

Cadena Contiene

Objetivos:

Validar si una cadena contiene otra al inicio

Análisis de “cadena contiene”.

En palabras concretas el objetivo es tener una cadena y una derivación algo como una subcadena. En este caso algunos ejemplos prácticos serían:

Ocelote (cadena 1) - Elote (cadena 2)

Tostadora (cadena 1) - Tostar (cadena 2)

En este caso, en el proyecto se debe de validar si una cadena contiene a otra, por lo tanto no puede funcionar si la segunda cadena no se encuentra dentro de nuestra primera cadena por ejemplo:

Correspondientemente (cadena 1) - Corresponder (cadena 2)

En esta ocasión se usó la biblioteca `#string.h`, dado que nos permite usar funciones para manipular cadenas desde cambiar caracteres, comparar cadenas, copiar, entre otras funciones.

Se empezó con la declaración de `int main` como entrada del programa, pero se le adjunto `(void)` vacío en español, haciendo que la función `main` no reciba nada. Declaramos 2 cadenas mediante **char**, de manera que sean vectores de caracteres, esto tiene como propósito declarar variables de tipo cadena, así como la longitud máxima de las cadenas en 50 y 100 (dado que en este caso la mayoría

de palabras no supera arriba de 40 caracteres en la mayoría de casos).

Se hace uso de impresiones en pantalla para digitar la cadena y esta sea posteriormente analizada por los condicionales **if-else** dado que si la condición es verdadera (true) será ejecutada, en caso contrario si es falsa (false) no será ejecutado y continuará automáticamente el programa con la siguiente instrucción.

Aunque también está dentro del condicional de tipo **if-else** otra función **strstr**, que nos devuelve un puntero hacia donde se ubica la subcadena, sin modificar la cadena original. En sí lo que hace es buscar una subcadena de caracteres dentro de una cadena mayor. Está por lo general tiene que regresar un valor, regresa **NULL** en caso que no sea encontrada o a la posición donde inicia la subcadena.

Una vez ingresamos ambas cadenas, nos indica si se cumplen las condiciones imprimiendo en pantalla “Es posible” o “No es posible”.

Pseudocódigo

INICIO

Función res<-cdn (cdn01, cdn02)

cdn01:CADENA

cdn02:CADENA

ESCRIBIR "Ingrese una cadena"

LEER cdn01

ESCRIBIR "Ingrese otra cadena"

