



Exercícios: Funções

1. Crie um programa que leia três números. Para cada número, imprima o dobro. Use uma função que recebe como parâmetro um número inteiro e devolve o seu dobro. O valor calculado deve ser impresso na função principal.
2. Faça um programa receba dois números e execute as seguintes funções:
 - Verificar se o número digitado é positivo ou negativo. Sendo que o valor de retorno será 1 se positivo, 0 se negativo ou -1 se for igual a 0.
Obedeça ao protótipo da função: `int verifica(int num)`.
 - Receber dois números inteiros positivos por parâmetro e retorne a soma dos N números inteiros existentes entre eles.
`int soma_entre(int num1, int num2)`.
3. Faça uma função que receba por parâmetro dois valores X e Z. Calcule e retorne o resultado de X^Z para o programa principal. Atenção não utilize nenhuma função pronta de exponenciação.
4. Crie um programa que receba três valores (obrigatoriamente maiores que zero), representando as medidas dos três lados de um triângulo. Elabore funções que:

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.

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.

5. Foi realizada um 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. Determine, por meio de outra função a média de idade das pessoas com olhos castanhos e cabelos pretos. Mostre esse resultado no programa principal.
 - 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 está entre 18 e 35 (inclusive) e que tenham olhos azuis e cabelos louros.

6. Os dois dígitos de verificação do CPF (constituído de 9 dígitos) são calculados através do algoritmo abaixo:

(a) Etapa 1: cálculo de DV1

- Soma 1: soma dos produtos de cada dígito por um peso de 2 a 10, na ordem inversa (do nono para o primeiro).
- Multiplique a soma 1 por 10 e calcule o resto da divisão do resultado por 11. Se der 10, DV1 é zero, caso contrário o DV1 é o próprio resto.

(b) Etapa 2: cálculo de DV2

- Soma 2: soma dos produtos de cada dígito por um peso de 3 a 11, também na ordem inversa.
- Adicione a Soma 2 ao dobro do DV1, multiplique por 10 e calcule o resto da divisão do resultado por 11. Se der 10, DV2 é zero, caso contrário o DV2 é o próprio resto.

(c) Etapa 3: Multiplique DV1 por 10, some com DV2 e você tem o número de controle do CPF.

Exemplo: para o CPF 398 136 146, temos:

- Etapa 1: $2 \times 6 + 3 \times 4 + 4 \times 1 + 5 \times 6 + 6 \times 3 + 7 \times 1 + 8 \times 8 + 9 \times 9 + 10 \times 3 = 258$
 $2580 \bmod 11 = 6$, portanto, DV1 = 6
- Etapa 2: $3 \times 6 + 4 \times 4 + 5 \times 1 + 6 \times 6 + 7 \times 3 + 8 \times 1 + 9 \times 8 + 10 \times 9 + 11 \times 3 = 299$
 $(299 + 6 \times 2) \times 10 \bmod 11 = 3150 \bmod 11 = 8$, portanto DV2 = 8.

7. Faça uma função chamada 'simplifica' que recebe como parâmetro o numerador e o denominador de uma fração. Esta função deve simplificar a fração recebida dividindo o numerador e o denominador pelo maior fator possível. Por exemplo, a fração 36/60 simplifica para 3/5 dividindo o numerador e o denominador por 12. A função deve modificar as variáveis passadas como parâmetro. Faça uma função main para testar esta função.
Dica: seja M o menor valor entre o numerador e o denominador. Para cada número primo de 2 até M, divida o numerador e o denominador por este primo enquanto for possível.

8. Escrever um programa que obtenha a data atual (ler dia, mês e ano) e exiba-a na tela no formato textual por extenso. A escrita da data por extenso deve ser realizada por um procedimento separado. Exemplo: Data: 01/01/2000, Imprimir: Sábado, 1 de janeiro de 2000.

9. Faça uma função que receba como parâmetro o valor de um ângulo em graus e calcule o valor do seno desse ângulo usando sua respectiva série de Taylor:

$$\sin x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots \text{ for all } x$$

Onde x é o valor do ângulo em radianos. Considerar pi = 3.141593 e n variando de 0 até 5.

10. Faça uma função que receba como parâmetro o valor de um ângulo em graus e calcule o valor do cosseno desse ângulo usando sua respectiva série de Taylor:

$$\cos x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots \text{ for all } x$$

Onde x é o valor do ângulo em radianos. Considerar pi = 3.141593 e n variando de 0 até 5.

11. Faça uma função que receba como parâmetro o valor de um ângulo em graus e calcule o valor do seno hiperbólico desse ângulo usando sua respectiva série de Taylor:

$$\sinh x = \sum_{n=0}^{\infty} \frac{x^{2n+1}}{(2n+1)!} = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \dots \text{ for all } x$$

Onde x é o valor do ângulo em radianos. Considerar pi = 3.141593 e n variando de 0 até 5.

