

CADENA CONTIENE

BRIGADA 6

Análisis, describiendo el problema e identificando los datos de entrada y de salida “cadena contiene”.

Al realizar el programa de cadena contiene, debemos entender es que se refiere que en este programa tiene como finalidad que una cadena contiene otra cadena dentro de la cadena principal. Es decir esta la cadena principal y una sub cadena.

Por ejemplo:

Almacenar (cadena 1) - Alma (cadena 2)

Literatura (cadena 1) - Litera (cadena 2)

Panificadora (cadena 1) - Pan (cadena 2)

Por lo tanto, para este proyecto se debe validar si una cadena contiene a otra, por lo cual no puede funcionar si la segunda cadena no se encuentra en la primera cadena por ejemplo:

Extemporáneamente (cadena 1) - Extemporáneo (cadena 2)

Para el desarrollo de nuestro código:

En cuanto a las librerías que utilizamos fue:

- **Stdio.h:** Esta biblioteca contiene las definiciones de macros, las constantes, las declaraciones de funciones y la definición de tipos usados por varias operaciones estándar de entrada y salida.
- **String.h:** Esta biblioteca contiene un conjunto de funciones para manipular cadenas: copiar, cambiar caracteres, comparar cadenas, etc.

Empezamos a declarar con **int main** como entrada de nuestro programa, junto con **int main** agregamos **(void)** que significa vacío, es decir que a la función main no se le está mandando nada, podría omitirse el void dentro de los paréntesis, el compilador asume que no se enviará nada.

Después para declarar dos cadenas utilizamos **char**, del cual para definir variables de tipo cadena, estas se definen como vectores de caracteres, esto es, anteponiendo la palabra reservada char al identificador de la variable, y después entre corchetes la longitud máxima de cadena.

Después tenemos a la sentencia **if-else** del cual es usada para ejecutar una instrucción o bloque de instrucciones solo si una condición es cumplida. Donde la condición es la

expresión que será evaluada. Si esta condición es true (verdadera), el extracto es ejecutado. Si esta es falsa (false), el extracto es ignorado (no ejecutado) y el programa continua en la siguiente instruccióndespués de la estructura condicional.

Dentro de la condición if tenemos la función **strstr**, del cual tiene como utilidad de devolver un puntero a la posición donde se encuentra la sub cadena, solo que no modifica la cadena original. Además esta es una función del lenguaje de programación C que busca una sub cadena de caracteres dentro de una cadena mayor. Como toda función tiene que regresar un valor, regresa NULL en caso de no encontrarla o la posición en la que inicia la sub cadena.

Mediante la función **printf** podemos escribir datos en el dispositivo de salida estándar. La función printf devuelve el número de caracteres escritos. En el que nos servirá para la sentencia if-else en el caso de que la condición sea verdadera y si la condición es falsa.

Hablando de la función printf, al emplear el especificador de formato %s, la impresión del contenido de un arreglo de tipo char, como caracteres en la consola. La función irá tomando cada uno de los elementos del arreglo e irá mostrando en la pantalla cada uno de los caracteres correspondientes, hasta encontrar el delimitador de la cadena.

Pseudocódigo

INICIO

Función res<-Cadena (cadena1, cadena2)

cadena1:CADENA

cadena2:CADENA

ESCRIBIR "Introduce una cadena"

LEER cadena1

ESCRIBIR "Introduce otra cadena"

