

Exercícios: Funções

1. Crie uma função que receba como parâmetro um número inteiro e devolve o seu dobro.
2. Faça uma função para verificar se um número é positivo ou negativo. Sendo que o valor de retorno será 1 se positivo, -1 se negativo e 0 se for igual a 0.
3. Faça uma função para verificar se um número é um quadrado perfeito. Um quadrado perfeito é um número inteiro não negativo que pode ser expresso como o quadrado de outro número inteiro. Ex: 1, 4, 9...
4. Faça uma função e um programa de teste para o cálculo do volume de uma esfera. Sendo que o raio é passado por parâmetro. (considere $\pi=4$)
$$V = \frac{4}{3} * \pi * R^3$$
5. Faça uma função que receba 3 números inteiros como parâmetro, representando horas, minutos e segundos, e os converta em segundos.
6. Faça uma função que receba uma temperatura em graus Celsius e retorne-a convertida em graus Fahrenheit. A fórmula de conversão é: $F = C * (9.0/5.0) + 32$, sendo F a temperatura em Fahrenheit e C a temperatura em Celsius. O valor da resposta é apenas o inteiro correspondente.
7. Faça uma função que receba a altura e o raio de um cilindro circular e retorne o volume do cilindro. O volume de um cilindro circular é calculado por meio da seguinte fórmula: $V = \pi * raio^2 * altura$, onde $\pi = 3$.
8. Faça uma função que receba dois números e retorne qual deles é o maior.
9. Elabore uma função que receba três notas de um aluno como parâmetros e uma letra. Se a letra for A, a função deverá calcular a média aritmética das notas do aluno; se for P, deverá calcular a média ponderada, com pesos 5, 3 e 2 (valor aproximado).
10. Escreva uma função que receba um número inteiro maior do que zero e retorne a soma de todos os seus algarismos. Por exemplo, ao número 251 corresponderá o valor 8 (2 + 5 + 1). Se o número lido não for maior do que zero, o programa terminará com a mensagem "Número inválido".
11. Faça uma função que receba dois valores numéricos e um símbolo. Este símbolo representará a operação que se deseja efetuar com os números. Se o símbolo for + deverá ser realizada uma adição, se for - uma subtração, se for / uma divisão e se for * será efetuada uma multiplicação.
12. Crie um programa que receba três valores (obrigatoriamente maiores que zero), representando as medidas dos três lados de um triângulo. Elabore funções para:
 - (a) Determinar se eles lados formam um triângulo, sabendo que:
 - O comprimento de cada lado de um triângulo é menor do que a soma dos outros dois lados.
 - (b) Determinar e mostrar o tipo de triângulo, caso as medidas formem um triângulo. Sendo que:
 - Chama-se equilátero o triângulo que tem três lados iguais.
 - Denominam-se isósceles o triângulo que tem o comprimento de dois lados iguais.

- Recebe o nome de escaleno o triângulo que tem os três lados diferentes.
13. Faça uma função chamada `DesenhaLinha`. Ele deve desenhar uma linha na tela usando vários símbolos de igual (Ex: =====). A função recebe por parâmetro quantos sinais de igual serão mostrados.
 14. Faça uma função que receba dois números inteiros positivos por parâmetro e retorne a soma dos N números inteiros existentes entre eles.
 15. Faça uma função que receba por parâmetro dois valores x e z . Calcule e retorne o resultado de x^z para o programa principal.
 16. Faça uma função que retorne o maior fator primo de um número.
 17. Faça um algoritmo que receba um número inteiro positivo n e calcule o seu fatorial, $n!$.
 18. Escreva uma função para determinar a quantidade de números primos abaixo N .
 19. Crie uma função que receba como parâmetro um valor inteiro e gere como saída n linhas com pontos de exclamação, conforme o exemplo abaixo (para $n = 5$):
!
!!
!!!
!!!!
!!!!!

