

Lista de Exercícios de Estruturas de Dados e Algoritmos

Q1) Implemente uma função em C que receba uma string como parâmetro e substitua todas as suas letras por predecessoras do alfabeto, trocando maiúsculas por minúsculas. Por exemplo, a string "Amor" seria alterada para "zLNQ". Esta função deve obedecer o seguinte protótipo: **void shift_troca_string (char *str)**. A letra 'a' deve ser substituída por 'Z' (e 'A' por 'z'). Caracteres que não forem letras devem ser substituídos por '!'. Sabe-se que A = 65, Z = 90, a = 97 e z = 122.

Q2) Implemente uma função em C que receba uma string como parâmetro e retorne uma nova string com todos os caracteres minúsculos substituídos pelo caractere '?'. Por exemplo, se for passada a string "740-Charitas-Leme", a função deve retornar a string "740-Ch?r?t?s-L?m?". A assinatura desta função deve ser **char * codifica (char *str)**. A string passada como parâmetro não pode ser alterada. O espaço de memória para a nova string deve ser alocado dinamicamente.

Q3) Considere que você receba um vetor de inteiros com números ímpares e pares e um número **n**, sendo **n** o tamanho do vetor. Implemente uma função que separe os valores inteiros de um vetor em ímpares e pares, obedecendo a ordem de entrada destes números no vetor. Implemente os dois protótipos abaixo:

- a) **int* transf_i_p (int *vet, int n)**
- b) **void transf_i_p (int *vet, int n)**

Q4) Escreva uma função que receba, como entrada, uma cadeia de caracteres **s** e um inteiro **n**, e, em seguida, retire o "**prefixo**" da cadeia **s** de tamanho **n** (isto é, retire os **n** primeiros caracteres). Se a cadeia não tiver pelo menos **n** caracteres, deve ser impressa a mensagem "**erro**". Por exemplo, se **s** = "abcdefghi" e **n** = 3, então a cadeia "defghi" deve ser impressa; com a mesma cadeia **s** e **n** = 17, deve ser impresso "**erro**". O protótipo desta função é o seguinte: **void retira_inicio_n (char *str, int n)**.

Q5) Escreva uma função que receba, como entrada, uma cadeia de caracteres **s** e um inteiro **n**, e, em seguida, retire o "**sufixo**" da cadeia **s** de tamanho **n** (isto é, retire os **n** últimos caracteres). Se a cadeia não tiver pelo menos **n** caracteres, deve ser impressa a mensagem "**erro**". Por exemplo, se **s** = "abcdefghi" e **n** = 3, então a cadeia "abcdef" deve ser impressa; com a mesma cadeia **s** e **n** = 17, deve ser impresso "**erro**". O protótipo desta função é o seguinte: **void retira_fim_n (char *str, int n)**.

Q6) Escreva um programa que receba, como parâmetro de entrada, uma string, e verifique se esta string é uma "boa" senha. Entende-se por "boa" senha uma senha com as seguintes características de formação: (i) ela deve possuir, no mínimo, oito caracteres; (ii) ela deve ter, no mínimo, um caractere entre '0' a '9'; (iii) ela deve possuir, no mínimo, uma letra maiúscula; (iv) ela deve ter, no mínimo, uma letra minúscula; e (v) ela deve ter, pelo menos, um caractere do conjunto formado pelos demais caracteres do teclado.

Q7) Escreva um programa que receba duas strings, como parâmetros de entrada, e informe qual é a maior substring existente nas duas strings. Por exemplo, se as strings de entrada são ACCTGAACCTCCCCC e ACCTAGGACCCCCC, então a maior substring existente entre as duas strings será CCCCCC.

Q8) Escreva uma função que dadas duas strings, retorne UM se a primeira contém a segunda, ignorando maiúsculas e minúsculas, e ZERO, caso contrário. O protótipo da função é o seguinte: **int cic (char *str1, char *str2)**.

Q9) Escreva uma função que dadas duas strings, retorne UM se as strings são iguais, ignorando maiúsculas e minúsculas, e ZERO, caso contrário. O protótipo da função é o seguinte: **int igual (char *str1, char *str2)**.