12. Faça uma função que receba como parâmetro o valor de um ângulo em graus e calcule o valor do cosseno hiperbólico desse ângulo usando sua respectiva série de Taylor:

$$\cosh x = \sum_{n=0}^{\infty} \frac{x^{2n}}{(2n)!} = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \dots \text{ for all } x$$

Onde x é o valor do ângulo em radianos. Considerar pi = 3.141593 e n variando de 0 até 5.

13. Faça uma função que retorne o maior fator primo de um número.
14. Faça um algoritmo que receba um número inteiro positivo n e calcule:

- (a) n!
- (b) Somatório de 1 até n

Obs.: Crie uma função para cada letra.

15. 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.

16. 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.
17. Faça uma função chamada `DesenhaLinha`. Ele deve desenhar uma linha na tela usando vários símbolos de igual (Ex: =====). No programa principal execute várias chamadas a essa função. (use um loop)
18. Altere o exercício anterior para que a função aceite um argumento de quantos sinais de igual serão mostrados. Ex: `DesenhaLinha(4)` tem como saída `====`; `DesenhaLinha(10)` tem como saída `=====`. No programa principal execute várias chamadas a essa função. (use um loop)
19. Faça uma função 'Troque', que recebe duas variáveis reais A e B e troca o valor delas (i.e., A recebe o valor de B e B recebe o valor de A).

20. Faça uma função para calcular o número neperiano usando uma série (ver abaixo). A função deve ter como parâmetro o número de termos que serão somados (note que, quanto maior o número, a resposta estará mais próxima do valor e).

$$e = \sum_{n=0}^{\infty} \frac{1}{n!} = \frac{1}{0!} + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \dots$$

21. A função fatorial duplo é definida como o produto de todos os números naturais ímpares de 1 até algum número natural ímpar N.

Assim, o fatorial duplo de 5 é: **5!! = 1 * 3 * 5 = 15**

Faça uma função não-recursiva que receba um número inteiro positivo ímpar N e retorne o fatorial duplo desse número.

22. O fatorial quádruplo de um número N é dado por:

$$\frac{(2n)!}{n!}$$

Faça uma função não-recursiva que receba um número inteiro positivo N e retorne o fatorial quádruplo desse número.

23. O superfatorial de um número N é definida pelo produto dos N primeiros fatoriais de N. Assim, o superfatorial de 4 é **sf(4) = 1! * 2! * 3! * 4! = 288**.

Faça uma função não-recursiva que receba um número inteiro positivo N e retorne o superfatorial desse número.

24. O hiperfatorial de um número N, escrito H(n), é definido por:

$$H(n) = \prod_{k=1}^n k^k = 1^1 \cdot 2^2 \cdot 3^3 \dots (n-1)^{n-1} \cdot n^n$$

Faça uma função não-recursiva que receba um número inteiro positivo N e retorne o hiperfatorial desse número.

25. Um fatorial exponencial é um inteiro positivo N elevado à potência de N-1, que por sua vez é elevado à potência de N-2 e assim em diante. Ou seja:

$$n^{(n-1)^{(n-2)} \dots}$$

Faça uma função não-recursiva que receba um número inteiro positivo N e retorne o fatorial exponencial desse número.

26. Os números tribonacci são definidos pela seguinte recursão:

$$f(n) = 0 \text{ se } n = 0$$

$$f(n) = 0 \text{ se } n = 1$$

$$f(n) = 1 \text{ se } n=2$$

$$f(n) = f(n-1)+f(n-2)+f(n-3) \text{ se } n>2.$$

Faça uma função não-recursiva que receba um número N e retorne o N-ésimo termo da sequência de tribonacci.

27. Os números tetranacci iniciam com quatro termos pré-determinados e a partir daí todos os demais números são obtidos pela soma dos quatro números anteriores. Os primeiros números tetranacci são: 0, 0, 0, 1, 1, 2, 4, 8, 15, 29, 56, 108, 208...

Faça uma função não-recursiva que receba um número N e retorne o N-ésimo termo da sequência de tetranacci.

28. A sequência de Padovan é uma sequência de naturais $P(n)$ definida pelos valores iniciais **$P(0) = P(1) = p(2) = 1$**

e a seguinte relação recursiva

$$P(n) = P(n - 2) + P(n - 3) \text{ se } n \geq 3$$

Alguns valores da sequência são: 1, 1, 1, 2, 2, 3, 4, 5, 7, 9, 12, 16, 21, 28...

