

# Relatório do Laboratório de Introdução à Ciência da Computação II – Avaliativo 1

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>

// Função para verificar se é palíndromo
int eh_palindromo(char *str) {
    char processada[1000];
    int j = 0;

    // Processa a string: considera apenas
    caracteres alfanuméricos
    for(int i = 0; str[i] != '\0'; i++) {
        char c = str[i];
        if(isalnum(c)) {
            processada[j++] = tolower(c);
        }
    }
    processada[j] = '\0';

    int len = j;

    // Strings vazias ou com 1 caractere são
    palíndromos
    if(len <= 1) {
        return 1;
    }

    // Verifica se é palíndromo comparando
    caracteres das extremidades
    for(int i = 0; i < len / 2; i++) {
        if(processada[i] != processada[len - 1 -
i]) {
            return 0;
        }
    }

    return 1;
}

int main() {
    char linha[1000];

    // Lê linha por linha até EOF
    while(fgets(linha, sizeof(linha), stdin) !=
NULL) {
        // Remove o '\n' do final se existir
        int len = strlen(linha);
        if(len > 0 && linha[len-1] == '\n') {
            linha[len-1] = '\0';
        }

        // Verifica se é palíndromo e imprime
        resultado
        if(eh_palindromo(linha)) {
            printf("Sim\n");
        } else {
            printf("Não\n");
        }
    }

    return 0;
}
```

## Código Iterativo IA

O código ao lado é a forma iterativa para verificação de palíndromos gerado por inteligência artificial, mais especificamente o Claude Sonnet 4.

É possível notar que esse código utiliza da biblioteca ctype.h para tratar os caracteres de entrada, tornando-os case-insensitive (com a função tolower) e considerando somente alfanuméricos (com a função isalnum).

Além disso, a função eh\_palindromo é iterativa, no qual ele percorre caracter por caracter do input original, processando-o em uma string melhor formatada e depois percorre essa nova string para identificar se é ou não um palíndromo, printando “Sim” se for ou “Não” se não for.

Para ler o input, o código do Claude utilizou while(fgets) para ler linha por linha até acabar as entradas, armazenando em uma string de tamanho 1000 fixo. Ele também remove o “\n” que essa string recebe substituindo-a por “\0”, dessa forma garante uma leitura correta dos valores.

No geral, esse código é bem econômico no tempo e na quantidade de linhas a serem executadas, mesmo sendo iterativo, assim ele é um programa bem escrito, legível e com um bom desempenho.