

Ejercicio 1: Intercambio de valores (swap)

Escribí una función llamada `void swap(int *a, int *b)` que reciba dos punteros a enteros y **intercambie sus valores** usando una variable auxiliar.

Ejercicio 2: Suma de dos números usando punteros

Declarar dos variables enteras y dos punteros que apunten a ellas.

Calcular la suma de los dos valores usando únicamente los punteros (sin usar directamente las variables originales) y mostrar el resultado por pantalla.

Ejercicio 3: Acceso a array con punteros

Declarar un array de 5 enteros y un puntero que apunte al primer elemento. **Recorrer el array usando aritmética de punteros** `*(ptr + i)` e imprimir cada valor por pantalla.

Pista: Declarar el puntero con `int *ptr = arr;`. En el `for`, no uses `arr[i]`, sino `*(ptr + i)`.

Ejercicio 4: Encontrar el máximo de un array

Escribir una función `int max(int *arr, int size)` que reciba un array de enteros y su tamaño, y **devuelva el valor máximo contenido en el array**, utilizando únicamente punteros (sin usar índices).

Pista: Usá un puntero `p` para recorrer el array. Inicializá una variable `max` con `*arr`

Ejercicio 5: Invertir los elementos de un array

Escribir una función `void invertir(int *arr, int size)` que reciba un array de enteros y lo **invierta "in-place"** (sin usar un array auxiliar), utilizando punteros.

Usá dos punteros: uno al inicio (`arr`) y otro al final (`arr + size - 1`). Intercambiá sus valores con `swap`

Ejercicio 6: Convertir un string a mayúsculas

Escribir una función `void aMayusculas(char *str)` que reciba una cadena de caracteres y la **convierta a mayúsculas**, carácter por carácter, **sin usar funciones de `string.h`**.

Chequear si el carácter está entre `'a'` y `'z'`...Ver caracteres ASCII.

Ejercicio 7: Convertir un string a minúsculas

Escribir una función `void aMinusculas(char *str)` que reciba una cadena de caracteres y la **convierta a minúsculas**, carácter por carácter, **sin usar funciones de `string.h`**. Ver caracteres ASCII.

Chequear si el carácter está entre `'A'` y `'Z'`...Ver caracteres ASCII.

Ejercicio 8: Calcular la longitud de un string

Escribir una función `int longitud(const char *str)` que reciba una cadena de caracteres y **devuelva su longitud**, contando los caracteres hasta encontrar el carácter nulo `'\0'`. No se debe usar `strlen()` ni ninguna función de biblioteca.

Recorre el string con un puntero o un bucle `while (*p != '\0')`

Ejercicio 9: Crear un programa que permita al usuario ingresar 15 números en un arreglo de enteros. El programa debe calcular diversas estadísticas sobre este arreglo utilizando funciones que reciban los datos por referencia y modifiquen variables a través de punteros.

Requisitos:

1. Crear las siguientes funciones:

- Una función para leer datos en un arreglo (recibe el arreglo y modifica su contenido)
- Una función para calcular múltiples estadísticas de un arreglo: suma, promedio, mayor, menor, cantidad de pares e impares (todos estos valores deben ser modificados por referencia, NO deben ser retornados por la función)
- Una función para imprimir un arreglo

2. Utilizar paso por referencia en todas las funciones que modifiquen datos.

3. El programa principal debe:

- Solicitar al usuario que ingrese los elementos del arreglo
- Mostrar el arreglo completo - Mostrar todas las estadísticas calculadas (suma, promedio, mayor valor, menor valor, cantidad de números pares y cantidad de números impares)

IMPORTANTE:

- Debes utilizar aritmética de punteros en lugar de notación de índices donde sea posible.
- Para modificar variables dentro de las funciones, debes pasarlas por referencia (usando punteros).
- Recuerda que para modificar el valor al que apunta un puntero debes usar el operador de indirección (*).

PISTAS:

- Para pasar una variable por referencia, debes pasar su dirección de memoria: &variable
- Para declarar un parámetro que recibe una referencia, usa un puntero: int *parametro
- Para acceder o modificar el valor referenciado, usa: *parametro - En arreglos, puedes acceder a los elementos con aritmética de punteros: *(arr + i) en lugar de arr[i]
- Para calcular estadísticas, debes inicializar los valores (por ejemplo, el mayor y menor pueden inicializarse con el primer elemento del arreglo)