invalida.\n");

            break;

    }

} while(op != 0);

system("PAUSE");

}
```

● Validación:

```
// Validacion

int validar(char mensj[], int ri, int rf)
{

    int num;

    // Cadena que va a leer el mensaje ingresado

    char cadena[50];

    // Proceso y desarrollo

    do

    {

        printf("%s", mensj);

        // Borrar basura

        fflush(stdin);

        gets(cadena);

        num = atoi(cadena);

    } while (num < ri || num > rf);

    // Retorna el valor que num, entre los rangos dados

    return num;

}
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

- **Pedir datos:**

```
//Pedir datos

void datos(char nombre[])

{

    system ("CLS");

    printf("Introduzca su nombre: ");

    // Borrar basura

    fflush(stdin);

    gets(nombre);

}
```

- **Salida 1:**

```
// SALIDA 1

void salida1(char nombre[])

{

    system ("CLS");

    //Imprimir nombre de la salida

    printf("Salida 1: MAYUSCULAS\n");

    // Recorre todos los caracteres uno por uno

    for (int i = 0; nombre[i] != '\0'; i++)

    {

        // Comprobar si estan en minusculas

        if (nombre[i] >= 'a')

        {

            if (nombre[i] <= 'z')

            {

                // Si las condiciones son verdaderas se hace la conversion a mayusculas

                nombre[i] = nombre[i] - ('a' - 'A');

            }

        }

    }

}
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
    }

    }

    // Imprime la palabra en mayusculas o igual
    printf("%c", nombre[i]);

}

printf("\n");

system("PAUSE");

}
```

● Salida 2:

```
// SALIDA 2

void salida2(char nombre[])

{

    system("CLS");

    //Imprimir nombre de la salida

    printf("Salida 2: MAYUSCULAS AL REVES\n");

    // Declarar variable para la longitud/largo

    int length = 0;

    // Calcular

    while (nombre[length] != '\0')

    {

        length++;

    }

    // Recorre todos los caracteres uno por uno

    for (int i = length - 1; i >= 0; i--)

    {

        // Comprobar si estan en minusculas

        if (nombre[i] >= 'a')
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
{  
  
    if (nombre[i] <= 'z')  
  
    {  
  
        // Si las condiciones son verdaderas se hace la conversion a mayusculas  
  
        nombre[i] = nombre[i] - ('a' - 'A');  
  
    }  
  
}  
  
// Imprime la palabra en mayusculas o igual pero al reves  
  
printf("%c", nombre[i]);  
  
}  
  
printf("\n");  
  
system("PAUSE");  
  
}
```

● Salida 3:

```
// SALIDA 3  
  
void salida3(char nombre[])  
  
{  
  
    system ("CLS");  
  
    // Imprimir nombre de la salida  
  
    printf("Salida 3: UNA LETRA POR RENGLON\n");  
  
    // Recorre todos los caracteres uno por uno  
  
    for (int i = 0; nombre[i] != '\0'; i++)  
  
    {  
  
        if (nombre[i] >= 'a')  
  
        {  
  
            if (nombre[i] <= 'z')  
  
            {
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
        nombre[i] = nombre[i] - ('a' - 'A');

    }

}

// Imprime la cadena y al usar \n podemos sacar una por renglon

printf("%c\n", nombre[i]);

}

// Para que la palabra se imprima en cada linea

printf("\n");

system("PAUSE");

}
```

● Salida 4:

```
// SALIDA 4

void salida4(char nombre[])
{
    system ("CLS");

    // Imprimir nombre de la salida

    printf("Salida 4: UNA LETRA POR RENGLON AL REVES\n");

    // Declarar variable para la longitud/largo

    int length = 0;

    // Calcular

    while (nombre[length] != '\0')
    {
        length++;
    }

    // Recorre todos los caracteres uno por uno y hace que salga al revés la palabra

    for (int i = length - 1; i >= 0; i--)
    {
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
    if (nombre[i] >= 'a')
    {
        if (nombre[i] <= 'z')
        {
            nombre[i] = nombre[i] - ('a' - 'A');
        }
    }

    // Imprime la cadena y al usar \n podemos sacar una por renglon
    printf("%c\n", nombre[i]);
}

// Para que la palabra se imprima en cada linea
printf("\n");

system("PAUSE");
}
```