LEER cadena2

SI (strstr(Cadena2 está en Cadena1) !=NULL) Entonces

Escribir "Existe " " dentro de " " ", cadena2, cadena1

FIN SI

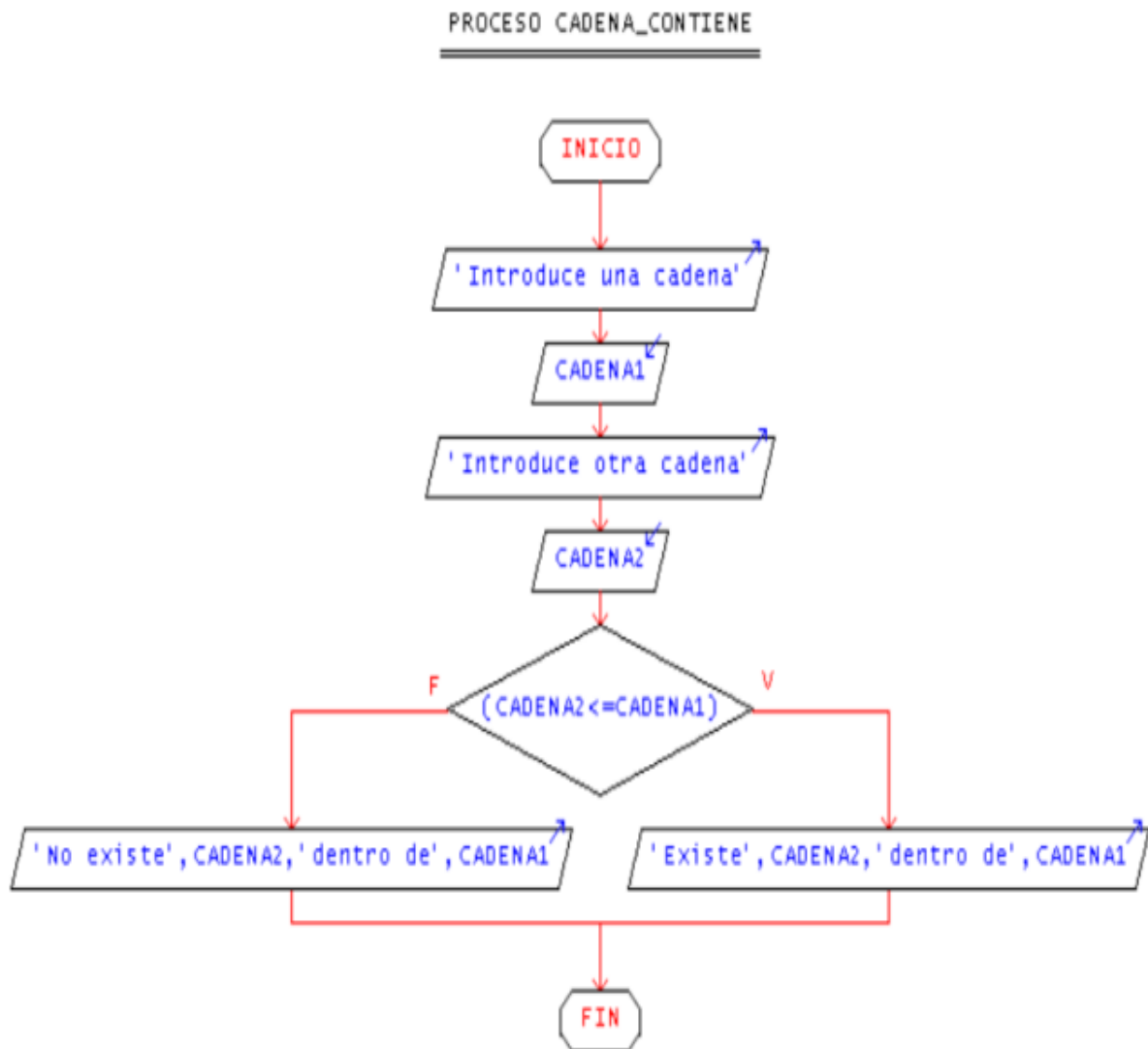
DE LO CONTRARIO

Escribir "No existe " " dentro " " ", cadena2, cadena1

FIN DE LO CONTRARIO

FIN

Diagrama de flujo



Programa en C

```
#include <stdio.h>

#include <string.h>

int main(void) {

    char cadena1[100];

    char cadena2[50];

    printf("\t\n Introduce una cadena");

    scanf("%s", cadena1);

    printf("\t\n Introduce otra cadena");

    scanf("%s", cadena2);


    if (strstr(cadena1, cadena2) != NULL)

    {

        printf("\t\n Existe %s dentro de %s\n", cadena2, cadena1);

    } else {

        printf("\t\n No existe %s dentro de %s\n", cadena2, cadena1);

    }

}
```

PRUEBA DE ESCRITORIO

```
Introduce una cadena panificadora  
  
Introduce otra cadena pan  
  
Existe pan dentro de panificadora  
  
Process returned 37 (0x25)    execution time : 13.815 s  
Press any key to continue.
```

```
Introduce una cadena ferrocarril  
  
Introduce otra cadena ferroso  
  
No existe ferroso dentro de ferrocarril  
  
Process returned 43 (0x2B)    execution time : 13.587 s  
Press any key to continue.
```

TEST

```
/*  
  
char cadena1[] = "gentrificacion";  
  
char cadena2[] = "gente";  
  
buscar(cadena1, cadena2), 0  
  
  
  
char cadena1[] = "almacenes";  
  
char cadena2[] = "alma";  
  
buscar(cadena1, cadena2), 1  
  
*/  
  
#include "..\index.c"  
  
int main(void)  
{  
  
char cadena1[] = "gentrificacion";  
  
char cadena2[] = "gente";  
  
if (buscar(cadena1, cadena2), 0)  
{  
  
printf("\t\n Existe %s dentro de %s\n", cadena2, cadena1);  
  
} else {  
  
printf("\t\n No existe %s dentro de %s\n", cadena2, cadena1);  
  
}  
  
}
```



```
#include "..\index.c"

int main(void)
{
    char cadena1[] = "almacenes";
    char cadena2[] = "alma";
    if (buscar(cadena1, cadena2), 1)
    {
        printf("\t\n Existe %s dentro de %s\n", cadena2, cadena1);
    } else {
        printf("\t\n No existe %s dentro de %s\n", cadena2, cadena1);
    }
}
```