

Lista de exercícios de revisão

1 - Defina uma estrutura (*struct*) para representar uma data, armazenando dia, mês e ano. Crie uma função que recebe uma instância desse tipo de dados e imprime a data no formato dd/mm/aaaa.

2 - Crie uma função que recebe duas datas A e B, representadas pela estrutura definida no exercício anterior e faz a comparação entre ambas. Sua função deve devolver -1 caso A seja anterior a B, 0 caso ambas data sejam iguais e 1 caso A seja posterior a B.

3 - Crie uma função que recebe um vetor de `int` de comprimento n e inverte o conteúdo do vetor, sem alocar um vetor auxiliar.

4 - Um *palíndromo* é uma sequência de caracteres que é igual ao seu próprio inverso (p.ex. “arara”). Crie uma função que recebe um vetor de `char` de comprimento n e devolve 1 caso a sequência seja um palíndromo e 0 caso contrário.

5 - Crie uma função que recebe um vetor de `char` de comprimento n e imprime o número de ocorrências de cada letra presente no *string*. Sua função deve ser insensível a caixa baixa/alta. Por exemplo, para a entrada “Ovo”, sua função deve imprimir:

o: 2

v: 1

6 - Crie uma função que recebe um vetor de números inteiros de comprimento n e calcula o tamanho da maior sequências de zeros do vetor.

7 - Considere dois números k e n inteiros positivos, o algoritmo mais imediato para calcular k^n envolve $(n - 1)$ multiplicações. É possível criar um algoritmo mais eficiente (que demanda um número menor de multiplicações) se notarmos que $k^n = k^{n/2} \cdot k^{n/2}$ para n par. Crie uma função recursiva para calcular k^n de maneira eficiente. Suponha que k^n pode ser representado por um valor do tipo `int`.

8 - Use a definição abaixo para uma célula de uma lista ligada:

```
typedef struct reg {  
    int conteudo;  
    struct reg *prox;  
} celula;
```

Crie uma função que recebe uma lista ligada e um valor inteiro `a` e remove da lista todas as células que contém o valor `a`.