● Salida 5:

```
void salida5(char nombre[])
{
    system("CLS");

    // Imprimir nombre de la salida
    printf("Salida 5: LA PALABRA MENOS UNA LETRA POR RENGLON\n");

    // Declarar variable para la longitud/largo
    int length = 0;

    // Calcular
    while (nombre[length] != '\0')
    {
        length++;
    }
}
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
// La longitud de la palabra y va disminuyendo hasta llegar a 1

for (int i = length; i >= 1; i--)

{

    // Recorre cada letra y convierte la letra actual a mayúsculas

    for (int j = 0; j < i; j++)

    {

        if (nombre[j] >= 'a')

        {

            if (nombre[j] <= 'z')

            {

                nombre[j] = nombre[j] - ('a' - 'A');

            }

        }

    }

    // Imprimie la palabra

    for (int j = 0; j < i; j++)

    {

        printf("%c", nombre[j]);

    }

    // Para que la palabra se imprima en cada linea

    printf("\n");

}

printf("\n");

system("PAUSE");

}
```

● Salida 6:

```
// SALIDA 6
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
void salida6(char nombre[])
{
    system ("CLS");

    // Imprimir nombre de la salida

    printf("Salida 6: LA PALABRA MENOS UNA LETRA AL REVES\n");

    // Declarar variable para la longitud/largo

    int length = 0;

    // Calcular

    while (nombre[length] != '\0')

    {
        length++;
    }

    // Declarar la copia y asignarle un tamaño

    char copia_nombre[50];

    // Copiar el nombre a la copia

    for (int i = 0; i < length; i++)

    {
        copia_nombre[i] = nombre[i];
    }

    // Que el ultimo caracter sea nulo

    copia_nombre[length] = '\0';

    // Muestra la cadena gradualmente

    while (length > 1)

    {
        // Imprimir la cadena copiada en mayusculas al reves

        for (int i = length - 1; i >= 0; i--)

        {
            if (copia_nombre[i] >= 'a')

            {
```




Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
        if (copia_nombre[i] <= 'z')
        {
            copia_nombre[i] = copia_nombre[i] - ('a' - 'A');
        }
    }

    printf("%c", copia_nombre[i]);
}

// Para pasar a la siguiente linea
printf("\n");

// Acortar la cadena copiada eliminando el primer carácter
for (int i = 0; i < length - 1; i++)
{
    copia_nombre[i] = copia_nombre[i + 1];
}

// Para que el nuevo ultimo caracter si sea el nulo
copia_nombre[length - 1] = '\0';

// Para que se vea que la cadena ya no tiene un caracter
length--;
}

system("PAUSE");
}
```

● Salida 7:

```
// SALIDA 7

void salida7(char nombre[])
{
    system ("CLS");

    // Imprimir nombre de la salida
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
printf("Salida 7: LA PALABRA MENOS UNA LETRA DEL INICIO\n");

// Declarar variable para la longitud/largo

int length = 0;

// Calcular

while (nombre[length] != '\0')

{

    length++;

}

// Declarar la copia y asignarle un tamaño

char copia_nombre[50];

// Copiar el nombre a la copia

for (int i = 0; i < length; i++)

{

    copia_nombre[i] = nombre[i];

}

// Que el ultimo caracter sea nulo

copia_nombre[length] = '\0';

// Itera y muestra la cadena gradualmente

while (length > 1)

{

    // Imprimir la cadena copiada en mayúsculas

    for (int i = 0; i < length; i++)

    {

        if (copia_nombre[i] >= 'a' && copia_nombre[i] <= 'z')

        {

            copia_nombre[i] = copia_nombre[i] - ('a' - 'A');

        }

        printf("%c", copia_nombre[i]);

    }

}
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
printf("\n");

// Acortar la cadena copiada eliminando el primer carácter

for (int i = 0; i < length - 1; i++)

{

    copia_nombre[i] = copia_nombre[i + 1];

}

// Para que el nuevo ultimo caracter si sea el nulo

copia_nombre[length - 1] = '\0';

