



Exercícios: Strings

1. Faça um programa que então leia uma string e a imprima.
2. Crie um programa que calcula o comprimento de uma string (não use a função `strlen`).
3. Crie um programa que compara duas strings (não use a função `strcmp`).
4. faça um programa que leia um nome e imprima as 4 primeiras letras do nome.
5. Digite um nome, calcule e retorne quantas letras tem esse nome.
6. Ler nome, sexo e idade. Se sexo for feminino e idade menor que 25, imprime o nome da pessoa e a palavra "ACEITA", caso contrário imprimir "NÃO ACEITA".
7. Faça um programa que conte o número de 1's que aparecem em um string. Exemplo: 0011001 -> 3
8. Escreva um programa que substitui as ocorrências de um caractere 0 em uma string por outro caractere 1.
9. Entre com um nome e imprima o nome somente se a primeira letra do nome for "a" (maiúscula ou minúscula).
10. Faça um programa que receba uma palavra e a imprima de trás-para-frente.
11. Faça um programa que receba do usuário uma string. O programa imprime a string sem suas vogais.
12. Faça um programa que receba uma palavra e calcule quantas vogais (a, e, i, o, u) possui essa palavra. Entre com um caractere (vogal ou consoante) e substitua todas as vogais da palavra dada por esse caractere.
13. Ler uma frase e contar quantos caracteres são brancos. Lembre-se que uma frase é um conjunto de caracteres (vetor).
14. Faça um programa que leia uma palavra (máximo de 50 letras) e some 1 no valor ASCII de cada caractere da palavra. Imprima a string resultante.
15. Leia uma cadeia de caracteres e converta todos os caracteres para maiúscula. **Dica:** subtraia 32 dos caracteres cujo código ASCII está entre 65 e 90.
16. Escreva um programa para converter uma cadeia de caracteres de letras maiúsculas em letras minúsculas
17. Leia um vetor contendo letras de uma frase inclusive os espaços em branco. Retirar os espaços em branco do vetor e depois escrever o vetor resultante.
18. Faça um programa em que troque todas as ocorrências de uma letra L1 pela letra L2 em uma string. A string e as letras L1 e L2 devem ser fornecidas pelo usuário.

19. Escreva um programa que leia a idade e o primeiro nome de várias pessoas. Seu programa deve terminar quando uma idade negativa for digitada. Ao terminar, seu programa deve escrever o nome e a idade das pessoas mais jovens e mais velhas.
20. Faça um programa que preencha um vetor com os modelos de cinco carros (exemplos de modelos: Fusca, Gol, Vectra, etc.). Preencha outro vetor com o consumo desses carros, isto é, quantos quilômetros cada um deles faz com um litro de combustível. Calcule e mostre:
- (a) O modelo de carro mais econômico;
 - (b) Quantos litros de combustível cada um dos carros cadastrados consome para percorrer uma distância de 1.000 quilômetros.
21. Faça um programa que receba duas frases distintas e imprima de maneira invertida, trocando as letras A por *.
22. Ler o nome e o valor de uma determinada mercadoria de uma loja. Sabendo que o desconto para pagamento à vista é de 10% sobre o valor total, calcular o valor a ser pago à vista. Escrever o nome da mercadoria, o valor total, o valor do desconto e o valor a ser pago à vista.
23. Escreva um programa que recebe uma string S e inteiros não-negativos I e J e imprima o segmento S[I..J].
24. Escreva um programa que recebe do usuário uma string S, um caractere C, e uma posição I e devolve o índice da primeira posição da string onde foi encontrado o caractere C. A procura deve começar a partir da posição I.
25. Escreva um programa que leia duas palavras e diga qual delas vem primeiro na ordem alfabética. **Dica:** 'a' é menor do que 'b'.
26. O código de César é uma das mais simples e conhecidas técnicas de criptografia. É um tipo de substituição na qual cada letra do texto é substituída por outra, que se apresenta no alfabeto abaixo dela um número fixo de vezes. Por exemplo, com uma troca de três posições, 'A' seria substituído por 'D', 'B' se tornaria 'E', e assim por diante. Implemente um programa que faça uso desse Código de César (3 posições), entre com uma string e retorne a string codificada. **Exemplo:**
- String: a ligeira raposa marrom saltou sobre o cachorro cansado
Nova string: D OLJHLUD UDSRVD PDUURP VDOWRX VREUH R FDFKRUUR FDQVDGR
27. Faça um programa que, dada uma string, diga se ela é um palíndromo ou não. Lembrando que um palíndromo é uma palavra que tenha a propriedade de poder ser lida tanto da direita para a esquerda como da esquerda para a direita. **Exemplo:**
- ovo
arara
Socorram-me, subi no ônibus em Marrocos.
Anotaram a data da maratona
28. Construa um programa que leia duas strings fornecidas pelo usuário e verifique se a segunda string lida esta contida no final da primeira, retornando o resultado da verificação.
29. Implemente um programa que leia duas strings, **str1** e **str2**, e um valor inteiro positivo N. Concatene não mais que N caracteres da string **str2** à string **str1** e termine **str1** com NULL.

