

Lista de Exercícios - Funções

IMPORTANTE!!!

As soluções deverão obedecer rigorosamente os cabeçalhos propostos;

1. Escreva uma função que receba três parâmetros do tipo inteiro e retorne a quantidade de múltiplos do terceiro parâmetro entre os valores do primeiro e o segundo parâmetros.

int ex01(int a, int b, int c);

2. Escreva uma função que receba um parâmetro do tipo float, que represente uma temperatura em fahrenheit, e retorne um valor do tipo float, que representa a temperatura correspondente em graus centígrados. A conversão de graus fahrenheit para centígrados é obtida por **centigrados = 5.0 * (fahrenheit - 32.0)/9.0**.

float ex02(float fahrenheit);

3. Escreva uma função que receba um parâmetro do tipo inteiro e retorne verdadeiro (1) se o valor recebido é um número primo, caso contrário retorne falso (0).

int ex03(int valor);

4. Escreva uma função que receba dois parâmetros do tipo inteiro e retorne a soma de todos os inteiros no intervalo informado, inclusive os valores passados.

int ex04(int a, int b);

5. Escreva uma função que receba como parâmetro um inteiro positivo e retorne a raiz quadrada do mesmo, sem usar a função sqrt. Para isso, você precisa saber que a raiz quadrada de um número N é igual à quantidade de números ímpares consecutivos (a partir do 1) cuja soma é igual a N (ou o mais próxima possível de N). Ou melhor, exemplificando:

- Qual a raiz quadrada de 16? $1 + 3 + 5 + 7 = 16$
 - ☐ 4 números ímpares consecutivos foram somados. Então 4 é a raiz quadrada de 16.
- Qual a raiz quadrada de 25? $1 + 3 + 5 + 7 + 9 = 25$
 - ☐ 5 números ímpares consecutivos foram somados. Então 5 é a raiz quadrada de 25.
- Qual a raiz quadrada de 36? $1 + 3 + 5 + 7 + 9 + 11 = 36$
 - ☐ 6 números ímpares consecutivos foram somados. E 6 é a raiz quadrada de 36.
- Qual a raiz quadrada de 30?

- A raiz quadrada inteira de 30 é 5 (aproximando para baixo), porque $25 < 30 < 36$, logo a parte inteira da raiz quadrada de 30 é igual à raiz quadrada de 25.

float ex05(int v);

6. Escreva uma função que receba quatro parâmetros do tipo inteiro e retorne um valor float, que será a média ponderada entre o primeiro e o segundo parâmetro, sendo que o terceiro parâmetro será o peso para o primeiro parâmetro e o quarto o peso para o segundo.

float ex06(int, int, int, int);

7. Escreva uma função que receba um parâmetro n do tipo inteiro, positivo, e retorne a soma dos n termos da seguinte sequência:

$$S2 = \frac{1}{2} + \frac{2}{4} + \frac{3}{6} + \frac{4}{8} + \cdots + \frac{n}{2n}$$

float ex07(int);

8. Escreva uma função que receba dois números inteiros como parâmetros e retorne um valor, também inteiro, que represente a multiplicação de um número pelo outro, sem utilizar o operador de multiplicação (*).

Obs.: Lembrar que uma multiplicação pode ser definida por uma sucessão de somas.

int ex08(int, int);

9. Escreva uma função que receba como parâmetro um número inteiro e retorne verdadeiro (1) se é um número perfeito ou falso (0) senão é um número perfeito.

Obs.: Número perfeito é aquele cuja soma de seus divisores, exceto ele próprio, é igual ao número. Ex.: $6 = 1 + 2 + 3$.

int ex09(int);

10. Escreva uma função que receba dois parâmetros do tipo float, o primeiro representa o número de itens vendidos e o segundo o valor total das vendas, e retorne o valor da comissão a ser paga, de acordo com os seguintes critérios:

- para pedidos com menos de 20 itens, a comissão é de 10% do valor total do pedido;
- para pedidos de 20 a 49 itens, a comissão é de 15% do valor total do pedido;
- para pedidos de 50 a 74 itens, a comissão é de 20% do valor total do pedido;
- para pedidos iguais ou superiores a 75 itens, a comissão é de 25%.

float ex10(float, float);

11. Escreva uma função que receba um parâmetro do tipo float, que representa o raio de um círculo, e retorne a área do círculo, através da fórmula $A = PI * R * R$. Para a constante PI utilize o valor 3.14.