// Para que se vea que la cadena ya no tiene un caracter

length--;

}

printf("\n");

system("PAUSE");

}
```

● Salida 8:

```
// SALIDA 8

void salida8(char nombre[])

{

    system ("CLS");

    // Imprimir nombre de la salida

    printf("Salida 8: LA PALABRA MENOS UNA LETRA DEL INICIO AL REVES\n");

    // Declarar variable para la longitud/largo

    int length = 0;

    // Calcular

    while (nombre[length] != '\0')

    {

        length++;

    }

}
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
}

// Leer una cadena al revés hasta llegar al primer elemento que es el 0

for (int i = length - 1; i >= 0; i--)

{

    // Convertir la letra actual a mayúsculas

    for (int j = i; j >= 0; j--)

    {

        if (nombre[j] >= 'a' && nombre[j] <= 'z')

        {

            nombre[j] = nombre[j] - ('a' - 'A');

        }

    }

    // Imprimir la palabra actual al revés

    for (int j = i; j >= 0; j--)

    {

        printf("%c", nombre[j]);

    }

    // Para que la palabra se imprima en cada linea

    printf("\n");

}

system("PAUSE");

}
```

● Salida 9:

```
// SALIDA 9

void salida9(char nombre[])

{

    system ("CLS");
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
// Imprimir nombre de la salida

printf("Salida 9: SOLO CONSONANTES\n");

// Imprimir el nombre en mayúsculas
for (int i = 0; nombre[i] != '\0'; i++)
{
    if (nombre[i] >= 'a')
    {
        if(nombre[i] <= 'z')
        {
            nombre[i] = nombre[i] - ('a' - 'A');
        }
    }
}

// Imprimir solo las vocales
for (int i = 0; nombre[i] != '\0'; i++)
{
    if ( nombre[i] != 'A')
    {
        if (nombre[i] != 'E')
        {
            if ( nombre[i] != 'I')
            {
                if ( nombre[i] != 'O')
                {
                    if ( nombre[i] != 'U')
                    {
                        printf ( "%c", nombre[i] );
                    }
                }
            }
        }
    }
}
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
    }  
    }  
    }  
}  
  
printf("\n");  
  
system("PAUSE");  
}
```

● Salida 10:

```
// SALIDA 10  
  
void salida10(char nombre[])  
{  
    system ("CLS");  
  
    //Imprimir nombre de la salida  
    printf("Salida 10: SOLO VOALES\n");  
  
    // Imprimir el nombre en mayúsculas  
    for (int i = 0; nombre[i] != '\0'; i++)  
    {  
        if (nombre[i] >= 'a')  
        {  
            if(nombre[i] <= 'z')  
            {  
                nombre[i] = nombre[i] - ('a' - 'A');  
            }  
        }  
    }  
}  
  
// Imprimir solo las vocales  
for (int i = 0; nombre[i] != '\0'; i++)
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
{  
  
    if ( nombre[i] == 'A')  
  
    {  
  
        printf ( "%c", nombre[i] );  
  
    }  
  
    if (nombre[i] == 'E')  
  
    {  
  
        printf ( "%c", nombre[i] );  
  
    }  
  
    if ( nombre[i] == 'I')  
  
    {  
  
        printf ( "%c", nombre[i] );  
  
    }  
  
    if ( nombre[i] == 'O')  
  
    {  
  
        printf ( "%c", nombre[i] );  
  
    }  
  
    if ( nombre[i] == 'U')  
  
    {  
  
        printf ( "%c", nombre[i] );  
  
    }  
  
}  
  
printf("\n");  
  
system("PAUSE");  
  
}
```

Resultados y Conclusiones:

Parte 2:



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

● Declaracion de funciones

```
// Menu Principal
int msges();
void menu();

// Validacion:
// Declaracion de la función es_numero
int es_numero(char caracter);
// Declaracion de la función es_espacio
int es_espacio(char caracter);
// Función de validacion
int validarCadena(const char *cadena);
// Validar la cadena
void validar(char cadena[]);
//Funcion para la pedir de datos
void datos(char cadena[]);
// Ejercicios de la Parte 2
void ej1_pt2(char cadena[]);
void ej2_pt2(char cadena[]);
void ej3_pt2(char cadena[]);
int ej4_pt2(char cadena[]);
int ej5_pt2(char cadena[]);
void ej6_pt2(char cadena[]);
int ej7_pt2(char cadena[]);
void ej8_pt2(char cadena[]);
void ej9_pt2(char cadena[]);
```

● Menu

