

# Linguagem C

## Lista de Exercícios

### Exercícios de Sequencia

1. Faça um programa em C que converta temperaturas expressas em graus Celsius para graus Fahrenheit. Seu programa deve solicitar a digitação do valor a ser convertido (C). A relação entre graus Celsius e graus Fahrenheit é  $C = 5/9 (F - 32)$ .
2. Faça um programa em C que converta temperaturas expressas em graus Celsius para graus Kelvin. Seu programa deve solicitar a digitação do valor a ser convertido (C). A relação entre graus Celsius e graus Kelvin é  $C = K - 273,15$ .
3. Faça um programa em C que converta temperaturas expressas em graus Celsius para graus Kelvin. Seu programa deve solicitar a digitação do valor a ser convertido (C). A relação entre graus Celsius e graus Rankine é  $C = (R/1.8) - 273,15$ .
4. Faça um programa em C que calcula a área em centímetros quadrados de um triângulo. Seu programa deve solicitar a digitação da medida em centímetros da base (B) e da altura (A) do triângulo. A relação entre essas grandezas é  $Area = (B * A) / 2$ .
5. Faça um programa em C que calcula a área em centímetros quadrados de um quadrado. Seu programa deve solicitar a digitação da medida em centímetros do lado (L) do quadrado. A relação entre essas grandezas é  $Area = L^2$ .
6. Faça um programa em C que calcula a área em centímetros quadrados de um retângulo. Seu programa deve solicitar a digitação da medida em centímetros do lado menor (m) e do lado maior (M) do retângulo. A relação entre essas grandezas é  $Area = m * M$ .
7. Faça um programa em C que calcula a área em centímetros quadrados de um losango. Seu programa deve solicitar a digitação da medida em centímetros da diagonal menor (d) e da diagonal maior (D) do losango. A relação entre essas grandezas é  $Area = (d * D) / 2$ .
8. Faça um programa em C que calcula a área em centímetros quadrados de um paralelogramo. Seu programa deve solicitar a digitação da medida em centímetros da base (B) e da altura (A) do paralelogramo. A relação entre essas grandezas é  $Area = B * A$ .
9. Faça um programa em C que calcula a área em centímetros quadrados de um trapézio. Seu programa deve solicitar a digitação da medida em centímetros da base menor (b), da base maior (B) e da altura (A) do trapézio. A relação entre essas grandezas é  $Area = ((b+B) * A) / 2$ .
10. Faça um programa em C que calcula a área em centímetros quadrados de um polígono regular. Seu programa deve solicitar a digitação da quantidade de lados (Q) do polígono, bem como da medida em

centímetros de sua base (B) e de sua apótema (A), ou seja, a reta imaginária que une seu centro ao meio de sua base. A relação entre essas grandezas é  $\text{Area} = (Q.B.A)/2$ .

11. Faça um programa em C que calcula a área em centímetros quadrados de um círculo. Seu programa deve solicitar a digitação da medida em centímetros do raio (R) do círculo. A relação entre essas grandezas é  $\text{Area} = \pi R^2$ , sendo  $\pi$  constante e aproximadamente igual a 3,1415.
12. Faça um programa em C que calcula o Índice de Massa Corpórea (ou BMI, Body Mass Index) de uma pessoa. Seu programa deve solicitar a digitação do peso em kilogramas da pessoa, bem como de sua altura em centímetros. O BMI é dado pelo peso dividido pelo quadrado da altura.

## Exercícios de if

13. Faça um programa em C que solicite a digitação de dois valores quaisquer, informando-os, em seguida, em ordem crescente.
14. Faça um programa em C que solicite a digitação de três valores quaisquer, informando-os, em seguida, em ordem crescente.
15. Faça um programa em C que solicite a digitação de quatro valores quaisquer, informando-os, em seguida, em ordem crescente.
16. Faça um programa em C que converta temperaturas expressas em graus Celsius para graus Fahrenheit (operação nº 1) e vice versa (operação nº 2), temperaturas expressas em graus Celsius para graus Kelvin (operação nº 3) e vice versa (operação nº 4), temperaturas em graus Celsius para graus Rankine (operação nº 5) e vice versa (operação nº 6). Seu programa deve solicitar a digitação da operação desejada, bem como do valor a ser convertido. A relação entre as unidades de medida envolvidas pode ser encontrada nos exercícios de 1 a 3.
17. Faça um programa em C que solicite a digitação de três valores representando, respectivamente, as horas, os minutos e os segundos de um horário, verificando, a seguir se os mesmos representam ou não um horário válido.
18. Faça um programa em C que solicite a digitação de três valores representando, respectivamente, o dia, o mês e o ano de uma data, verificando, a seguir se os mesmos representam ou não uma data válida (lembrar dos meses de 30 e 31 dias, bem como do mês de fevereiro e seus 28 ou 29 dias; anos bissextos são anos divisíveis por 4 mas não por 100, ou então anos divisíveis por 400).
19. Faça um programa em C que solicite a digitação da medida em centímetros de três retas, verifique se as mesmas podem ou não formar um triângulo, escrevendo, em seguida, na tela, o resultado dessa verificação.
20. Faça um programa em C que solicite a digitação da medida em centímetros de três retas, escrevendo, em seguida, na tela, quando for possível com elas formar um triângulo, a classificação desse triângulo (equilátero = triângulo com todos os lados iguais; isósceles = triângulo com exatamente 2 lados iguais; ou escaleno = triângulo com todos os lados diferentes).
21. Faça um programa em C que solicite a digitação da medida em graus de três ângulos, verifique se os mesmos podem ou não ser os ângulos internos de um triângulo, escrevendo, em seguida, na tela, o resultado dessa verificação.
22. Faça um programa em C que solicite a digitação do tamanho em graus dos ângulos internos de um triângulo, escrevendo, em seguida, na tela, quando for possível um triângulo com tais ângulos internos, a classificação desse triângulo (acutângulo = triângulo com todos os ângulos agudos, ou seja, menores que

90°; retângulo = triângulo com um ângulo reto, ou seja, valendo 90°; ou obtusângulo = triângulo com um ângulo obtuso, ou seja, maior que 90°).

