



Ayudantía 1 Estructura de datos

Profesor: Sebastián Sáez San Juan

Ayudantes:

Diego Duhalde

Benjamín Wiedmaier

Fernando Andrés Zamora Carrasco

1. Explique qué es un puntero y cómo se declara en C. Proporcione un ejemplo.
2. Describa la diferencia entre un puntero y una variable normal. ¿Qué ventajas ofrece el uso de punteros?
3. Explique qué es la aritmética de punteros y cómo se puede utilizar para recorrer un array.
4. ¿Qué es un puntero nulo y cuándo se debe utilizar?
5. Escriba el código apropiado para resolver los siguientes ejercicios
 - a) Escriba un programa en C que declare un puntero a un entero, asigne un valor a esa variable a través del puntero e imprima el valor y la dirección de memoria.
 - b) Escriba un programa en C que utilice punteros para intercambiar los valores de dos variables.
 - c) Escriba un programa en C que utilice un puntero para recorrer un array y calcular la suma de sus elementos.

RESPUESTAS

1. Un puntero es una variable cuyo contenido es una dirección de memoria, es decir, apunta a la ubicación donde se almacena otra variable. Se declara especificando el tipo de dato al que apuntará, seguido de un asterisco y el nombre del puntero. Un ejemplo puede ser el siguiente:
Tenemos una variable entera a , a la cuál podemos asignar un puntero p de la siguiente forma:
`int *p = &a;`
2. Una variable normal son aquellas que almacenan directamente un valor, como un número, un carácter, etc., mientras que un puntero almacena la dirección de memoria de otra variable. Las ventajas de usar punteros incluyen:
 - Permiten el paso eficiente de grandes estructuras de datos a funciones sin necesidad de copiar toda la estructura.
 - Facilitan la creación y manipulación de estructuras de datos dinámicas como listas enlazadas, pilas y colas.
 - Permiten el acceso y modificación directa de memoria, lo cual puede ser útil en programación de bajo nivel y optimización de rendimiento.
3. La aritmética de punteros consiste en realizar operaciones (como suma o resta) sobre las direcciones de memoria contenidas en un puntero. Esto resulta muy útil para recorrer arrays, ya que permite acceder a elementos consecutivos de manera eficiente. Por ejemplo, si tenemos un array de enteros y un puntero apuntando al primer elemento, podemos incrementar el puntero para acceder a los siguientes elementos del array.
4. Un puntero nulo es un puntero que no apunta a ninguna dirección de memoria válida. Se utiliza para indicar que el puntero no está inicializado o que no se le ha asignado una dirección de memoria. En C, se puede declarar un puntero nulo asignándole el valor NULL. Es útil para evitar errores de acceso a memoria no válida y para comprobar si un puntero ha sido inicializado correctamente.
5. Los códigos que resuelven los problemas se encuentran en el repositorio del ramo en la rama de "Ayudantías". [Link](#).