```
// Funcion main

int main()

{

    menu();

    return 0;

}
```

● Mensajes

```
// Menu que se muestra al usuario

int msges()

{
```




Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
int op;

system ("CLS");

printf ("\n MENU DE LA ACTIVIDAD 7 \n");

printf ("\n PARTE 2 \n");

printf("1.- Mayusculas \n");

printf("2.- Minusculas \n");

printf("3.- Capital \n");

printf("4.- Cantidad \n");

printf("5.- Inversa \n");

printf("6.- Sin espacios \n");

printf("7.- Alfabeto \n");

printf("8.- Todo \n");

printf("9.- Palindromo \n");

printf("0.- SALIR \n");

printf("ESCOGE UNA OPCION: ");

scanf ("%d",&op);

return op;

}
```

● Casos

```
// Casos depende la seleccion del usuario

void menu()

{

    int op;

    char cadena[50];

    do{

        op=msges();
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
switch (op)

{

    case 1:

        datos(cadena);

        ej1_pt2(cadena);

        break;

    case 2:

        datos(cadena);

        ej2_pt2(cadena);

        break;

    case 3:

        datos(cadena);

        ej3_pt2(cadena);

        break;

    case 4:

        datos(cadena);

        ej4_pt2(cadena);

        break;

    case 5:

        datos(cadena);

        ej5_pt2(cadena);

        break;

    case 6:

        datos(cadena);

        ej6_pt2(cadena);

        break;

    case 7:

        datos(cadena);

        ej7_pt2(cadena);
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
        break;

    case 8:

        datos(cadena);

        ej8_pt2(cadena);

        break;

    case 9:

        datos(cadena);

        ej9_pt2(cadena);

        break;

    case 0:

        printf("Saliste del programa.\n");

        break;

    default:

        printf("Opcion Invalida.\n");

        break;

}

}while (op != 0);

}
```

● Validación

```
// Validacion
int validarCadena(const char cadena[])
{
    // Declarar variables e inicializarlas en 0
    int contieneNumero = 0;
    int contieneDobleEspacio = 0;
    int contieneMinuscula = 0;
    // Recorre cada caracter hasta encontrar el nulo
    for (int i = 0; cadena[i] != '\0'; i++)
    {
        char caracter = cadena[i];
        // Si encuentra que tiene un numero se establece en 1 para indicarlo
        if (caracter >= '0')
        {
            if(caracter <= '9')
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
        {
            contieneNumero = 1;
        }
    }
    if (caracter == ' ')
    {
        if (cadena[i + 1] == ' ')
        {
            contieneDobleEspacio = 1;
        }
    }
    if (caracter >= 'a')
    {
        if(caracter <= 'z')
        {
            contieneMinuscula = 1;
        }
    }
}
// Verifica si alguna de las dos se establecio en 1
if (contieneNumero || contieneDobleEspacio || contieneMinuscula)
{
    // La cadena no cumple con los requisitos
    return 0;
}
else
{
    // La cadena cumple con los requisitos
    return 1;
}
}

int es_numero(char caracter)
{
    if (caracter >= '0')
    {
        if (caracter <= '9')
        {
            return 1; // Es numero
        }
    }
    return 0; // No es numero
}

int es_espacio(char caracter)
{
    return (caracter == ' ');
}

int es_minuscula(char caracter)
{
    if (caracter >= 'a')
    {
        if (caracter <= 'z')
        {
            return 1; // Es minuscula
        }
    }
    return 0; // No es minuscula
}
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
        return 1; // Es minuscula
    }
}
return 0; // No es minuscula
}
//Pedir datos
void datos(char cadena[])
{
    system ("CLS");
    printf( "Introduzca una palabra: " );
    fflush(stdin);
    gets(cadena);
}
```