23. Faça um programa em C que solicite a digitação dos coeficientes a e b de uma equação de 1º grau, informando a raiz da equação.
24. Faça um programa em C que solicite a digitação dos coeficientes a, b e c de uma equação de 2º grau, informando se a equação não possui raízes, a única raiz da equação, quando esse for o caso, as duas raízes da equação.
25. Faça um programa em C que solicite a digitação de um número natural até 9, escrevendo então na tela o valor fornecido por extenso.
26. Faça um programa em C que solicite a digitação de um número natural entre -9 e 9, escrevendo então na tela o valor fornecido por extenso. Faça o programa de forma inteligente, procurando reduzir a quantidade de comandos printf.
27. Faça um programa em C que solicite a digitação de um número natural entre -99 e 99, escrevendo então na tela o valor fornecido por extenso. Faça o programa de forma inteligente, procurando reduzir a quantidade de comandos printf.
28. Faça um programa em C que solicite a digitação de um número natural entre -999 e 999, escrevendo então na tela o valor fornecido por extenso. Faça o programa de forma inteligente, procurando reduzir a quantidade de comandos printf.
29. Faça um programa em C que solicite a digitação de um valor monetário entre R\$ -9,99 e R\$ 9,99, escrevendo então na tela o valor fornecido por extenso. Faça o programa de forma inteligente, procurando reduzir a quantidade de comandos printf. Usar as palavras “real”, “reais”, “centavo” e “centavos” de forma apropriada. Não escrever “zero reais” e nem “zero centavos”, exceto no caso do saldo ser igual a zero.

## Exercícios de switch

Reescreva em C os exercícios de `if` acima, agora usando, quando apropriado, o comando `switch`. Exercícios nos quais o comando `switch` não seria cabível não precisam ser refeitos.

## Exercícios de Repetição

Quando cabível, resolva os exercícios abaixo uma vez, usando como comando repetitivo apenas o comando `while`, outra vez, usando como comando repetitivo apenas o comando `do-while` e uma última vez, usando como comando repetitivo apenas o comando `for`.

30. Escreva um programa em C que exibe os termos entre 1 e 1000 (ambos inclusive) da sequência oscilante com primeiro termo igual a 1 e demais termos obtidos alternadamente somando 4 e subtraindo 2 do termo anterior.
31. Escreva um programa em C que calcula e exibe a soma dos divisores de um número natural dado.
32. Escreva um programa em C que solicita a digitação de um número natural, verifica se o mesmo é primo (número divisível apenas por 1 e por si próprio), exibindo na tela uma mensagem diagnóstica.
33. Escreva um programa em C que solicita a digitação dos limites de um intervalo de números naturais, exibindo então na tela todos os números naturais do intervalo (incluindo os limites) que são primos.

34. Escreva um programa em C que solicita a digitação de um número natural, verifica se o mesmo é perfeito (número que é igual à soma de seus divisores distintos de si próprio), exibindo na tela uma mensagem diagnóstica (6 é; 10 não é).
35. Escreva um programa em C que solicita a digitação dos limites de um intervalo de números naturais, exibindo então na tela todos os números naturais do intervalo (incluindo os limites) que são perfeitos.
36. Escreva um programa em C que solicita a digitação de dois números naturais, verifica se o mesmo são amigos (ambos tem a soma de seus divisores distintos de si próprio igual ao outro), exibindo na tela uma mensagem diagnóstica (220 e 284 são; 123 e 321 não são).
37. Escreva um programa em C que solicita a digitação dos limites de um intervalo de números naturais, exibindo então na tela todos os pares de números naturais do intervalo (incluindo os limites) que são amigos.
38. Escreva em C um programa que calcula e mostra o número natural que é o inverso (tem os mesmos dígitos, porém em ordem reversa) de um número natural dado.
39. Escreva um programa em C que solicita a digitação de um número natural, verifica se o mesmo é palíndromo (número cujos dígitos lidos da esquerda para direita ou da direita para a esquerda se equivalem), exibindo na tela uma mensagem diagnóstica.
40. Escreva um programa em C que solicita a digitação de dois números naturais, rodando os bits do primeiro número digitado para a esquerda tantas vezes quanto expressar o segundo número digitado, exibindo na tela o número natural resultante. Assuma que o número rodado é um número de 4 bytes.
41. Escreva um programa em C que solicita a digitação de dois números naturais, rodando os bits do primeiro número digitado para a direita tantas vezes quanto expressar o segundo número digitado, exibindo na tela o número natural resultante. Assuma que o número rodado é um número de 4 bytes.
42. Escreva um programa em C que solicita a digitação de dois números naturais, tornando 1 no primeiro número digitado o bit expresso pelo segundo número digitado, exibindo então na tela o número natural resultante. Assuma que o número que tem o bit ajustado é um número de 4 bytes.
43. Escreva um programa em C que solicita a digitação de dois números naturais, tornando 0 no primeiro número digitado o bit expresso pelo segundo número digitado, exibindo então na tela o número natural resultante. Assuma que o número que tem o bit ajustado é um número de 4 bytes.