Faça uma função não-recursiva que receba um número N e retorne o N -ésimo termo da sequência de Padovan.

29. Os números de Pell são definidos pela seguinte recursão

$$p(n) = 0 \text{ se } n = 0$$

$$p(n) = 1 \text{ se } n = 1$$

$$p(n) = 2p(n - 1) + p(n - 2) \text{ se } n \geq 2.$$

Alguns números desta sequência são: 0, 1, 2, 5, 12, 29, 70, 169, 408, 985...

Faça uma função não-recursiva que receba um número N e retorne o N -ésimo número de Pell.

30. Os números de Catalan são definidos pela seguinte recursão

$$c(n) = 1 \text{ se } n = 0$$

$$c(n) = \frac{2(2n-1)}{n+1} C(n - 1) \text{ se } n > 0$$

Alguns números desta sequência são: 1, 1, 2, 5, 14, 42, 132, 429, 1430, 4862, 16796, 58786...

Faça uma função não-recursiva que receba um número N e retorne o N -ésimo número de Catalan.

31. 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.

32. Uma palavra de Fibonacci é definida por

$$f(n) = b \text{ se } n=0$$

$$f(n) = a \text{ se } n = 1$$

$$f(n-1)+f(n-2) \text{ se } n \geq 2$$

Aqui $+$ denota a concatenação de duas strings. Esta sequência inicia com as seguintes palavras: b , a , ab , aba , $abaab$, $abaababa$, $abaababaabaab$, ...

Faça uma função não-recursiva que receba um número N e retorne a N -ésima palavra de Fibonacci.

33. Construa um programa que receba da linha de comando, com a qual o programa é executado, dois operandos e um operador na notação infixada. O programa deve efetuar a operação de forma adequada, ou seja, levando em consideração os operandos envolvidos. Exemplo (no caso do executável se chamar `calcule`):

`calcule 7.2 + 3` retorna 10.2

`calcule 3 / -4` retorna 0

`calcule 3 / +4.0` retorna 0.75

`calcule 7 % 2.0` Operação inválida!

`calcule 7 g 2.0` Operação inválida!

calcule $7 * 2$ Segundo operando inválido!
calcule $.62 - 4$ Primeiro operando inválido!

34. Faça uma função e um programa de teste para o cálculo do volume de uma esfera. Sendo que o raio é passado por parâmetro.

$$V = 4/3 * \pi * R^3$$

35. Crie um programa que contenha uma função que receba como parâmetro um valor inteiro e gere como saída n linhas com pontos de exclamação, conforme o exemplo abaixo (para $n = 5$):

```
!  
!!  
!!!  
!!!!  
!!!!!
```

36. Crie um programa contendo as seguintes funções que recebem um vetor V de 10 números reais como parâmetro:

- Impressão normal do vetor.

```
{void imprime_normal (float vet[10])
```

- Impressão inversa.

```
void imprime_inversa (float vet[10])
```

- Função que retorna a média aritmética dos elementos do vetor.

```
float media_aritmetica(float vet[10])\\
```

Observação: Tente obedecer ao protótipo das funções e para essas funções faça um programa.

37. Elabore um programa contendo 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. A média calculada deverá ser devolvida ao programa principal para então ser impressa na tela.

38. Faça um programa contendo 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. O resultado deverá ser mostrado no programa principal.

39. Faça um programa que apresente o menu a seguir, permita ao usuário escolher a opção desejada, receba os dados necessários para executar a operação e mostre o resultado. Verifique a possibilidade de opção inválida. Use funções na escrita do programa. MENU

- 1. Imposto
- 2. Novo salário
- 3. Classificação

- (a) OPCA01 Mostrar a seguinte tabela relativa aos impostos:
 SALARIO PERCENTUAL DE IMPOSTO
 Menor que 500: 5%
 Entre 500 a 850: 15%
 Mais que 850: 20%
- (b) OPCA02 receber o salário bruto do funcionário, e aplicar a dedução de imposto, segundo a tabela acima.
- (c) OPCA03 mostrar a classificação do funcionário, de acordo com as seguintes regras:
- Até 700 reais - mal remunerado;
 - Maior que 700 reais - bem remunerado.
- Caso o usuário selecione a opção 3, mas não tenha selecionado a opção 2, peça a ele para entrar com o valor do salário.

40. Desenvolva um programa em Linguagem C que permita fazer as seguintes operações sobre um vetor que contém no máximo 100 números inteiros positivos. Use funções na escrita do programa. Apresente um menu aos usuários, com as seguintes opções:

- (a) Inserir um elemento em uma posição específica. O usuário informa a posição e o elemento.
- (b) Inserir um elemento no fim do array, ou seja, na última posição livre.
- (c) Inserir um elemento no início do array, ou seja, na primeira posição livre.
- (d) Remover um elemento de uma posição específica. O usuário informa a posição.
- (e) Modificar o valor de um elemento de uma posição específica, fornecida pelo usuário.
- (f) Exibir os elementos do vetor

Lembrete: inserções e remoções podem causar deslocamento dos elementos do array.

