



UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Disciplina: **Programação Procedimental**

Curso: **Ciência da Computação**

Professora: **Elaine Ribeiro Faria**

Aula Prática no 12

Objetivo

Praticar o conceito de modularização de código usando o CodeBlocks.

1. Codifique, compile e execute um programa em C que leia dois vetores com 10 elementos cada e calcule quantos números positivos cada um desses vetores contém. Para isso, faça uma função que leia um vetor e uma função que dado um vetor calcule o número de elementos positivos (faça duas versões dessa função, uma que use passagem por referência e outra que não use). O programa principal deverá chamar cada uma dessas funções duas vezes, uma para cada vetor.
2. Codifique, compile e execute um programa em C que contenha uma estrutura CONTA que represente a conta corrente de um dado cliente de um Banco. O tipo CONTA deve possuir os seguintes campos:
 - Nome – Nome do correntista.
 - NumCC – Número da conta corrente (sem traço).

Para testar este novo tipo realize as seguintes tarefas:

(a) Crie uma função que preencha os campos de um correntista.

(b) Crie uma função que imprima na tela os dados do correntista.

Agora refaça o programa considerando que um vetor de correntistas de 5 posições.

3. Codifique, compile e execute um programa em C que armazene as características físicas de 50 habitantes de uma certa região. De cada habitante foram coletados os seguintes dados: sexo, cor dos olhos (azuis, verdes ou castanhos), cor dos cabelos (louros, pretos ou castanhos) e idade. Faça uma função que leia esses dados em um vetor de structs. A seguir, faça uma função que imprima as informações do vetor na tela. Faça uma função que dado o vetor de structs retorna o número de pessoas que são maior de idade. Use passagem de parâmetros por referência.
4. Codifique, compile e execute um programa em C que receba como parâmetro um número inteiro negativo n e o modifique, atribuindo a ele a soma entre todos os números inteiros entre n e -1 . Exemplo: se o número digitado for -4 , o resultado será $(-4) + (-3) + (-2) + (-1) = -10$
5. Codifique, compile e execute um program em C que receba como parâmetro um número inteiro não negativo n e permita ao usuário escolher entre calcular o fatorial de n ou o somatório de 1 a n . As funções fatorial e somatório devem ser implementadas usando recursão.
6. O quadrado de um número natural n é dado pela soma dos n primeiros números ímpares consecutivos. Por exemplo:
 $1^2=1$
 $2^2=1+3$
 $3^2=1+3+5$
 $4^2=1+3+5+7$

Dado um número n , calcule seu quadrado usando a soma de ímpares ao invés de produto. Use uma função recursiva para resolver o problema.

7. Refazer o exercício 7 da última aula, usando recursividade

Codifique, compile e execute um programa em C que contenha uma função que verifique se um valor é perfeito ou não. Um valor é dito perfeito quando ele é igual a soma dos seus divisores excetuando ele próprio. (Ex: 6 é perfeito, $6 = 1 + 2 + 3$, que são seus divisores).