Universidad Autónoma deBaja California

Facultad de Ingeniería, Arquitectura yDiseño

Ingeniero en Computación

Asignatura:

Programación Estructurada

Actividad 7:

Brayan Arturo Rocha Meneses

Matricula:

371049

Ensenada Baja California 25 de Septiembre del 2023

Introducción:

Los dos programas que se presentan son programas en lenguaje C que realizan diferentes operaciones en cadenas de caracteres. El primer programa se encarga de manipular una cadena de texto ingresada por el usuario, ofreciendo opciones como convertir a mayúsculas, invertir la cadena, mostrar letras una por una, entre otras. El segundo programa también trabaja con una cadena ingresada por el usuario y ofrece operaciones como convertir a mayúsculas, minúsculas, capitalizar, verificar si es un palíndromo, entre otras.

A continuación, se describirán las funciones y operaciones de ambos programas.

Programa 1:

Este programa en C se centra en la manipulación de una cadena de texto ingresada por el usuario.

1. msges(): Esta función muestra un menú que presenta 10 opciones diferentes para operar en la cadena de texto. El usuario elige una opción y se devuelve el número correspondiente.

2. menu(): Esta función contiene un bucle que permite al usuario seleccionar una operación del menú y ejecutarla en la cadena de texto ingresada.

3. validar(): Se utiliza para validar la entrada del usuario, asegurándose de que el número ingresado esté dentro de un rango específico.

4. datos(): Esta función solicita al usuario que ingrese una cadena de texto y la almacena en un arreglo.

5. Las funciones `salida1()` a `salida10()` realizan las siguientes operaciones en la cadena de texto:

   salida1(): Convierte la cadena a mayúsculas.

   salida2(): Convierte la cadena a mayúsculas y la muestra en orden inverso.

   salida3(): Muestra cada letra de la cadena en una línea separada.

   salida4(): Muestra cada letra de la cadena en orden inverso en líneas separadas.

salida5(): Muestra la cadena con letras eliminadas una por una desde el final.

salida6(): Muestra la cadena con letras eliminadas una por una desde el final, en orden inverso.

salida7(): Muestra la cadena con letras eliminadas una por una desde el inicio.

salida8(): Muestra la cadena con letras eliminadas una por una desde el inicio, en orden inverso.

salida9(): Muestra solo las consonantes de la cadena.

salida10(): Muestra solo las vocales de la cadena.

Programa 2:

Este programa también trabaja con una cadena de texto ingresada por el usuario y ofrece operaciones de formato y análisis.

1. msges(): Muestra un menú con 10 opciones diferentes para operar en la cadena de texto y devuelve la elección del usuario.

2. menu(): Contiene un bucle que permite al usuario seleccionar una operación del menú y ejecutarla en la cadena de texto ingresada.

3. validarCadena(): Verifica si la cadena de texto cumple con ciertos requisitos, como no contener números ni dobles espacios.

4. datos(): Solicita al usuario que ingrese una cadena de texto y la almacena en un arreglo.

5. Las funciones `ej1_pt2()` a `ej9_pt2()` realizan las siguientes operaciones en la cadena de texto:

ej1_pt2(): Convierte la cadena a mayúsculas.

ej2_pt2(): Convierte la cadena a minúsculas.

ej3_pt2(): Capitaliza la cadena (la primera letra de cada palabra en mayúscula).

ej4_pt2(): (Pendiente de implementación)

ej5_pt2(): (Pendiente de implementación)

ej6_pt2(): (Pendiente de implementación)

ej7_pt2(): (Pendiente de implementación)

ej8_pt2(): (Pendiente de implementación)

ej9_pt2(): Verifica si la cadena es un palíndromo (se lee igual de izquierda a derecha que de derecha a izquierda).

Ambos programas proporcionan una variedad de operaciones para manipular y analizar cadenas de texto, permitiendo al usuario realizar diversas tareas de procesamiento de texto.

Competencia:

Diseñar y construir funciones recursivas, para utilizar las ventajas de la programación modular en la solución de problemas de procesamiento de información, con actitud propositiva y organizada.