20. Escreva uma função que gera um triângulo lateral de altura $2*n-1$ e n largura. Por exemplo, a saída para $n = 4$ seria:

```
*
**
***
****
***
**
*
```

21. Escreva uma função que gera um triângulo de altura e lados n e base $2*n-1$. Por exemplo, a saída para $n = 6$ seria:

```
      *
     ***
    *****
   ********
  *********
 *****
*****
```

22. Faça um algoritmo que receba um número inteiro positivo n e calcule o somatório de 1 até n .

23. Faça uma função que receba um número N e retorne a soma dos algarismos de $N!$. Ex: se $N = 4$, $N! = 24$. Logo, a soma de seus algarismos é $2 + 4 = 6$.

24. Faça uma função que receba um vetor de inteiros e retorne o maior valor.
25. Faça uma função que receba um vetor de reais e retorne a média dele.
26. Faça uma função que receba um vetor de inteiros e o preencha com números aleatórios sem repetição.
27. Faça uma função que receba como parâmetro um vetor X de 30 elementos inteiros e retorne, também por parâmetro, dois vetores A e B. O vetor A deve conter os elementos pares de X e o vetor B, os elementos ímpares.
28. Faça uma função que calcule o desvio padrão de um vetor v contendo n números. Desvio Padrão: $\sqrt{\frac{1}{n-1} \sum_{i=1}^n (v[i] - m)^2}$ onde m é a média do vetor.
29. Crie um programa contendo as seguintes funções que recebem um vetor V de n números reais como parâmetro:
- Impressão normal do vetor.
 - Impressão inversa.
 - Função que retorna a média aritmética dos elementos do vetor.
30. Faça uma função que receba uma matriz 4 x 4 e retorne quantos valores maiores do que 10 ela possui.
31. Faça uma função que receba uma matriz de 3 x 3 elementos. Calcule a soma dos elementos que estão acima da diagonal principal.
32. Faça uma função que receba uma matriz de 3 x 3 elementos. Calcule e retorne a soma dos elementos que estão abaixo da diagonal principal.
33. Faça uma função que receba uma matriz de 3 x 3 elementos. Calcule e retorne a soma dos elementos que estão na diagonal principal.
34. Faça uma função que receba uma matriz de 3 x 3 elementos. Calcule e retorne a soma dos elementos que estão na diagonal secundária.
35. Escreva uma função que recebe uma matriz quadrada de ordem N e calcule a sua transposta (se B é a matriz transposta de A então $a_{ij} = b_{ji}$).
36. Faça uma função que verifica se uma matriz quadrada de ordem N é a matriz identidade.
37. Faça uma função que recebe, por parâmetro, uma matriz $A[4][4]$ e retorna a soma dos seus elementos.
38. Faça uma função que recebe, por parâmetro, uma matriz $A[3][3]$ e retorna a soma dos elementos da sua diagonal principal e da sua diagonal secundária.