## Exercícios de Subprograma

Tendo conhecimento do conceito de subprograma, refaça todos os exercícios acima, empregando, da melhor forma possível, este conceito. Faça programas em C para testar os subprogramas.

## Exercícios de Macro

44. Escreva uma macro em C que troca o conteúdo de duas variáveis inteiras. Dica: blocos de comandos podem ajudar. Faça um programa para testar sua macro.
45. Escreva uma macro em C que retorna o maior de 3 números fornecidos. Dica: expressões condicionais podem ajudar. Faça um programa para testar sua macro.

# Exercícios de Recursão

46. Faça uma função em C que, recebendo como parâmetro um único número inteiro, resulta 1, caso o mesmo seja negativo, ou 0, caso contrário.

Na implementação de sua função, fica proibido declarar variáveis, fazer uso de operadores aritméticos (+, -, \*, / e %), dos operadores relacionais (!=, <, <=, > e >=), de operadores lógicos (!, && e ||), dos comandos de seleção (if com else e switch), de comandos repetitivos (while, do-while e for) e de qualquer função de biblioteca.

Dica: use ++, --, if sem else, return, recursão e função auxiliar sujeita às mesmas restrições acima.

Faça um programa em C para testar sua função.

47. Faça uma função em C que, recebendo como parâmetro dois números inteiros, resulta a soma deles.

Importante lembrar que os números com os quais vocês trabalharão são inteiros, logo, podem ser positivos ou negativos

Fica proibido declarar variáveis, fazer uso de operadores aritméticos (+, -, \*, / e %), dos operadores relacionais (!=, <, <=, > e >=), de operadores lógicos (!, && e ||), dos comandos de seleção (if com else e switch), de comandos repetitivos (while, do-while e for), de funções auxiliares e de qualquer função de biblioteca.

Dica: use ++, --, if sem else, return, recursão e as funções feitas a partir do exercício 22.

Faça um programa em C para testar sua função.

48. Faça uma função em C que, recebendo como parâmetro dois números inteiros, resulta a subtração deles (o primeiro menos o segundo).

Importante lembrar que os números com os quais vocês trabalharão são inteiros, logo, podem ser positivos ou negativos

Fica proibido declarar variáveis, fazer uso de operadores aritméticos (+, -, \*, / e %), dos operadores relacionais (!=, <, <=, > e >=), de operadores lógicos (!, && e ||), dos comandos de seleção (if com else e switch), de comandos repetitivos (while, do-while e for), de funções auxiliares e de qualquer função de biblioteca.

Dica: use ++, --, if sem else, return, recursão e as funções feitas a partir do exercício 22.

Faça um programa em C para testar sua função.

49. Faça uma função em C que, recebendo como parâmetro dois números inteiros, resulta a multiplicação deles.

Importante lembrar que os números com os quais vocês trabalharão são inteiros, logo, podem ser positivos ou negativos

Fica proibido declarar variáveis, fazer uso de operadores aritméticos (+, -, \*, / e %), dos operadores relacionais (!=, <, <=, > e >=), de operadores lógicos (!, && e ||), dos comandos de seleção (if com else e switch), de comandos repetitivos (while, do-while e for), de funções auxiliares e de qualquer função de biblioteca.

Dica: use ++, --, if sem else, return, recursão e as funções feitas a partir do exercício 22.

Faça um programa em C para testar sua função.

50. Faça uma função em C que, recebendo como parâmetro dois números inteiros, resulta a divisão inteira deles (o primeiro dividido pelo segundo).

Importante lembrar que os números com os quais vocês trabalharão são inteiros, logo, podem ser positivos ou negativos

Fica proibido declarar variáveis, fazer uso de operadores aritméticos (+, -, \*, / e %), dos operadores relacionais (!=, <, <=, > e >=), de operadores lógicos (!, && e ||), dos comandos de seleção (if com else e switch), de comandos repetitivos (while, do-while e for), de funções auxiliares e de qualquer função de biblioteca.

Dica: use ++, --, if sem else, return, recursão e as funções feitas a partir do exercício 22.

Faça um programa em C para testar sua função.

51. Faça uma função em C que, recebendo como parâmetro dois números inteiros, resulta o resto da divisão inteira deles (o resto da divisão inteira do primeiro deles pelo segundo).

Importante lembrar que os números com os quais vocês trabalharão são inteiros, logo, podem ser positivos ou negativos

Fica proibido declarar variáveis, fazer uso de operadores aritméticos (+, -, \*, / e %), dos operadores relacionais (!=, <, <=, > e >=), de operadores lógicos (!, && e ||), dos comandos de seleção (if com else e switch), de comandos repetitivos (while, do-while e for), de funções auxiliares e de qualquer função de biblioteca.

Dica: use ++, --, if sem else, return, recursão e as funções feitas a partir do exercício 22.

Faça um programa em C para testar sua função.