● Pedir datos

```
//Pedir datos
void datos(char cadena[])
{
    system ("CLS");
    printf( "Introduzca una palabra: " );
    fflush(stdin);
    gets(cadena);
}
```

● 1 MAYUSCULAS

```
// PARTE 2
// Ejercicio 1 Parte 2
// Función que reciba como parámetro una cadena y la convierta a MAYUSCULAS
void ej1_pt2(char cadena[])
{
    system ("CLS");
    // Titulo
    printf("CONVERTIR A MAYUSCULAS\n");
    // Recorre todos los caracteres uno por uno
    for (int i = 0; cadena[i] != '\0'; i++)
    {
        // Comprobar si estan en minusculas
        if (cadena[i] >= 'a')
        {
            if (cadena[i] <= 'z')
            {
                // Si las condiciones son verdaderas se hace la conversion a mayusculas
                cadena[i] = cadena[i] - ('a' - 'A');
            }
        }
    }
}
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
    }  
}  
// Imprime la palabra en mayusculas o igual  
printf("%c", cadena[i]);  
}  
printf("\n");  
getchar();  
}
```

● 2 MINUSCULAS

```
// Ejercicio 2 Parte 2  
// Función Que reciba como parámetro una cadena y la convierta a MINUSCULAS  
void ej2_pt2(char cadena[])  
{  
    system("CLS");  
    // Titulo  
    printf("CONVERTIR A MINUSCULAS\n");  
    // Recorre todos los caracteres uno por uno  
    for (int i = 0; cadena[i] != '\0'; i++)  
    {  
        // Comprobar si estan en mayusculas  
        if (cadena[i] >= 'A')  
        {  
            if (cadena[i] <= 'Z')  
            {  
                // Si las condiciones son verdaderas se hace la conversion a minusculas  
                cadena[i] = cadena[i] - ('A' - 'a');  
            }  
        }  
    }  
    // Imprime la palabra en mayusculas o igual  
    printf("%c", cadena[i]);  
}  
printf("\n");  
getchar();  
}
```

● 3 CAPITAL

```
// Ejercicio 3 Parte 2  
// Función que reciba como parámetro una cadena y la convierta a CAPITAL  
void ej3_pt2(char cadena[])  
{  
    system("CLS");  
    // Titulo  
    printf("CONVERTIR A CAPITAL\n");  
    // Bandera para indicar que estamos en la primera letra
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
int primera_letra = 1;
// Recorre todos los caracteres uno por uno
for (int i = 0; cadena[i] != '\0'; i++)
{
    // Comprobar si estan en minusculas
    if (cadena[i] >= 'a')
    {
        if (cadena[i] <= 'z')
        {
            // Si es la primera letra
            if (primera_letra)
            {
                // Convierte en mayUscula
                cadena[i] = cadena[i] - ('a' - 'A');
                // Ya no estamos en la primera letra
                primera_letra = 0;
            }
        }
    }
    else
    {
        if(cadena[i] == ' ')
        {
            primera_letra = 1;
        }
    }
    // Imprime la palabra en mayusculas o igual
    printf("%c", cadena[i]);
}

printf("\n");
getchar();
}
```

● 4 LONGITUD

```
// Ejercicio 4 Parte 2
// Función que reciba como parámetro una cadena y retorne la cantidad de
// caracteres que tiene la cadena.
int ej4_pt2(char cadena[])
{
    system("CLS");
    // Titulo
    printf("CANTIDAD DE CARACTERES\n");
    // Declarar variable para la longitud/largo
    int length = 0;
    // Calcular
    while (cadena[length] != '\0')
    {
        // Incremento
    }
}
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
        length++;
    }
    // Imprimir tamaño de la cadena
    printf("Su tamaño es de: %d\n", length);
    getchar();
    return 0;
}
```

● 5 INVERSA

```
// Ejercicio 5 Parte 2
// Función que reciba como parámetro una cadena y retorne una cadena con sus
// caracteres acomodados de forma inversa (al revés)
int ej5_pt2(char cadena[])
{
    system("CLS");
    // Título
    printf("CADENA INVERSA\n");
    // Declarar variable para la longitud/largo
    int length = 0;
    // Calcular
    while (cadena[length] != '\0')
    {
        // Incrementa longitud
        length++;
    }
    // Recorre todos los caracteres uno por uno
    for (int i = length - 1; i >= 0; i--)
    {
        // Imprime la cadena al revés
        printf("%c", cadena[i]);
    }
    printf("\n");
    getchar();
    return 0;
}
```

