

1ª Lista de Exercícios de Programação C

Resolva os exercícios dados abaixo utilizando técnicas de modularidade (funções C definidas pelo programador).

INSTRUÇÕES IMPORTANTES (não deixe de ler antes de iniciar):

- 1) O objetivo desta lista é desenvolver a programação modular. Esses exercícios poderiam ser resolvidos usando apenas uma função principal “main”, onde tudo é feito dentro dela. Mas esse não é o objetivo! O que será avaliado é sua capacidade de escrever funções em C que tornem o seu programa mais modular (com partes independentes e reutilizáveis) e assim, mais legíveis.
 - 2) **VALOR 4.0 pontos:** A lista contém 20 exercícios que somam bem mais que 4 pontos. Assim sendo, você deverá escolher de forma livre quais questões resolverá de tal forma que somem no máximo 4.0 pontos. A numeração escolhida deve estar claramente indicada.
 - 3) Os exercícios escolhidos para entrega da solução serão corrigidos conforme a ordem de sua numeração na lista. Aqueles que façam a soma dos pontos ultrapassar o máximo de 4, serão anulados.
 - 3) Você deverá entregar um único arquivo PDF contendo, para cada exercício escolhido por você, o respectivo código fonte (legível) e um print de tela da execução do seu programa que demonstre que funcionou.
 - 4) ENTREGA: **30 de junho**. Ainda será informado o e-mail para o envio do arquivo PDF do seu trabalho.
 - 5) Será descontado 1.0 ponto por dia de atraso e também por cada descumprimento dessas instruções.
-

Atenção! Algumas questões se utilizam de conceitos aritméticos que não defino aqui propositalmente. Pesquise quando não conhecer o termo, pois são termos relativamente bem conhecidos. A pesquisa faz parte da questão.

1ª QUESTÃO (0.5) Escreva um programa modular que receba um número e calcule a soma de todos os seus divisores próprios (excluindo ele mesmo).

2ª QUESTÃO (0.5) Escreva um programa modular que receba dois inteiros positivos a e b e imprima todos os números perfeitos no intervalo [a, b].

3ª QUESTÃO (0.5) Dado um número inteiro positivo, escreva um programa modular que exiba sua decomposição em fatores primos (Ex.: a fatoração de 12 é $12 = 2.2.3$).

4ª QUESTÃO (0.5) Dado um número inteiro positivo, verifique se ele é um palíndromo (ex.: 12321 é palíndromo, 234432 também é, todo número de um dígito é palíndromo também).

5ª QUESTÃO (0.5) Leia 10 inteiros, armazene em um vetor e mostre o maior, o menor, a média aritmética dos 10 valores e a distância entre essa média e cada um dos 10 valores armazenados.

6ª QUESTÃO (0.5) Leia dois inteiros e calcule o MMC entre eles usando o MDC.

7ª QUESTÃO (0.5) Escreva um programa que leia um número inteiro positivo e exiba sua representação binária.

8ª QUESTÃO (0.5) Dado um número natural, gere e imprima a sequência de Collatz até a sequência convergir para 1.

9ª QUESTÃO (0.5) Calcule a soma dos 3 maiores termos de uma sequência de Collatz cujo 1º termo é dado pelo usuário.

10ª QUESTÃO (1.0) Faça primeiro uma função que calcule a diferença absoluta entre a soma dos termos pares e a soma dos termos ímpares de uma sequência de Collatz, com o 1º termo da sequência dado pelo usuário. E, em seguida, faça um programa que exiba tal diferença para cada sequência de Collatz gerada a partir de um 1º termo pertencente ao intervalo $[a,b]$, dados a e b . Exemplo: se você der o intervalo $[8, 11]$, serão geradas 4 sequências de Collatz diferentes com 1º termo respectivamente 8, 9, 10 e 11. Cada uma delas tem sua própria diferença absoluta entre somatório de pares e ímpares. São essas 4 diferenças que deverão ser exibidas.

11ª QUESTÃO (1.0) Liste todos os números primos até um número n , usando o método do crivo de Eratóstenes.

12ª QUESTÃO (1.0) Agora liste todos os números primos até um número n , usando um outro método diferente.

13ª QUESTÃO (0.5) Leia dois inteiros a e b e imprima a soma dos dígitos de todos os números no intervalo $[a, b]$.

14ª QUESTÃO (0.5) Implemente a operação de potenciação usando multiplicações sucessivas, sem usar a função predefinida “pow”.

15ª QUESTÃO (1.0) Escreva um programa que se utilize de uma função para verificar se um número é triangular, ou seja, se é um número da forma $k.(k+1)/2$, para algum k natural, e com auxílio desta função liste todos os números triangulares até um limite estabelecido pelo usuário.

16ª QUESTÃO (1.0) Liste todos os pares de números amigáveis até um limite n .

17ª QUESTÃO (0.5) Para um número n dado, calcule a soma alternada dos seus dígitos (ex: $5384 \rightarrow 5 - 3 + 8 - 4$).

18ª QUESTÃO (0.5) Dado um número, exiba quantos dígitos pares e quantos ímpares ele possui.

19ª QUESTÃO (1.5) Converta um número inteiro positivo para qualquer base numérica entre 2 e 9.

20ª QUESTÃO (2.0) Escreva um programa modular que liste, até um limite dado pelo usuário, todos os números ímpares que são fatoráveis em apenas dois números primos diferentes. Para tal, escreva uma função que testa essa condição “ser fatorável em apenas dois números primos diferentes”, para um certo número ímpar dado como entrada. Exemplo: 77 é um ímpar que atende à propriedade desejada, pois $77 = 7 \times 11$, e 7 e 11 são primos.