## Exercícios de *String*

52. Escreva uma função em C que resulta o número natural representado, em binário, pela cadeia de caracteres fornecida (e que deve permanecer intacta e que deve conter somente caracteres que correspondem aos dígitos binários válidos). Faça um programa em C para testar sua função.
53. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.
54. Escreva uma função em C que resulta uma cadeia de caracteres que contém somente caracteres que correspondem aos dígitos binários válidos e que representa, em binário, o número natural fornecido. Faça um programa em C para testar sua função.
55. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.
56. Escreva uma função em C que resulta o número natural representado, em octal, pela cadeia de caracteres fornecida (e que deve permanecer intacta e que deve conter somente caracteres que correspondem aos dígitos octais válidos). Faça um programa em C para testar sua função.
57. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.
58. Escreva uma função em C que resulta uma cadeia de caracteres que contém somente caracteres que correspondem aos dígitos octais válidos e que representa, em octal, o número natural fornecido. Faça um programa em C para testar sua função.
59. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.
60. Escreva uma função em C que resulta o número natural representado, em decimal, pela cadeia de caracteres fornecida (e que deve permanecer intacta e que deve conter somente caracteres que correspondem aos dígitos decimais válidos). Faça um programa em C para testar sua função.
61. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.
62. Escreva uma função em C que resulta uma cadeia de caracteres que contém somente caracteres que correspondem aos dígitos decimais válidos e que representa, em decimal, o número natural fornecido. Faça um programa em C para testar sua função.
63. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.

64. Escreva uma função em C que resulta o número natural representado, em hexadecimal, pela cadeia de caracteres fornecida (e que deve permanecer intacta e que deve conter somente caracteres que correspondem aos dígitos hexadecimais válidos). Faça um programa em C para testar sua função.
65. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.
66. Escreva uma função em C que resulta uma cadeia de caracteres que contém somente caracteres que correspondem aos dígitos hexadecimais válidos e que representa, em hexadecimal, o número natural fornecido. Faça um programa em C para testar sua função.
67. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.
68. Escreva uma função em C que resulta o número natural representado, na base numérica dada, pela cadeia de caracteres fornecida (e que deve permanecer intacta e que deve conter somente caracteres que correspondem aos dígitos válidos na base numérica dada). Faça um programa em C para testar sua função.
69. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.
70. Escreva uma função em C que resulta uma cadeia de caracteres que contém somente caracteres que correspondem aos dígitos válidos na base numérica dada e que representa, na base numérica em questão, o número natural fornecido. Faça um programa em C para testar sua função.
71. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.
72. Escreva uma função em C que resulta o número inteiro representado pela cadeia de caracteres fornecida (e que deve permanecer intacta). Faça um programa em C para testar sua função.
73. Escreva uma função em C que resulta uma cadeia de caracteres que representa o número inteiro fornecido. Faça um programa em C para testar sua função.
74. Escreva uma função em C que resulta o número real representado pela cadeia de caracteres fornecida (e que deve permanecer intacta). A função deve lidar também com números em notação científica. Faça um programa em C para testar sua função.
75. Escreva uma função em C que resulta uma cadeia de caracteres que representa o número real fornecido. A função deve lidar também com números em notação científica. Faça um programa em C para testar sua função.
76. Escreva um procedimento em C que converte eventuais caracteres invisíveis que compõem uma cadeia de caracteres dada em seqüências de \, e.g., <TAB> pelo caractere \ seguindo pelo caractere t. Faça um programa em C para testar sua função.
77. Escreva um procedimento em C que elimine os espaços em branco supérfluos de uma cadeia de caracteres fornecida, i.e., que retire deela os espaços em branco iniciais e finais, e que substitua nela seqüências de dois ou mais espaços em branco por apenas um espaço em branco. Faça um programa em C para testar sua função.
78. Escreva um procedimento em C que manipule uma cadeia de caractere dada e expanda subcadeias da forma "<Char<sub>0n01n- 79. Escreva um procedimento em C que converta letras minúsculas, que eventualmente componham uma cadeia de caracteres fornecida, em letras maiúsculas. Faça um programa em C para testar sua função.
- 80. Rescreva em C o procedimento acima, usando expressões condicionais (? :). Faça um programa em C para testar sua função.</sub>

81. Escreva um procedimento em C que inverta uma cadeia de caracteres dada. Faça um programa em C para testar sua função.
82. Escreva em C uma função que resulta o inverso de uma cadeia de caracteres dada. Faça um programa em C para testar sua função.
83. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função. Faça um programa em C para testar sua função.
84. Escreva uma função em C para verificar se uma cadeia de caracteres fornecida é palíndrome, resultando 1 em caso afirmativo, ou zero, caso contrário. Faça um programa em C para testar sua função.
85. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.
86. Escreva um procedimento em C que troque os caracteres <TAB> de uma cadeia de caracteres dada por 8 espaços em branco. Faça um programa em C para testar sua função.
87. Escreva uma função em C que resulta uma cadeia de caracteres em tudo idêntica à fornecida, exceto pelo fato de que nela todos os caracteres <TAB> serão substituídos por 8 espaços em branco. Faça um programa em C para testar sua função.
88. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.
89. Escreva um procedimento em C que troque subcadeias de 8 espaços em branco de uma cadeia de caracteres fornecida por um caractere <TAB>. Faça um programa em C para testar sua função.
90. Escreva uma função em C que resulta uma cadeia de caracteres em tudo idêntica à fornecida, exceto pelo fato de que nela toda subcadeia de 8 espaços em branco serão substituídos por um caractere <TAB>.
91. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.
92. Escreva uma função em C que resulte a posição mais a esquerda de um dado caractere em uma cadeia de caracteres fornecida. Faça um programa em C para testar sua função.
93. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.
94. Escreva uma função em C que resulte a posição mais a direita de um dado caractere em uma cadeia de caracteres fornecida. Faça um programa em C para testar sua função.
95. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.
96. Escreva uma função em C que resulte quantas vezes um dado caractere ocorre em uma cadeia de caracteres fornecida. Faça um programa em C para testar sua função.
97. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.
98. Escreva uma função em C resulte a posição mais a esquerda de uma dada cadeia de caracteres que contem um dos caracteres de uma cadeia de caracteres também fornecida. Faça um programa em C para testar sua função.
99. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.
100. Escreva uma função em C resulte a posição mais a direita de uma dada cadeia de caracteres que contem um dos caracteres de uma cadeia de caracteres também fornecida. Faça um programa em C para testar sua função.
101. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.
102. Escreva uma função em C resulte quantas vezes uma dada cadeia de caracteres contem um dos caracteres de uma cadeia de caracteres também fornecida. Faça um programa em C para testar sua função.



103. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.
104. Escreva uma função em C que resulte a posição mais a esquerda de uma dada subcadeia em uma cadeia de caracteres fornecida. Faça um programa em C para testar sua função.
105. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.
106. Escreva uma função em C que resulte a posição mais a direita de uma dada subcadeia em uma cadeia de caracteres fornecida. Faça um programa em C para testar sua função.
107. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.
108. Escreva uma função em C que resulta quantas vezes uma cadeia de caracteres fornecida contém uma dada subcadeia. Faça um programa em C para testar sua função.
109. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.
110. Escreva um procedimento em C para remover todas as ocorrências de um certo caractere dado de uma cadeia de caracteres fornecida. Faça um programa em C para testar sua função.
111. Escreva uma função em C que resulte uma cadeia de caracteres em tudo idêntica a uma cadeia de caracteres fornecida, exceto pelo fato de que nela todas as ocorrências de um caractere dado teriam sido removidas. Faça um programa em C para testar sua função.
112. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.
113. Escreva um procedimento em C para remover todas as ocorrências dos caracteres que compõem a cadeia de caracteres fornecida de uma dada cadeia de caracteres que também seria fornecida. Faça um programa em C para testar sua função.
114. Escreva uma função em C que resulte uma cadeia de caracteres em tudo idêntica a uma cadeia de caracteres fornecida, exceto pelo fato de que dela teriam sido removidas todas as ocorrências dos caracteres que compõem uma cadeia de caracteres que também seria fornecida. Faça um programa em C para testar sua função.
115. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.
116. Escreva um procedimento em C para remover de uma cadeia de caracteres dada todas as ocorrências de uma certa subcadeia que também seria fornecida. Faça um programa em C para testar sua função.
117. Escreva uma função em C que resulte uma cadeia de caracteres em tudo idêntica a uma cadeia de caracteres fornecida, exceto pelo fato de que nela todas as ocorrências de uma subcadeia também fornecida teriam sido removidas. Faça um programa em C para testar sua função.
118. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.

## Exercícios de *String* alocado dinamicamente

Reescreva todas as funções solicitadas nos exercícios da seção Exercícios de String acima que retornam cadeias de caracteres, de forma a retornar cadeias de caracteres alocadas dinamicamente. Faça programas em C para testar as funções refeitas.

# Exercícios de Vetor

119. Escreva uma função em C que resulte a soma dos elementos de um vetor de números inteiros dado (a quantidade de elementos do vetor também será fornecida).
120. Escreva uma função em C que resulte a média dos elementos de um vetor de números inteiros dado (a quantidade de elementos do vetor também será fornecida).
121. Escreva uma função em C que resulte o maior elemento de um vetor de números inteiros dado (a quantidade de elementos do vetor também será fornecida).
122. Escreva uma função em C que resulte quantos elementos de um vetor de números inteiros dado estão abaixo da média dos elementos do mesmo vetor (a quantidade de elementos do vetor também será fornecida).
123. Escreva uma função em C que resulte 1, se todos os elementos de um vetor de números inteiros dado forem par, ou 0, caso contrário.
124. Escreva um procedimento em C que inverta um dado vetor de números inteiros (a quantidade de elementos do vetor também será fornecida). Faça um programa em C para testar sua função.
125. Escreva em C uma função que resulta o inverso de um dado vetor de números inteiros (a quantidade de elementos do vetor também será fornecida). Não vetor resultante não deve ser alocado dinamicamente. Faça um programa em C para testar sua função.
126. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.
127. Escreva uma função em C para verificar se um dado vetor de caracteres está em ordem crescente, resultando 1 em caso afirmativo, ou zero, caso contrário (a quantidade de elementos do vetor também será fornecida). Faça um programa em C para testar sua função.
128. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.
129. Escreva uma função em C que resulte a posição mais a esquerda de um dado número inteiro em um vetor de números inteiros (a quantidade de elementos do vetor também será fornecida). Faça um programa em C para testar sua função.
130. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.
131. Escreva uma função em C que resulte a posição mais a direita de um dado número inteiro em um vetor de números inteiros (a quantidade de elementos do vetor também será fornecida). Faça um programa em C para testar sua função.
132. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.
133. Escreva uma função em C que resulte quantas vezes um dado número inteiro ocorre em um vetor de números inteiros fornecido (a quantidade de elementos do vetor também será fornecida). Faça um programa em C para testar sua função.
134. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.
135. Escreva uma função em C resulte a posição mais a esquerda de uma dado vetor de números inteiros que contem um dos números inteiros de um vetor de números inteiros também fornecido (a quantidade de elementos do vetor também será fornecida). Faça um programa em C para testar sua função.

136. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.
137. Escreva uma função em C resulte a posição mais a direita de uma dado vetor de números inteiros que contem um dos números inteiros de um vetor de números inteiros também fornecido (a quantidade de elementos do vetor também será fornecida). Faça um programa em C para testar sua função.
138. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.
139. Escreva uma função em C resulte quantas vezes um dado vetor de números inteiros contem um dos números inteiros de um vetor de números inteiros também fornecido (a quantidade de elementos do vetor também será fornecida). Faça um programa em C para testar sua função.
140. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.
141. Escreva uma função em C que resulte a posição mais a esquerda de uma dada seqüência de números inteiros presente num vetor de números inteiros dado (a quantidade de elementos do vetor também será fornecida) em um outro vetor (maior) de números inteiros que será fornecido (a quantidade de elementos desse outro vetor também será fornecida). Faça um programa em C para testar sua função.
142. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.
143. Escreva uma função em C que resulte a posição mais a direita de uma dada seqüência de números inteiros presente num vetor de números inteiros dado (a quantidade de elementos do vetor também será fornecida) em um outro vetor (maior) de números inteiros que será fornecido (a quantidade de elementos desse outro vetor também será fornecida). Faça um programa em C para testar sua função.
144. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.
145. Escreva uma função em C que resulte quantas vezes um dado vetor de números inteiros (a quantidade de elementos do vetor também será fornecida) contem uma seqüência de números inteiros em um outro vetor (menor) de números inteiros que será fornecido (a quantidade de elementos desse outro vetor também será fornecida). Faça um programa em C para testar sua função.
146. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.
147. Escreva um procedimento em C para remover todas as ocorrências de um certo número inteiro dado de um vetor de números inteiros fornecida (a quantidade de elementos desse outro vetor também será fornecida E DEVERÁ SER ATUALIZADA). Faça um programa em C para testar sua função.
148. Escreva um procedimento em C para remover todas as ocorrências dos números inteiros que compõem a seqüência de números inteiros em um dado vetor de números inteiros (a quantidade de elementos do vetor também será fornecida) de um vetor de números inteiros (maior) que também será fornecido (a quantidade de elementos desse outro vetor também será fornecida E DEVERÁ SER ATUALIZADA). Faça um programa em C para testar sua função.
149. Escreva um procedimento em C para remover de um vetor de números inteiros dado (a quantidade de elementos do vetor também será fornecida) todas as ocorrências de uma certa seqüência de números inteiros em um outro vetor de números inteiros que também sera

fornecido (a quantidade de elementos desse outro vetor também será fornecida). Faça um programa em C para testar sua função.

## Exercícios de Vetor alocado dinamicamente

Reescreva em C todas as funções solicitadas nos exercícios da seção Exercícios de Vetor acima que retornam vetores, de forma a retornar vetores alocados dinamicamente. Faça programas em C para testar as funções refeitas.

## Exercícios de Matriz

150. Escreva uma função em C que resulte a soma dos elementos de uma dada matriz de até 100x100 números reais (a quantidade de linhas e de colunas da área útil da matriz também será fornecida). Faça um programa em C para testar sua função.
151. Escreva uma função em C que resulte a soma dos elementos da diagonal principal de uma dada matriz quadrada de até 100x100 números reais (a dimensão área útil da matriz também será fornecida). Faça um programa em C para testar sua função.
152. Escreva uma função em C que resulte a média geométrica dos elementos de uma dada matriz de até 100x100 números reais (a quantidade de linhas e de colunas da área útil da matriz também será fornecida). Faça um programa em C para testar sua função.
153. Escreva uma função em C que resulte a razão entre a média aritmética dos elementos da diagonal principal e a média dos elementos da diagonal secundária de uma dada matriz quadrada de até 100x100 números reais (a dimensão da área útil da matriz também será fornecida). Faça um programa em C para testar sua função.
154. Escreva uma função em C que resulte o maior elemento da diagonal secundária de uma matriz quadrada de até 100x100 números reais (a dimensão da área útil da matriz também será fornecida). Faça um programa em C para testar sua função.
155. Escreva uma função em C que resulte o menor elemento de uma dada matriz quadrada de até 100x100 números reais (a dimensão da área útil da matriz também será fornecida). Faça um programa em C para testar sua função.
156. Dadas duas matrizes de até 100x100 números reais (a quantidade de linhas e colunas da área útil da primeira, que é igual em dimensão à área útil da segunda, também serão fornecidas), escreva um procedimento em C que as some, dispondo o resultado em uma dada terceira matriz de até 100x100 números reais e com com área útil com a mesma dimensão que a das duas anteriores. Faça um programa em C para testar sua função.
157. Dadas duas matrizes de até 100x100 números reais (a quantidade de linhas e colunas da área útil da primeira, que é igual em dimensão à área útil da segunda, também serão fornecidas), escreva uma função em C que resulte a soma das duas (uma matriz de até 100x100 números reais e com com área útil com a mesma dimensão que a das duas matrizes fornecidas). Faça um programa em C para testar sua função.
158. Dadas duas matrizes de até 100x100 números reais (a quantidade de linhas e colunas da área útil da primeira, que é igual em dimensão à área útil da segunda, também serão fornecidas), escreva um procedimento em C que da primeira subtraia a segunda, dispondo o resultado em uma dada terceira matriz de até 100x100 números reais e com com área útil com a mesma dimensão que a das duas anteriores. Faça um programa em C para testar sua função.
159. Dadas duas matrizes de até 100x100 números reais (a quantidade de linhas e colunas da área útil da primeira, que é igual em dimensão à área útil da segunda, também serão fornecidas),

escreva uma função em C que resulte a diferença das duas (uma matriz de até 100x100 números reais e com com área útil com a mesma dimensão que a das duas fornecidas). Faça um programa em C para testar sua função.

