

Exercícios de revisão para a prova:

Exe_01. Faça um algoritmo que leia dois nomes completos, contendo apenas um nome e um sobrenome. O programa deve exibir os nomes completos da seguinte forma:

- a) Ordenados alfabeticamente pelo nome.
- b) Ordenados alfabeticamente pelo sobrenome.

Exe_02. Leia as sentenças a seguir e marque V se a mesma for verdadeira ou F se ela for falsa.

V F Funções recursivas precisam de três coisas: retornar algo, possuir parâmetros e serem chamadas por outras funções.

V F Em um algoritmo que utilize o processo de recursão, é necessária a existência de uma condição de parada a fim de se evitar que o algoritmo entre em um loop.

V F Uma função recursiva geralmente usa algum tipo de estrutura de repetição.

V F Uma função recursiva precisa de casos de base, isto é, de argumentos para os quais já sabemos os valores da função.

V F Uma função é dita recursiva quando ela "chama" a si mesma.

Exe_03. Qual é a saída dos algoritmos abaixo?

- a) Primeiro

```
1  #include <stdio.h>
2
3  void ex2(int a, int b) {
4      if(a <= 1)
5          return;
6      printf("%d / %d = %d\n", a, b, a / b);
7      ex2(a/2, b);
8  }
9
10 int main() {
11     ex2(24, 2);
12     return 0;
13 }
```

b) Segundo

```
1 #include <stdio.h>
2
3 int ex3(int n) {
4     if(n == 5)
5         return n;
6     else
7         return 3 + ex3(n+1);
8 }
9
10 int main() {
11     printf("%d", ex3(0));
12     return 0;
13 }
```

Exe_03. Crie um algoritmo que converta:

- a) Uma pilha para uma fila.
- b) Uma fila para uma pilha.

Exe_04. Defina: lista estática homogênea, lista estática heterogenia, lista dinâmica homogênea, lista dinâmica heterogenia, lista simplesmente encadeada e lista duplamente encadeada.

Exe_05. Faça um algoritmo que aplique o conceito de uma lista duplamente encadeada na inserção e impressão de dados na lista.

- a) Possibilite excluir itens da lista.
- b) Permita listar os X primeiros itens da lista ou os X últimos itens da lista.