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

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

42. 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:

```

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

43. Escreva uma função para determinar a quantidade de números primos abaixo n.
44. Escreva uma função para encontrar a raiz quadrada de um número n usando o método de Newton.
45. Faça uma função que receba 3 números inteiros como parâmetro, representando horas, minutos e segundos, e os converta em segundos.
46. Faça uma função que receba um inteiro N como parâmetro, calcule e retorne o resultado da seguinte série:

$$S = 2/4 + 5/5 + 10/6 + \dots + (N^2 + 1)/(N + 3)$$
47. Escreva uma função que recebe uma matriz real Amxn e determina a sua transposta (se B é a matriz transposta de A então $a_{ij} = b_{ji}$) e depois imprime.
48. Faça uma função que verifica se uma matriz Amxm é a matriz identidade.
49. Faça uma função que recebe, por parâmetro, uma matriz A[5][5] e retorna a soma dos seus elementos
50. Faça uma função que recebe, por parâmetro, uma matriz A[6][6] e retorna a soma dos elementos da sua diagonal principal e da sua diagonal secundária
51. Faça uma função que recebe, por parâmetro, uma matriz A[7][6] e retorna a soma dos elementos da linha 5 e da coluna 3.
52. Faça uma função que receba, por parâmetro, duas matrizes A[4][6] e B[6][4] e retorna uma matriz C, também por parâmetro, que seja o produto matricial de A e B.
53. Faça uma função que recebe, por parâmetro, 2 vetores de 10 elementos inteiros e que calcule e retorne, também por parâmetro, o vetor união dos dois primeiros
54. Escreva uma função que compare e retorne verdadeiro, caso uma seja anagrama da outra, e falso, caso contrario
55. 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 zero
56. Uma sequência de Fibonacci generalizada, de f0 a f1 é definida como fibg(f0, f1, 0), fibg(f0, f1, 1), fibg(f0, f1, 2), ..., onde:

$$\text{fibg}(f0, f1, 0) = f0$$

$$\text{fibg}(f0, f1, 1) = f1$$

$$\text{fibg}(f0, f1, n) = \text{fibg}(f0, f1, n-1) + \text{fibg}(f0, f1, n-2), \text{ se } n \geq 2.$$

Escreva uma função não-recursiva em C para calcular fibg(f0, f1, n).

57. 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;

58. Faça uma função que calcule o desvio padrão de um vetor v contendo n números

$$\text{Desvio Padrão: } \sqrt{\frac{1}{n-1} \sum_{i=0}^n (v[i] - m)^2}$$

onde m é a media do vetor.

59. 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...

60. 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);
```

61. Considerando a estrutura

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

para representar um vetor no R3, 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.

62. Crie uma função que calcula o comprimento de uma string e que possui a seguinte assinatura: **void tamanho(char *str, int *strsize).**

63. Crie uma função que compara duas strings e que possui a seguinte assinatura: **void compara(char *str1, char *str2)**.
64. Implemente a função tamanhoN() a qual recebe duas strings, **str1** e **str2**, e um valor inteiro positivo N. A função concatena não mais que N caracteres da string apontada por str2 à string apontada por str1 e termina str1 com NULL.
65. Faça uma rotina que receba como parâmetro um vetor de caracteres e seu tamanho. A função deverá de 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.
66. Faça uma sub-rotina "Br_UpCase" que dado um caractere qualquer retorne o mesmo caractere sempre em maiúsculo, aceitando inclusive os caracteres acentuados da língua portuguesa (por exemplo: á, é, í, ó, ú, ç, ã, õ, â, ê, ô, â, ü). Faça um programa que leia uma palavra (string) e chame esta sub-rotina para cada um dos caracteres desta palavra, exibindo o resultado após a conversão para maiúsculo.
- Dicas:**
Letra = Palavra[X]; { Copia o caractere índice X da string "Palavra" para a variável tipo char "Letra" } Palavra[5] = Letra; { Copia o conteúdo da variável tipo char "Letra" para o quinto caractere da string armazenada na variável "Palavra" } Lembre-se: A string armazenada em uma variável pode ser menor que o tamanho reservado para ela. O final da string é representado pelo caractere '\0' (fim de string).
67. Faça uma rotina que receba como parâmetro um vetor de caracteres e seu tamanho. A função deverá de 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.