160. Dadas duas matrizes de até 100x100 números reais (a quantidade de linhas da área útil da primeira, que é igual à quantidade de colunas da área útil da segunda, e a quantidade de colunas da área útil da primeira, que é igual à quantidade de linhas da área útil da segunda, também serão fornecidas), escreva um procedimento em C que multiplica a primeira pela segunda, dispondo o resultado em uma dada terceira matriz de até 100x100 números reais e com com área útil quadrada com dimensão igual a quantidade de linhas da primeira, que é igual à quantidade de colunas da segunda. Faça um programa em C para testar sua função.
161. Dadas duas matrizes de até 100x100 números reais (a quantidade de linhas da área útil da primeira, que é igual à quantidade de colunas da área útil da segunda, e a quantidade de colunas da área útil da primeira, que é igual à quantidade de linhas da área útil da segunda, também serão fornecidas), escreva uma função em C que resulta a multiplicação das duas (uma matriz de até 100x100 números reais e com com área útil quadrada com dimensão igual a quantidade de linhas da primeira, que é igual à quantidade de colunas da segunda). Faça um programa em C para testar sua função.

## Exercícios de Matriz alocada dinamicamente

Reescreva em C todas as funções solicitadas nos exercícios da seção Exercícios de Matriz acima que retornam matrizes, de forma a retornar matrizes alocadas dinamicamente. Faça programas em C para testar as funções refeitas.

## Exercícios de struct

Considere declarado o seguinte tipo:

```
typedef
struct Aluno
{
    char ra        [ 9],
        nome      [ 41],
        endereço  [101],
        cidade    [ 31],
        uf        [ 3],
        fone      [ 11],
        cel       [ 11],
        email     [ 41];

    unsigned int idade;
    double      cd;
}
Aluno;
```

162. Escreva um procedimento em C para incluir um dado aluno em um vetor de Aluno fornecido (a quantidade de elementos do vetor também será fornecida E DEVERÁ SER ATUALIZADA). Considere que os alunos estão em ordem crescente de ra no vetor e que essa ordem deve ser mantida pela inclusão. Faça um programa em C para testar seu procedimento.

163. Escreva uma função em C para verificar se o aluno que tem um dado ra consta em um vetor de Aluno fornecido (a quantidade de elementos do vetor também será fornecida). Sua função deve resultar 1, caso o aluno procurado seja encontrado, ou 0, caso contrário. Faça um programa em C para testar sua função.
164. Escreva uma função em C para verificar se o aluno que tem um dado ra consta em um vetor de Aluno fornecido (a quantidade de elementos do vetor também será fornecida). Considere que os alunos estão em ordem crescente de ra no vetor e faça busca binária, resultando 1, caso o aluno procurado seja encontrado, ou 0, caso contrário. Faça um programa em C para testar sua função.
165. Escreva um procedimento em C para remover o Aluno que tem um dado ra de um vetor de Aluno fornecido (a quantidade de elementos do vetor também será fornecida E DEVERÁ SER ATUALIZADA). Faça um programa em C para testar seu procedimento.
166. Escreva um procedimento em C para ordenar em ordem crescente de ra um dado vetor de Aluno (a quantidade de elementos do vetor também será fornecida). Faça um programa em C para testar seu procedimento.
167. Escreva uma função em C que resulte quantos alunos menores de idade existem em um dado vetor de Aluno (a quantidade de elementos do vetor também será fornecida). Faça um programa em C para testar sua função.
168. Dado um vetor de Aluno (a quantidade de elementos do vetor também será fornecida), escreva uma função em C que resulte quantos são os alunos que possuem o telefone fornecido. Faça um programa em C para testar sua função.
169. Dado um vetor de Aluno (a quantidade de elementos do vetor também será fornecida), escreva uma função em C que resulte quantos são os alunos de um dado estado que possuem coeficiente de desempenho superior ou igual a um valor (número real) fornecido. Faça um programa em C para testar sua função.
170. Dado um vetor de Aluno (a quantidade de elementos do vetor também será fornecida), escreva uma função em C que resulte quantas diferentes cidades do estado dado possuem alunos na instituição. Faça um programa em C para testar sua função.
171. Dado um vetor de Aluno (a quantidade de elementos do vetor também será fornecida), escreva uma função em C que resulte a média dos coeficientes de desempenho dos alunos de um dado estado. Faça um programa em C para testar sua função.

