

Tema: Recuperação 02

Atividade: Procedimentos e funções sobre grupos de dados

#### INSTRUÇÕES:

- Esses exercícios são extras, opcionais e recomendados, principalmente, para os alunos que obtiveram aproveitamento abaixo de 60% da segunda avaliação teórica. O objetivo é rever aplicações de conceitos essenciais sobre grupos de dados.
- Desenvolver um método para cada um dos enunciados abaixo.
- Cada método deverá conter, em seu cabeçalho, como comentário (`/** e */`), a documentação essencial: nome e matrícula, identificação, objetivo, parâmetros e condições especiais, se houver, e relatório de testes (exemplos de valores usados e condições testadas).

SUGESTÃO: Montar um menu para a escolha do método a ser testado (ver modelo em Lista00.c).

Testes deverão ser realizados e os valores usados deverão ser guardados no final do programa como comentários (`/* e */`).

O uso de recursão é recomendado, mas opcional; se desejar utilizá-lo, fazer também a implementação da forma não-recursiva.

01.) FAZER um programa com um método ( exercicio01 ) para:

- definir e usar uma função inteira para calcular quantos são os divisores de um número inteiro;
- ler valores inteiros do arquivo DADOS.TXT, o último será zero, e não deverá ser processado;
- calcular e gravar em RESULTADOS.TXT, dentre os números lidos, os que tiverem a maior e a menor quantidade de divisores.

02.) FAZER um programa com um método ( exercicio02 ) para:

- definir e usar uma função inteira para calcular a soma dos divisores de um número inteiro;
- definir uma função lógica para saber se um valor inteiro é perfeito: se a soma de seus outros divisores é igual a ele mesmo;
- calcular e GRAVAR em PERFEITOS.TXT, no intervalo entre 1000 e 9999, quantos e quais são os números perfeitos.

03.) FAZER um programa com um método ( exercicio03 ) para:

- definir e usar uma função inteira para calcular o máximo divisor comum entre dois inteiros;
- ler dois valores inteiros ( M, N ) do arquivo NUMEROS.TXT;
- calcular e mostrar o máximo divisor comum ( M.D.C. ) entre eles.

04.) FAZER um programa com um método ( exercicio04 ) para:

- definir e usar uma função inteira para calcular o mínimo múltiplo comum entre dois inteiros;
- ler dois valores inteiros ( M, N ) do arquivo NUMEROS.TXT;
- calcular e mostrar o mínimo múltiplo comum ( M.M.C. ) entre eles.

05.) FAZER um programa com um método ( exercicio05 ) para:

- ler valores inteiros do teclado, mas só serão válidos os valores iguais a zero ou a um; se algum dígito for fornecido errado, deverá ser lido novamente;
  - armazená-los em um arranjo de inteiros (int);
  - supondo serem dígitos de um número binário, convertê-los para decimal mediante o uso de uma função.
- DICA: Para ler novamente em caso de erro, usar do-while.  
Exemplo:

```
// Arranjo binario = { 1, 0, 1, 1 }  
int x = paraDecimal ( binario );    // x = 13
```

06.) FAZER um programa com um método ( exercicio06 ) para:

- ler um número inteiro ( N ) para indicar o tamanho de um arranjo;
- ler ( N ) valores inteiros do arquivo CRESCENTE.TXT e inserir cada um no arranjo;
- definir uma função para receber o arranjo como parâmetro e inverter a ordem de seus elementos;
- gravar o arranjo invertido no arquivo INVERTIDOS.TXT, colocando a quantidade de dados elementos na primeira linha.

DICA: Trocar o último com o primeiro e prosseguir até a metade.

Exemplo:

```
// int arranjo1 [ ] = { 1, 2, 3, 4 };  
int arranjo2 [4];  
inverter ( arranjo2, arranjo1 ); //arranjo2 = [ 4, 3, 2, 1 ]
```

07.) FAZER um programa com um método ( exercicio07 ) para:

- ler do arquivo DADOS1.TXT um número inteiro ( N ) para indicar o tamanho de um arranjo;
- ler ( N ) valores inteiros do arquivo DADOS1.TXT e inserir cada dado em um arranjo;
- ler do arquivo DADOS2.TXT um número inteiro ( N ) para indicar o tamanho de outro arranjo;
- ler ( N ) valores inteiros do arquivo DADOS2.TXT e inserir cada dado no outro arranjo;
- filtrar e mostrar os elementos comuns aos dois arranjos, sem repetições;
- gravar o resultado no arquivo FILTRADOS.TXT, colocando a quantidade de dados únicos na primeira linha.

08.) FAZER um programa com um método ( exercicio08 ) para:

- ler do arquivo MATRIZ1.TXT  
a quantidade de linhas, colunas e elementos  
de uma matriz de inteiros;
  - definir uma função lógica para verificar  
se a matriz lida apresenta a característica abaixo.
- OBS.: Não usar *break* ou *continue*.

```
1  1  1  1
1  2  4  8
1  3  9 27
1  4 16 64
```

09.) FAZER um programa com um método ( exercicio09 ) para:

- ler do arquivo MATRIZ1.TXT  
a quantidade de linhas, colunas e elementos  
de uma matriz de inteiros;
  - ler do arquivo MATRIZ2.TXT  
a quantidade de linhas, colunas e elementos  
de outra matriz de inteiros;
  - definir uma função (compareTo) para comparar  
se ambas são iguais, parando tão logo sejam diferentes  
e retornar 0, se forem iguais,  
(+1) se o elemento diferente na primeira for maior, e  
(- 1) se o elemento diferente na primeira for menor.
- OBS.: Não usar *break* ou *continue*.

10.) FAZER um programa para:

- ler um número inteiro ( N ) para indicar  
o número de supermercados cujos produtos serão avaliados;
- ler o nome e o código de cada supermercado;
- ler o preço de um produto de cada supermercado;
- calcular o preço médio desse produto;
- informar pelo menos dois supermercados com preços  
inferiores à média.