30. Leia duas cadeias de caracteres A e B. Determine quantas vezes a cadeia A ocorre na cadeia B.
31. Faça um programa que contenha um menu com as seguintes opções:
- (a) Ler uma string S1 (tamanho máximo 20 caracteres);
 - (b) Imprimir o tamanho da string S1;
 - (c) Comparar a string S1 com uma nova string S2 fornecida pelo usuário e imprimir o resultado da comparação;
 - (d) Concatenar a string S1 com uma nova string S2 e imprimir na tela o resultado da concatenação;
 - (e) Imprimir a string S1 de forma reversa;
 - (f) Contar quantas vezes um dado caractere aparece na string S1. Esse caractere desse ser informado pelo usuário;
 - (g) Substituir a primeira ocorrência do caractere C1 da string S1 pelo caractere C2. Os caracteres C1 e C2 serão lidos pelo usuário;
 - (h) Verificar se uma string S2 é substring de S1. A string S2 deve ser informada pelo usuário;
 - (i) Retornar uma substring da string S1. Para isso o usuário deve informar a partir de qual posição deve ser criada a substring e qual é o tamanho da substring.
32. Leia uma cadeia de caracteres no formato “DD/MM/AAAA” e copie o dia, mês e ano para 3 variáveis inteiras. Antes disso, verifique se as barras estão no lugar certo, e se DD, MM e AAAA são numéricos.
33. Faça um programa para ler uma tabela contendo os nomes dos alunos de uma turma de 5 alunos. O programa deve solicitar ao usuário os nomes do aluno, sempre perguntando se ele deseja inserir mais um nome na lista. Uma vez lidos todos os alunos, o usuário irá indicar um nome que ele deseja verificar se está presente na lista, onde o programa deve procurar pelo nome (ou parte deste nome) e se encontrar deve exibir na tela o nome completo e o índice do vetor onde está guardado este nome.
34. Faça um programa que encontre o conjunto de 5 dígitos consecutivos na sequência abaixo que gere o maior produto:

73167176531330624919225119674426574742355349194934
 96983520312774506326239578318016984801869478851843
 85861560789112949495459501737958331952853208805511
 12540698747158523863050715693290963295227443043557
 66896648950445244523161731856403098711121722383113
 62229893423380308135336276614282806444486645238749
 30358907296290491560440772390713810515859307960866
 70172427121883998797908792274921901699720888093776
 65727333001053367881220235421809751254540594752243
 52584907711670556013604839586446706324415722155397
 53697817977846174064955149290862569321978468622482
 83972241375657056057490261407972968652414535100474
 82166370484403199890008895243450658541227588666881
 16427171479924442928230863465674813919123162824586
 17866458359124566529476545682848912883142607690042

24219022671055626321111109370544217506941658960408
07198403850962455444362981230987879927244284909188
84580156166097919133875499200524063689912560717606
05886116467109405077541002256983155200055935729725
71636269561882670428252483600823257530420752963450