## Exercícios de Lista Ligada

Considere declarados os seguintes tipos:

```
typedef
    struct No*
    pNo;

typedef
    struct No
    {
        int info;
        pNo prox;
    }
    No;
```

172. Escreva um procedimento em C que, dado um pNo, elimina da lista que nele tem início todas as ocorrências de um dado número inteiro. Faça um programa em C para testar seu procedimento.
173. Escreva uma função em C que resulta a lista que tem início em um dado pNo sem cada uma das ocorrências de um dado número inteiro. Faça um programa em C para testar sua função.
174. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.
175. Escreva um procedimento em C para inverter a lista que tem início em um dado pNo. Faça um programa em C para testar seu procedimento.
176. Escreva uma função em C que resulta a lista que tem início em um dado pNo invertida. Faça um programa em C para testar sua função.
177. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.
178. Escreva um procedimento em C para concatenar a uma lista que tem início em um dado pNo, uma lista que tem início em outro pNo fornecido. Faça um programa em C para testar seu procedimento.
179. Escreva uma função em C que resulta a concatenação de duas listas que têm início dois pNo fornecidos. Faça um programa em C para testar sua função.
180. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.
181. Escreva um procedimento em C para concatenar incluir um dado número inteiro em uma lista que tem início em um pNo fornecido. Faça um programa em C para testar seu procedimento.
182. Escreva uma função em C que resulta a lista ordenada de forma crescente que tem início em um dado pNo com um número inteiro fornecido incluído no local apropriado para manter a ordem crescente. Faça um programa em C para testar sua função.
183. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.
184. Escreva um procedimento em C para ordenar de forma crescente uma lista que tem início em um pNo fornecido. Faça um programa em C para testar seu procedimento.
185. Escreva uma função em C que resulta ordenada a lista que tem início em um dado pNo. Faça um programa em C para testar sua função.
186. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.
187. Escreva uma função em C que compara duas listas que têm início em dados pNo, resultando 1, caso contenham os mesmos números e na mesma ordem, ou 0, caso contrário. Faça um programa em C para testar sua função.
188. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.
189. Escreva um procedimento em C para intercalar em uma lista ordenada de forma crescente que tem início em um dado pNo, os elementos de uma segunda lista ordenada de forma crescente que tem início em um pNo fornecido. Faça um programa em C para testar seu procedimento.

190. Escreva uma função em C que resulta um pNo que dá acesso a uma lista ordenada de forma crescente, produzida pela intercalação de duas listas ordenadas de forma crescente que têm início em dois pNo fornecidos. Faça um programa em C para testar sua função.
191. Rescreva recursivamente em C a função acima. Faça um programa em C para testar sua função.

## Exercícios de Arquivo de struct

Reescreva todas as funções solicitadas nos exercícios da seção Exercícios de struct acima, considerando que os alunos estão em um arquivo de Aluno em vez de estarem em um vetor de Aluno. Assuma ainda que a quantidade de alunos não será fornecida a nenhum dos subprogramas solicitados. Faça programas em C para testar as funções refeitas.

## Exercícios de Arquivo Texto

192. Escreva uma função em C que, dado um arquivo texto, resulte quantas palavras (seqüências de caracteres separadas por espaços em branco e/ou caracteres de pontuação) ele contém. Faça um programa em C para testar sua função.
193. Escreva um procedimento em C que, dados dois arquivos texto abertos, respectivamente para leitura e para escrita, copia o primeiro no segundo. Faça um programa em C para testar seu procedimento.
194. Escreva um procedimento em C que, dados dois arquivos texto abertos, respectivamente para leitura e para escrita, copia o primeiro no segundo, substituindo seqüências de linhas idênticas por apenas uma linha. Faça um programa em C para testar seu procedimento.
195. Escreva um procedimento em C que, dados dois arquivos texto abertos, respectivamente para leitura e para escrita, copia as n últimas linhas do primeiro no segundo. Faça um programa em C para testar seu procedimento.
196. Escreva uma função em C que, dados dois arquivos texto abertos para leitura, compara seus conteúdos e resulte o número da primeira linha na qual eles diferiram. Faça um programa em C para testar sua função.
197. Entenda que um padrão é um *string* que, eventualmente contém o caractere '?' que combina com qualquer caractere e o caractere '\*' que combina com qualquer seqüência de zero ou mais caracteres. Assim sendo, o padrão "ab?c\*de", combina com qualquer *string* que comece com o caractere 'a', seguido pelo caractere 'b' e por um caractere qualquer, seguido então por uma sequência de 0 ou mais caracteres que não termine com os caracteres 'd' e 'e', para então terminar com o caractere 'd', seguido pelo caractere 'e'.
- Escreva uma função em C que procura um padrão dado em um arquivo texto fornecido aberto para leitura, resultando 1 caso encontre, ou 0 caso contrário. Faça um programa em C para testar sua função.
198. Escreva um procedimento em C que calcula e mostra na tela a incidência percentual de cada diferente caractere encontrado em um dado arquivo texto aberto para leitura. Faça um programa em C para testar seu procedimento.
199. Escreva um procedimento em C que processa as linhas de um dado arquivo texto aberto para leitura, escrevendo seu conteúdo em um dado arquivo texto aberto para escrita em linhas que não ultrapassem 80 caracteres e sem mutilar palavras (seqüências de caracteres separadas por espaços em branco e/ou caracteres de pontuação). Faça um programa em C para testar seu procedimento.



200. Considerando que um programa escrito em C é um arquivo texto, escreva um procedimento copia o conteúdo de um dado arquivo texto aberto para leitura, que contem um programa em C, para um dado arquivo texto aberto para escrita, sem incluir neste último os comentários de linha (introduzido por //) e de trecho (delimitados por /\* e \*/). Faça um programa em C para testar seu procedimento.
201. Escreva um procedimento copia o conteúdo de um dado arquivo texto aberto para leitura para um dado arquivo texto aberto para escrita, sem incluir neste último os espaços em branco supérfluos (presentes no início e no final das linhas, bem como aqueles que excedem a quantidade de 1, entre as palavras). Faça um programa em C para testar seu procedimento.