Fundamentos:

7.1 deficición y uso Estructuras

En C una estructura es una colección de variables que se referencian bajo el mismo nombre. Una estructura

proporciona un medio conveniente para mantener junta información que se relaciona. Una definición de estructura

forma una plantilla que se puede usar para crear variables de estructura. Las variables que forman la estructura son

llamados elementos estructurados.

Generalmente, todos los elementos en la estructura están relacionados lógicamente unos con otros. Por ejemplo, se

puede representar una lista de nombres de correo en una estructura. Mediante la palabra clave struct se le indica

al compilador que defina una plantilla de estructura.

```
struct direc
{
char nombre[30];
char calle[40];
char ciudad[20];
char estado[3];
unsigned int codigo;
};
```

Con el trozo de código anterior no ha sido declarada ninguna variable, tan sólo se ha definido el formato. Para declarar

una variable, se hará como sigue:

```
struct direc info_direc;
```

Se pueden declarar una o más variables cuando se define una estructura entre) y ;. Por ejemplo:

```
struct direc
{
char nombre[30];
char calle[40];
char ciudad[20];
char estado[3];
unsigned int codigo;
} info_direc, binfo, cinfo;
```

observar que direc es una etiqueta para la estructura que sirve como una forma breve para futuras declaraciones.

Como en este última declaración se indican las variables con esta estructura, se puede omitir el nombre de la

estructura tipo.

Las estructuras pueden ser también pre inicializadas en la declaración:

```
struct direc info_direc={"Vicente Fernandez","Fantasia
2000","Dorado","MMX",12345};
```

Para referenciar o accesar un miembro (o campo) de una estructura, C proporciona el operador punto ., por ejemplo,

para asignar a info_direc otro código, lo hacemos como:

```
info_direc.codigo=54321;
```

7.2 Definición de nuevos tipos de datos

Se señalo previamente (sección 2.4.1) que typedef se puede usar para definir nuevos nombres de datos

explicítamente, usando algunos de los tipos de datos de C, donde su formato es:

typedef <tipo> <nombre>;

Se puede usar typedef para crear nombres para tipos más complejos, como una estructura, por ejemplo:

typedef struct direc

{

char nombre[30];

char calle[40];

char ciudad[20];

char estado[3];

unsigned int codigo;

} sdirec;

sdirec info_direc={"Vicente Fernandez","Fantasia 2000","Dorado","MMX",12345};

en este caso direc sirve como una etiqueta a la estructura y es opcional, ya que ha sido definido un nuevo tipo de

dato, por lo que la etiqueta no tiene mucho uso, en donde sdirec es el nuevo tipo de datos e info_direc es una

variable del tipo sdirec, la cual es una estructura.

Con C también se pueden tener arreglos de estructuras:

typedef struct direc

{

char nombre[30];

char calle[40];

char ciudad[20];

char estado[3];

unsigned int codigo;

} info_direc;

info_direc artistas[1000];

por lo anterior, artistas tiene 1000 elementos del tipo info_direc. Lo anterior podría ser accesado de la

siguiente forma:

artistas[50].codigo=22222;

Procedimiento:

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	SALIDA 2	SALIDA 3
ENSENADA	ADANESNE	E N S E N A D A
SALIDA 4	SALIDA 5	SALIDA 6
A D A N E S N E	ENSENADA ENSENAD ENSENA ENSEN ENSE ENS EN E	ADANESNE ADANESNE ADANES ADANE ADAN ADA AD A
SALIDA 7	SALIDA 8	SALIDA 9

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

(VALIDADA AL 100% NO NUMEROS, NO DOBLES ESPACIOS Y SOLO MAYUSCULAS ENLA CADENA)

Conclusiones:

Estos dos programas en C ofrecen una serie de operaciones para manipular cadenas de texto ingresadas por el usuario. El primero se enfoca en transformaciones y desglose de la cadena, mientras que el segundo se centra en operaciones de formato y análisis. Ambos programas demuestran la versatilidad del lenguaje C para trabajar con cadenas de caracteres y ofrecen al usuario opciones útiles para modificar y analizar texto. Estos programas pueden ser útiles como ejemplos de cómo trabajar con cadenas en C y pueden servir como base para desarrollar aplicaciones más complejas que requieran manipulación de texto.

Anexo:

https://drive.google.com/file/d/1L7kOQ4ltvAuV4lUqbrm51Bwyz52zZrIB/view?usp=sharing