

Temas:

- Variables puntero.

Objetivos:

Que el estudiante logre...

- Identificar las características y usos de las variables puntero mediante su aplicación en problemas básicos y en el recorrido de arreglos.

Condiciones de presentación.

- Este trabajo práctico debe realizarse en forma individual.
- El código de los programas solicitados debe estar correctamente rotulado para su identificación, incluyendo comentarios de seguimiento y deberá ser enviado mediante la plataforma CUV.FCEYT hasta el día indicado como límite de presentación.
- La resolución completa de este trabajo práctico, incluyendo diagramas de flujo, deberá pasar a integrar la carpeta de práctica y autoevaluación del alumno.
- **Criterios de evaluación y aprobación.** Este trabajo práctico recibirá una calificación de aprobado o desaprobado. Para aprobar la presentación debe cumplir como mínimo con los siguientes ítems:
- El trabajo práctico debe estar desarrollado completamente.
- La codificación en lenguaje C debe realizarse siguiendo las recomendaciones de la cátedra.
- La presentación del código de los enunciados solicitados deberá realizarse en tiempo y forma mediante la plataforma CUV.FCEYT: <http://cuv.unse.edu.ar>.

Recursos Bibliográficos

- Introducción al Lenguaje C - Santos Espino.
- Introducción a la Programación con C – Marzal.
- Diapositivas, lecciones y video sobre el tema, disponibles en plataforma del CUV.

Tareas a desarrollar para cada enunciado

- ❖ Desarrolle la diagramación de flujo y la codificación en Lenguaje C de los siguientes enunciados.
- ❖ La resolución de todos los enunciados deberá estar desarrollada en la carpeta de práctica de la asignatura.
- ❖ La entrega de los programas en lenguaje C de los enunciados indicados, deberán ser enviados mediante la plataforma CUV según las fechas y horarios de presentación de cada uno.

Ejercicio 1: Codificando con punteros

Desarrollar un programa en lenguaje C que cumpla con las siguientes tareas:

- Declarar dos variables de tipo float llamadas x y y, y un puntero ptr de tipo float.
- Ingresar los valores de x y y, controlando que sean distintos de cero.
- Hacer que el puntero apunte a x, y mostrar:
 - La dirección de memoria de x.
 - El valor almacenado en x.
 - El valor accedido mediante el puntero (*ptr).
- Luego, hacer que el puntero apunte a y, y mostrar la misma información.
- Modificar el valor de y a través del puntero y comprobar que el cambio se refleja en la variable original.

Reflexionar: ¿qué ocurre si se modifica el valor de la variable a través del puntero?

Ejercicio 2: Swap usando punteros

Teniendo en cuenta el código presentado en el Ejercicio 1, modificar el código de modo que se lean de entrada los valores de las variables j y k y se intercambien sus valores usando punteros. Por ejemplo:

Salida: Ingrese el valor de la variable j

Entrada: 22

Salida: Ingrese el valor de la variable k

Entrada: 165

Salida: Ahora el valor de j es 165 y el valor de k es 22

<p><u>Ejercicio 2. Presentación: código en C.</u></p> <p><u>Fecha: 14/09</u></p>
--

Ejercicio 3: Aplicando punteros en conversión de cadenas

Un usuario ingresa una cadena de caracteres que representa un número entero positivo (por ejemplo "12345"). Toda entrada debe ser controlada, solo pueden ingresarse caracteres numéricos.

TRABAJO PRÁCTICO 3

Desarrollar un programa para convertir la cadena a su valor numérico utilizando punteros para recorrer carácter por carácter.

Una vez convertido el número, se deberá mostrar en pantalla:

- El número original.
- El doble del número.
- El número invertido (ejemplo: si se ingresó "12345", mostrar "54321").

Restricciones:

- Utilizar punteros siempre que sea posible para recorrer y acceder a los caracteres de la cadena.
- No usar funciones de conversión directas como atoi o strtol.
- Se permite el uso de funciones de la librería estándar como strlen para determinar la longitud de la cadena.

Ejemplos:

Entrada: 12345

Salida:

Número original: 12345

Doble del número: 24690

Número invertido: 54321

Entrada: 12p34

Salida: Error: la cadena contiene caracteres no numéricos.

Ejercicio 3. Presentación: código en C.
Fecha: 14/09

Ejercicio 4

Escribir un programa en lenguaje C que, usando punteros sobre una cadena de caracteres, determine si la palabra o frase ingresada es un isograma.

(*) Isograma: Una palabra o frase en la que ninguna letra aparece más de una vez.

Por ejemplo:

“murciélago” → es un isograma.

“programar” → no es un isograma (la letra "r" se repite).

Condiciones:

- Deben ignorarse los acentos, los espacios en blanco y las diferencias entre mayúsculas y minúsculas.
- El programa debe indicar claramente si la palabra o frase ingresada es o no un isograma.

Ejercicio 5

Una empresa de software educativo está desarrollando una aplicación para niños que les permita visualizar las palabras al revés para mejorar su atención y memoria.

Como parte de este desarrollo, se solicita implementar un programa en lenguaje C que:

- Lea una cadena de caracteres ingresada por el usuario.
- Muestre en pantalla la misma cadena pero invertida, es decir, con los caracteres en orden inverso al ingresado.
- El acceso y recorrido de la cadena deben realizarse usando punteros, evitando el uso de índices tradicionales de arreglos.

Ejercicio 6: Probando un código con puntero a puntero

Un sistema de mensajería desea que todos los mensajes se muestren en mayúsculas. Leer una cadena de caracteres ingresada por el usuario. Usar un puntero para recorrer la cadena y convertir cada letra a mayúscula directamente en el main. Mostrar la cadena convertida en pantalla.

Ejercicio 7: Ejercicio usando punteros

En un sistema de control de carrusel en un parque de diversiones, los carros del carrusel están numerados y dispuestos en un arreglo. Por motivos de reorganización, se necesita rotar los carros hacia la derecha un cierto número de posiciones.

El programa debe:

- Leer un arreglo de enteros que representan los identificadores de los carros en el carrusel.
- Leer un valor entero z que indica cuántas posiciones se deben desplazar hacia la derecha.
- Usar punteros para recorrer y modificar el arreglo, de modo que los elementos que estaban a la derecha aparezcan al inicio, manteniendo el orden relativo.
- Mostrar el arreglo modificado en pantalla. Utilice punteros.

Ejercicio 7. Presentación: código en C.
Fecha: 14/09