39. Faça uma função que receba, por parâmetro, uma matriz $A[7][6]$ e uma linha N e retorne a soma dos elementos dessa linha.
40. Faça uma função que receba, por parâmetro, uma matriz $A[7][6]$ e uma coluna N e retorne a soma dos elementos dessa coluna.
41. Faça uma função que receba, por parâmetro, duas matrizes quadradas de ordem N , A e B , e retorne uma matriz C , também por parâmetro, que seja o produto matricial de A e B .
42. Faça uma função que receba, por parâmetro, 2 vetores de 10 elementos inteiros e que calcule e retorne, também por parâmetro, o vetor união dos dois primeiros.
43. Escreva uma função que retorne a primeira posição de uma sub-string dentro de uma string. Caso a sub-string não seja encontrada, a função deve retornar -1.
44. Escreva uma função que compare e retorne verdadeiro, caso uma string seja anagrama da outra, e falso, caso contrário.
45. Crie uma função que calcula o comprimento de uma string e que possui a seguinte assinatura: **void tamanho(char *str, int *strsize).**
46. Crie uma função que compara duas strings e que retorna se elas são iguais ou diferentes.
47. Implemente a função a qual recebe duas strings, **str1** e **str2**, e concatena a string apontada por **str2** a string apontada por **str1**.
48. Implemente a função a qual recebe duas strings, **str1** e **str2**, e um valor inteiro positivo N . A função concatena no máximo mais que N caracteres da string apontada por **str2** a string apontada por **str1** e termina **str1** com **NULL**.
49. Faça uma função que dado um caractere qualquer retorne o mesmo caractere sempre em maiúsculo.
50. Faça uma rotina que receba como parâmetro um vetor de caracteres e seu tamanho. A função deverá ler uma string do teclado, caractere por caractere usando a função `getchar()` até que o usuário digite enter ou o tamanho máximo do vetor seja alcançado.
51. Faça uma função que receba duas strings e retorne a intercalação letra a letra da primeira com a segunda string. A string intercalada deve ser retornada na primeira string.
52. Faça um programa que faça operações simples de frações:
- Crie e leia duas frações p e q , compostas por numerador e denominador.
 - Encontre o máximo divisor comum entre o numerador e o denominador, e simplifique as frações.
 - Apresente a soma, a subtração, o produto e o quociente entre as frações lidas.
- Obs.:** Cria uma função para cada item.
53. Um racional é qualquer número da forma p/q , sendo p inteiro e q inteiro não nulo. É conveniente representar um racional por um registro:

```

~
struct racional{ int p, q;
};

```

Vamos convencionar que o campo q de todo racional é estritamente positivo e que o máximo divisor comum dos campos p e q é 1. Escreva:

- (a) uma função `reduz` que receba inteiros a e b e devolva o racional que representa a/b ;
- (b) uma função `neg` que receba um racional x e devolva o racional $-x$;
- (c) uma função `soma` que receba racionais x e y e devolva o racional que representa a soma de x e y ;
- (d) uma função `mult` que receba racionais x e y e devolva o racional que representa o produto de x por y ;
- (e) uma função `div` que receba racionais x e y e devolva o racional que representa o quociente de x por y ;

54. Considerando a estrutura:

```

struct Ponto{ int
x; int y; };

```

para representar um ponto em uma grade 2D, implemente uma função que indique se um ponto p está localizado dentro ou fora de um retângulo. O retângulo é definido por seus vértices inferior esquerdo $v1$ e superior direito $v2$. A função deve retornar 1 caso o ponto esteja localizado dentro do retângulo e 0 caso contrário. Essa função deve obedecer ao protótipo:

```

int dentroRet (struct Ponto* v1, struct Ponto* v2, struct Ponto* p);

```

55. Considerando a estrutura

```

struct Vetor{
float x; float y;
float z; };

```

para representar um vetor no R^3 , implemente uma função que calcule a soma de dois vetores. Essa função deve obedecer ao protótipo:

```

void soma (struct Vetor* v1, struct Vetor* v2, struct Vetor* res);

```

onde os parâmetros $v1$ e $v2$ são ponteiros para os vetores a serem somados, e o parâmetro res é um ponteiro para uma estrutura vetor onde o resultado da operação deve ser armazenado.

73. Foi realizada uma pesquisa de algumas características físicas de cinco habitantes de certa região. De cada habitante foram coletados os seguintes dados: sexo, cor dos olhos (A – Azuis ou C – Castanhos), cor dos cabelos (L – Louros, P – Pretos ou C – Castanhos) e idade.

- Faça uma função que leia esses dados em um vetor.

~

- Faça uma função que determine a média de idade das pessoas com olhos castanhos e cabelos pretos.
- Faça uma função que determine e devolva ao programa principal a maior idade entre os habitantes.
- Faça uma função que determine e devolva ao programa principal a quantidade de indivíduos do sexo feminino cuja idade esta entre 18 e 35 (inclusive) e que tenham olhos azuis e cabelos louros.