LEER cdn02

SI (strstr(cdn02 está en cdn01) !=NULL) Entonces

Escribir "Existe " " adentro de " " ", cdn02, cdn01

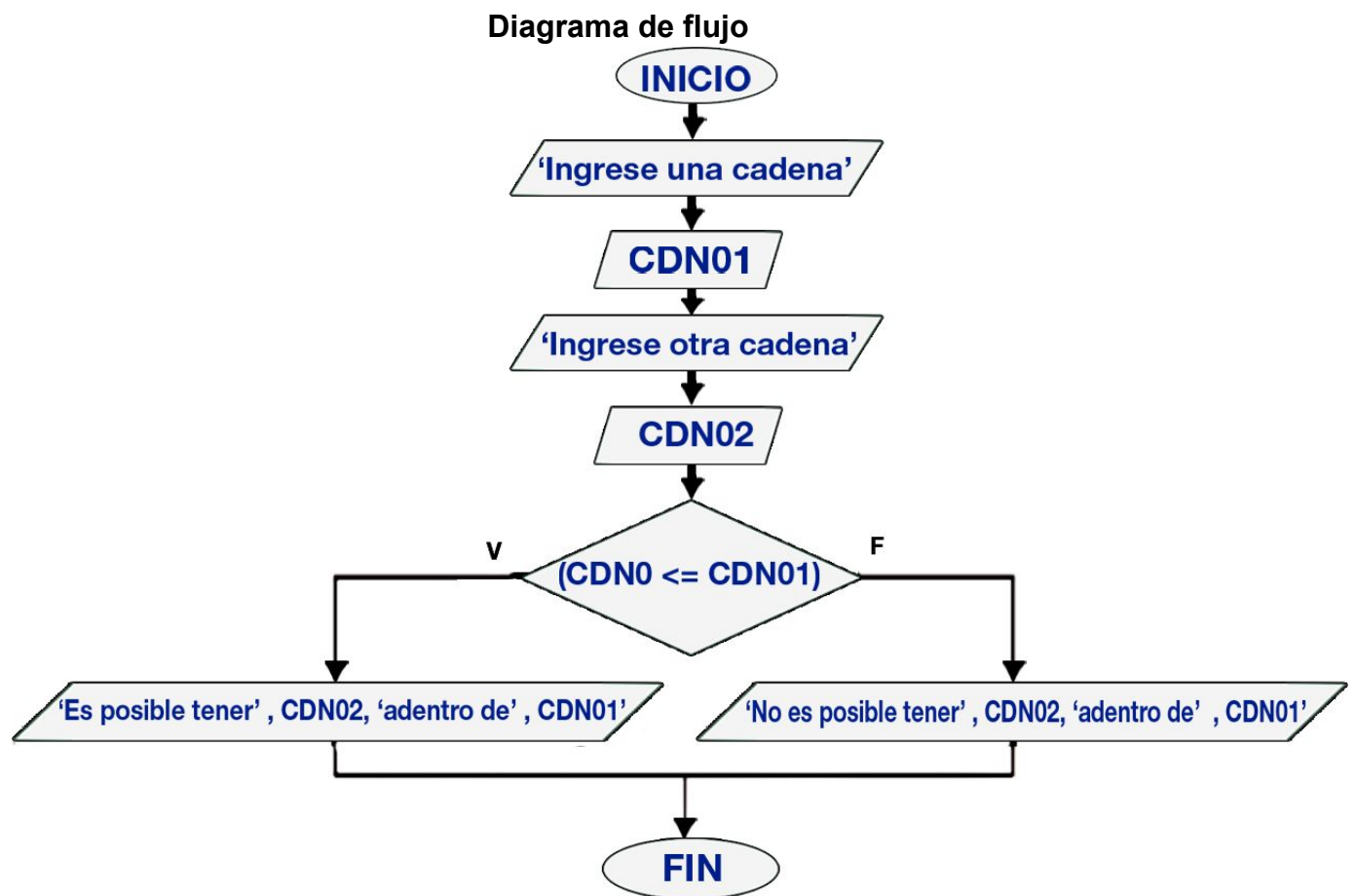
FIN SI

DE LO CONTRARIO

Escribir "No existe " " adentro de " " ", cdn02, cdn01

FIN DE LO CONTRARIO

FIN



Programa en C

```
#include <string.h>
#include <stdio.h>

int main(void)
{
    char cdn02[50], cdn01[100];
    printf("\t\n Ingrese una cadena");
    scanf("%s", cdn01);
    printf("\t\n Ingrese otra cadena");
    scanf("%s", cdn02);

    if (strstr(cdn01, cdn02) != NULL)
    {
        printf("\t\n Es posible tener %s adentro de %s\n", cdn02, cdn01);
    }
    else
    {
        printf("\t\n No es posible tener %s adentro de %s\n", cdn02, cdn01);
    }
}
```

```

1  #include <string.h>
2  #include <stdio.h>
3
4  int main(void)
5  {
6      char cdn02[50], cdn01[100];
7      printf("\t\n Ingrese una cadena");
8      scanf("%s", cdn01);
9      printf("\t\n Ingrese otra cadena");
10     scanf("%s", cdn02);
11
12     if (strstr(cdn01, cdn02) != NULL)
13     {
14         printf("\t\n Es posible tener %s adentro de %s\n", cdn02, cdn01);
15     }
16     else
17     {
18         printf("\t\n No es posible tener %s adentro de %s\n", cdn02, cdn01);
19     }
20 }

```

Prueba de escritorio del programa

```

Ingrese una cadena Ocelote
Ingrese otra cadena Ocesno

No es posible tener Ocelote dentro de Ocesno

```

Pruebas con la información del test

Gentrificación (cadena 1) y Gente (cadena 2)

Output	Clear
/tmp/Y6ByEV9lJf.o	
Ingrese una cadena Gentrificacion	
Gentrificacion	
Ingrese otra cadena Gente	
Gente	
No es posible tener Gente adentro de Gentrificacion	

Almacenes (cadena 1) y Alma (cadena 2)

Output

Clear

/tmp/Y6ByEV91Jf.o

Ingrese una cadena Almacenes

Almacenes

Ingrese otra cadena Alma

Alma

Es posible tener Alma adentro de Almacenes