



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS  
AVANZADOS

INSTITUTO ESTATAL DE CAPACITACIÓN  
DIPLOMADO DE SISTEMAS ELECTRÓNICOS

MÓDULO 3: PROGRAMACIÓN EN C / C++

JUEGO DE PUNTEROS

ELLIOT RUIZ SÁNCHEZ / PADTS IECA 8

ALEJANDRO CORONADO RIZO

JUEVES, 18 DE FEBRERO DEL 2021.

## Teoría básica

Los apuntadores son una de las capacidades del lenguaje C más difíciles de dominar. Estos les permiten a los programadores simular las llamadas por referencia, y crear y manipular estructuras de datos dinámicas, es decir, estructuras de datos que pueden crecer y encogerse en tiempo de ejecución.

Los apuntadores son variables cuyos valores son direcciones de memoria. Por lo general, una variable contiene directamente un valor específico. Por otro lado, un apuntador contiene la dirección de una variable que contiene un valor específico. En este sentido, el nombre de una variable hace referencia directa a un valor, y un apuntador hace referencia indirecta a un valor.

- El &, u operador de dirección, es un operador unario que devuelve la dirección de su operando.
- El operador \*, por lo general llamado operador de indirección u operador de desreferencia, devuelve el valor del objeto al que apunta su operando (es decir, un apuntador).

En C, los arreglos y los apuntadores están íntimamente relacionados, y a menudo se pueden utilizar de manera indistinta. Un nombre de arreglo puede interpretarse como un apuntador constante. Los apuntadores se pueden utilizar para realizar cualquier operación que involucre subíndices de arreglos.

Un apuntador a una función contiene la dirección de la función en memoria. Los apuntadores a funciones pueden pasarse a funciones, ser devueltos desde funciones, ser almacenados en arreglos y asignados a otros apuntadores a funciones.

Es importante mencionar que, según la documentación leída para este trabajo, se puede considerar un char como un apuntador natural.

De la misma manera, cabe recordar que un apuntador trabaja solo del primer elemento al último, como se vio en clase.

## Procedimiento y desarrollo

1. Usar un código de palíndromos como base para modificar y conseguir el código deseado, tales como:
  - a. Remover la condición de igualdad entre los primeros y últimos caracteres de la cadena.
  - b. Modificar la variable fija de almacenamiento para la cadena invertida por un puntero.

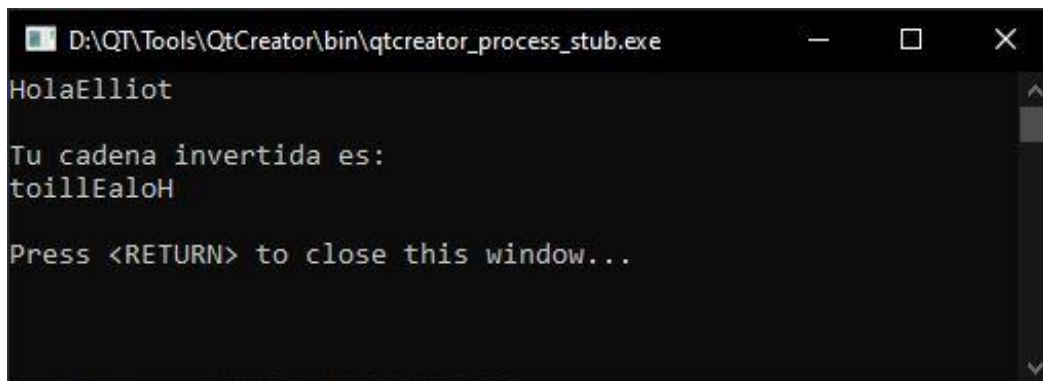
2. Escribir el código (evidencia anexada en el archivo .zip):

3. Adjuntar capturas de pantalla de los “warning” al correr el programa:

```
ptrletrauno = cadena[i];      Δ incompatible integer to pointer conversion assigning to 'char *' from 'char'; take the address with &
ptrletrados = cadena[i+1];    Δ incompatible integer to pointer conversion assigning to 'char *' from 'char'; take the address with &
cadena[i] = ptrletrados;      Δ incompatible pointer to integer conversion assigning to 'char' from 'char *'; dereference with *
cadena[i+1] = ptrletrauno;     Δ incompatible pointer to integer conversion assigning to 'char' from 'char *'; dereference with *
```

Estos errores se explican en el código, se dice que necesita el operador de dirección de “palabra” en la función para invertir la palabra pero al momento de ingresar este operador lo reconoce como un error fatal.

4. Correr el programa:



```
D:\QT\Tools\QtCreator\bin\qtcreator_process_stub.exe
HolaElliot

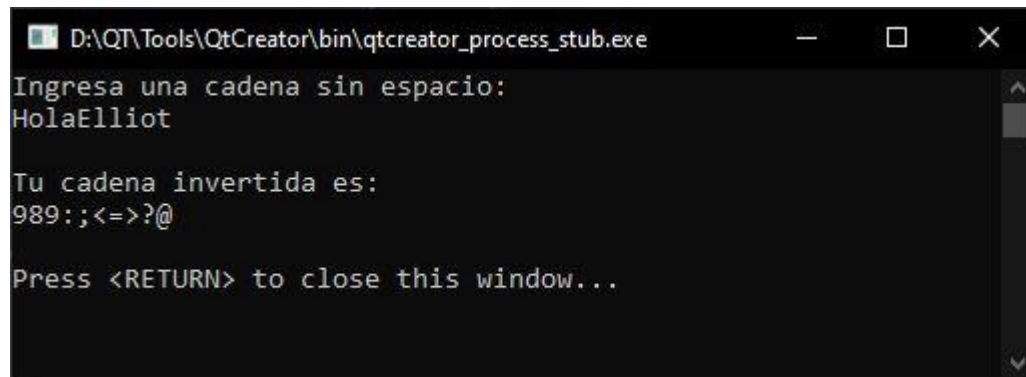
Tu cadena invertida es:
toillEaloH

Press <RETURN> to close this window...
```

5. Modificar el programa poniendo los operadores de dirección:

```
ptrletrauno = &cadena[i];
ptrletrados = &cadena[i+1];
cadena[i] = ptrletrados;
cadena[i+1] = ptrletrauno;
```

6. Correr el programa con las modificaciones para evidenciar el error:



```
D:\QT\Tools\QtCreator\bin\qtcreator_process_stub.exe
Ingresa una cadena sin espacio:
HolaElliot

Tu cadena invertida es:
989:;<=>?@

Press <RETURN> to close this window...
```

## Observaciones y conclusiones

El tema de punteros y, sobre todo su aplicación, sigue siendo un tema difícil, con lo cual, se tienen muchos errores. Es importante dar una lectura más extensa y, sobre todo, hacer prácticas para entender de una mejor manera el funcionamiento de los punteros.

Se aprendió mucha de la lectura sobre este tema e investigando distintos programas y como se modifican para el uso de los punteros, sobre todo el uso del programa "BubbleSort", este recurso fue encontrado en la biblioteca del CIMAT y fue de gran ayuda para comprender algunos datos e investigar otros.

Pero, tal como se comenta al principio, es importante retomar este tema ya que será muy importante en los temas siguientes y, sobre todo, en la aplicación en microcontroladores y sistemas digitales, ya que esto genera tener una mejor eficiencia al momento de usar los códigos en las aplicaciones de campo para generar mejores y más rápidas respuestas.