

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 caracter '?'. 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 caracter 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 caracter do conjunto formato 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 ACCTGAAC TCCCCC e ACCTAGGACCCCC, 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)**.

Q10) Dada uma matriz quadrada de dimensão 9, com valores de 1 a 9 em suas posições, escreva um programa que verifique se esta matriz é uma solução válida para o **Sudoku** (isto é, uma solução é válida no **Sudoku** se cada linha, cada coluna e cada bloco contém os números de 1 a 9 somente uma vez).

```

5 3 4 | 6 7 8 | 9 1 2
6 7 2 | 1 9 5 | 3 4 8
1 9 8 | 3 4 2 | 5 6 7
-----+-----+-----
8 5 9 | 7 6 1 | 4 2 3
4 2 6 | 8 5 3 | 7 9 1
7 1 3 | 9 2 4 | 8 5 6
-----+-----+-----
9 6 1 | 5 3 7 | 2 8 4
2 8 7 | 4 1 9 | 6 3 5
3 4 5 | 2 8 6 | 1 7 9

```

Q11) Considere duas matrizes de inteiros de dimensões **m1xn1** e **m2xn2**, onde m1 e n1 representam, respectivamente, o número de linhas e o número de colunas da primeira matriz, e m2 e n2 representam, respectivamente, o número de linhas e o número de colunas da segunda matriz. Escreva uma função que realize a multiplicação destas duas matrizes sem alterar nem a primeira e nem a segunda matriz: **int** mult (int m1, int n1, int **mat1, int m2, int n2, int **mat2)**

Q12) Escreva uma função que dada uma string composta por substrings separadas por "/", retorne todas as suas substrings da string original. Por exemplo, a string **/Fla/Flu/Bota/** será decomposta em:

Fla

Flu

Bota

Esta função não deve modificar a string original. O protótipo da função é o seguinte: **char** s2ss (char *str)**. Isto é, a resposta desta função será uma matriz de caracteres alocada dinamicamente.

Q13) A matriz de Hadamard $H(N)$, usada em projetos de programas corretores de erros, é uma matriz N por N , onde N é potência de dois, de elementos booleanos (isto é, elementos 0 e 1) que satisfaz a seguinte propriedade: dadas duas linhas distintas i e j , onde $0 \leq i < N$ e $0 \leq j < N$, desta matriz, a quantidade de elementos distintos nestas linhas é sempre igual a $N/2$. Abaixo exemplifica-se $H(1)$, $H(2)$ e $H(4)$.

```

H(1) | H(2) | H(4)
-----+-----+-----
1 | 1 1 | 1 1 1 1
  | 1 0 | 1 0 1 0
  |    | 1 1 0 0
  |    | 1 0 0 1

```

Para construir $H(M)$, onde $M = 2 * N$, divide-se a matriz $H(M)$ em quatro partes iguais, chamadas de quadrantes, repete-se três vezes a matriz $H(N)$ nos quadrantes de menores índices, e no quadrante de maiores índices de $H(M)$, inverte-se a matriz $H(N)$. Implemente um programa que imprima a matriz $H(N)$ na console. A dimensão desta matriz deve ser lida pelo seu programa.