

# Guia de Estudo – Exercícios C 00 e C 01

Este documento reúne os exercícios básicos de C (do estilo dos projetos da 42) com explicações detalhadas. O foco é: ponteiros, manipulação de memória, funções, operadores e lógica básica em C.

## Parte 1: Fundamentos do C

- `#include <write>` contém `write`.
- `#include <printf>` contém `printf`.
- `char` = 1 byte (ASCII).
- `int` = número inteiro (4 bytes).
- Ponteiro (`*`) guarda endereço de memória.
- `&` devolve o endereço.
- `*` (desreferenciação) acessa o valor armazenado.

## Parte 2: Exercícios C00

- EX00 – `ft_putchar` escreve um caracter com `write`.
- EX01 – `ft_print_alphabet` imprime 'a' até 'z'.
- EX02 – `ft_print_reverse_alphabet` imprime 'z' até 'a'.
- EX03 – `ft_print_numbers` imprime de '0' até '9'.
- EX04 – `ft_is_negative` imprime 'P' se positivo, 'N' se negativo.

## Parte 3: Exercícios C01

- EX00 – `ft_ft` altera valor de uma variável para 42 usando ponteiro.
- EX01 – `ft_ultimate_ft` 9 níveis de ponteiros, altera valor para 42.
- EX02 – `ft_swap` troca valores de duas variáveis via ponteiros.
- EX03 – `ft_div_mod` retorna divisão e resto via ponteiros.
- EX04 – `ft_ultimate_div_mod` altera diretamente `*a` e `*b` (divisão e resto).
- EX05 – `ft_putstr` imprime string até encontrar '\0'.
- EX06 – `ft_strlen` retorna tamanho de uma string.
- EX07 – `ft_rev_int_tab` inverte array de inteiros.
- EX08 – `ft_sort_int_tab` ordena array com Bubble Sort.

## Resumão para o Teste

- Ponteiros permitem alterar valores fora da função.
- Strings em C terminam com '\0'.
- Funções com ponteiros retornam múltiplos valores.
- `write` é baixo nível, `printf` tem formatação.
- Swap precisa de variável auxiliar.
- `while` é usado constantemente nos exercícios.
- Bubble Sort compara pares e empurra maiores para o final.