● 6 SIN ESPACIOS

```
// Ejercicio 6 Parte 2
// Función que reciba como parámetro una cadena y genere una nueva cadena
// basada en la original pero sin espacios.
void ej6_pt2(char cadena[])
{
    system("CLS");
    // Título
    printf("SIN ESPACIOS\n");
```




Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
// Declarar una nueva cadena para almacenar el resultado sin espacios
char nuevaCadena[50];
// Inicializar la longitud de la nueva cadena
int nuevaLongitud = 0;
// Recorre la cadena original
for (int i = 0; cadena[i] != '\0'; i++)
{
    if (cadena[i] != ' ')
    {
        // Copia los caracteres que no sean espacios a la nueva cadena
        nuevaCadena[nuevaLongitud] = cadena[i];
        nuevaLongitud++;
    }
}
// Agrega el caracter nulo al final de la nueva cadena para finalizarla
nuevaCadena[nuevaLongitud] = '\0';
// Imprime la nueva cadena sin espacios
printf("%s\n", nuevaCadena);
getchar();
}
```

● 7 ALFABETO

```
// Ejercicio 7 Parte 2
// Función que sirva para leer una cadena y solo permita 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.
int ej7_pt2(char cadena[])
{
    system("CLS");
    // Declarar variables booleanas para seguir el estado de la cadena
    bool tieneEspaciosAlPrincipio = false;
    bool terminaConEspacios = false;
    bool tieneNumeros = false;
    int i = 0;
    while (cadena[i] != '\0')
    {
        // Para los espacios
        if (cadena[i] == ' ')
        {
            if (i == 0)
            {
                tieneEspaciosAlPrincipio = true; // Tiene espacios al principio
            }
            if (cadena[i + 1] == '\0')
            {
                terminaConEspacios = true; // Termina con espacios
            }
        }
        // Para los numeros
        if (cadena[i] >= '0' && cadena[i] <= '9')
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
{
    tieneNumeros = true; // Tiene numeros
}
i++;
}
// Comprobar los indicadores para determinar si la cadena es válida
if (tieneEspaciosAlPrincipio)
{
    printf("No es valida (Tiene espacios al principio)\n");
}
if (terminaConEspacios)
{
    printf("No es valida (Termina con espacios)\n");
}
if (tieneNumeros)
{
    printf("No es valida (Tiene numeros)\n");
}
if (!tieneEspaciosAlPrincipio && !terminaConEspacios && !tieneNumeros)
{
    printf("Es valida\n");
}
else
{
    printf("No es valida\n");
}
getchar();
return 0;
}
```

● 8 TODO EN 1

```
// Ejercicio 8 Parte 2
// Función que reciba como parámetro una cadena, y utilizando las funciones anteriores,
// imprima en MAYUSCULAS, MINUSCULAS , CAPITAL, SIN ESPACIOS, ALREVES la cadena original.
void ej8_pt2(char cadena[])
{
    system ("CLS");
    // Titulo
    printf("TODO EN 1\n");
    // Mandar a llamar a las funciones
    ej1_pt2(cadena); // Convertir a MAYUSCULAS
    ej2_pt2(cadena); // Convertir a MINUSCULAS
    ej3_pt2(cadena); // Convertir a CAPITAL
    ej5_pt2(cadena); // Imprimir AL REVES
    ej6_pt2(cadena); // Imprimir SIN ESPACIOS
    getchar();
}
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

● 9 PALINDROMO

```
// Ejercicio 9 Parte 2
// Función que reciba como parámetro una cadena, y desplegar la leyenda si la cadena es
// un palíndromo SI o NO
void ej9_pt2(char cadena[])
{
    system ("CLS");
    // Titulo
    printf("PALINDROMO\n");
    // Llamar a mi funcion de validacon
    validarCadena(cadena);
    validar(cadena);
    getchar();
}
```