float ex11(float);

12. Escreva uma função que receba como parâmetro um valor do tipo inteiro e retorne o número primo mais próximo, acima dele. **Número primo é todo número inteiro que é divisível apenas por 1 e por ele próprio.** Alguns exemplos: 7 é o primo mais próximo de 8; 11 é o primo mais próximo de 10; 479 é o primo mais próximo de 471.

int ex12(int a);

13. Escreva uma função que receba dois parâmetros do tipo inteiro a e b, em seguida, faça a troca destes dois números, de forma que após chamar a função o conteúdo das variáveis passadas como parâmetros estejam trocados. Dica: a e b devem ser passados por referência.

void ex13(int *a, int *b);

14. Escreva uma função que receba dois parâmetros do tipo inteiro a e b, em seguida, decremente o primeiro e incremente o segundo, de forma que após chamar a função o valor de cada parâmetro tenha seus conteúdos alterados. Dica: a e b devem ser passados por referência.

void ex14(int *a, int *b);

15. Escreva uma função que receba como parâmetros três valores do tipo float, o primeiro representa o raio, o segundo o perímetro e o terceiro a área de um círculo. Em seguida, calcule e retorne através dos parâmetros perímetro e área o respectivo perímetro e área do círculo. **Dica:** perímetro e área devem ser passados por referência. **Obs.:** Ao chamar a função não será necessário fornecer um valor para perímetro e área, pois eles serão calculados pela função.

void ex15(float raio, float *perimetro, float *area);

16. A sequência de Fibonacci tem papel importante na explicação de fenômenos naturais. Ela é também bastante utilizada para fins estéticos, pela sua reconhecida harmonia. Exemplo disso foi sua utilização na construção do Partenon, em Atenas. A sequência dá-se inicialmente por dois números 1. A partir do terceiro elemento usa-se a expressão: $elementon = elementon-1 + elementon-2$. Exemplo de sequência: 1, 1, 2, 3, 5, 8.

Escreva uma função que receba um parâmetro do tipo inteiro, que representa o número de elementos da série de Fibonacci, e retorne a soma desses elementos.

int ex16(int n);

17. Escreva uma função que receba quatro parâmetros, onde:

- a. O primeiro, do tipo float, representa a altura de uma pessoa em metros;
- b. O segundo, do tipo inteiro, representa quantos centímetros essa pessoa cresce em um ano;
- c. O terceiro, do tipo float, representa a altura de uma segunda pessoa, em metros;
- d. O quarto, do tipo inteiro, representa quantos centímetros a segunda pessoa cresce em um ano;

Considerando que a primeira pessoa seja mais alta que a segunda, a função deverá retornar um valor inteiro representando a quantidade de anos que a segunda pessoa levará para ultrapassar a altura da primeira pessoa.

int ex17(float altura1, int cresce1, float altura2, int cresce2);

18. Escreva uma função que receba 3 parâmetros do tipo float X, Y e Z, e que verifique se esses valores podem ser os comprimentos dos lados de um triângulo e, neste caso, retornar qual o tipo de triângulo formado. Para que X, Y e Z formem um triângulo é necessário que a seguinte propriedade seja satisfeita: o comprimento de cada lado de um triângulo é menor do que a soma do comprimento dos outros dois lados. A função deve identificar o tipo de triângulo formado observando as seguintes definições:

1. Triângulo Equilátero: os comprimentos dos 3 lados são iguais.
2. Triângulo Isósceles: os comprimentos de 2 lados são iguais.
3. Triângulo Escaleno: os comprimentos dos 3 lados são diferentes

A função deverá retornar um número inteiro entre 1 e 3, representando o tipo de triângulo, ou 0 caso os valores não podem ser os comprimentos dos lados de um triângulo.

int ex18(float x, float y, float z);

19. Escreva uma função que receba quatro parâmetros, os três primeiros do tipo float, representando 3 notas de um aluno, e o quarto do tipo char. Se o quarto parâmetro for igual a 'A', a função deverá retornar a média aritmética das notas, se for igual a 'P', a função deverá retornar a média ponderada das notas, considerando como pesos 5, 3 e 2, se o quarto parâmetro for igual a 'H', a função deverá retornar a média harmônica das notas. Caso o quarto parâmetro não possua um valor válido, isto é, 'A', 'P' ou 'H', deverá adotar 'A'.

float ex19(float a, float b, float c, char d);