

INFORMAÇÕES SOBRE ESTA ATIVIDADE:

- Esta atividade valerá 2,0 pontos (1,0 ponto de trabalho mais 1,0 ponto de AAI).
- A atividade deve se iniciar no laboratório e ser entregue pelo SINEF **até dia 25/nov/2012**.
- Deve ser entregue um arquivo compactado chamado **Ativ10.zip** com um arquivo chamado **grupo.txt** contendo o nome dos integrantes e os arquivos **c** cujos nomes devem seguir o seguinte padrão: **Ativ10Ex01.c, Ativ10Ex02.c, ... Isto será avaliado!**
- Apenas **um** do grupo deverá entregar. **Isto também será avaliado!**
- Atividade não serão recebidas/pontuadas **após o prazo de entrega**. Portanto, fique atento ao prazo.
- Cópias** (total ou parcial) serão penalizadas com **nota zero**.
- A atividade é em grupo de, no máximo, **dois** alunos.

Atividade 10 - Lista de Exercícios – Arquivos

- Crie um programa C ANSI que abra um arquivo texto cujo nome seja informado via argumento de linha de comando (utilizar parâmetros formais **argc** e **argv** da função **main**) e conte o número de caracteres presentes nele. Imprima o número de caracteres na tela.
- Crie um programa C ANSI que receba o nome via argumento de linha de comando de um arquivo e gere uma cópia do mesmo.
- Crie um programa C ANSI que leia dez caracteres e armazene em um arquivo dez cópias de cada caractere. Depois exiba o conteúdo do arquivo.
- Crie um programa C ANSI que compare dois arquivos especificados pelo usuário (via argumento de linha de comando) e imprima sempre que os caracteres dos dois arquivos coincidirem. Por exemplo:

arquivo1.txt

Olá, pessoal!

arquivo2.txt

Oi, como vai?

Neste caso, os caracteres da primeira e da décima primeira posição são iguais nos dois arquivos. A saída do programa deve ser:

POSICAO 01 - O (79)

POSICAO 11 - a (97)

Obs: Os valores entre parênteses são os respectivos códigos ASCII dos caracteres.

- Altere o programa anterior para salvar a saída em "resultadoMerge.txt".
- Escreva um programa que leia para um arranjo, uma matriz unidimensional de *double* de um arquivo texto informado via argumento de linha de comando e depois imprima a média de todos os valores. A primeira linha indica o tamanho do arranjo e na segunda linha estão os valores separados por tabulação ('\t'). Use alocação dinâmica.

Exemplo do formato do arquivo:

6
1.1 2.2 3.3 4.4 5.5 6.6

- Escreva um programa similar ao anterior que leia para um arranjo, uma matriz bidimensional de inteiros de um arquivo texto, e depois imprima o soma de todos os valores. A primeira linha indica o número de linhas e colunas e em cada linha seguinte as colunas estão separadas por espaço (' '). Use alocação dinâmica.

Exemplo do formato do arquivo:

6 3
10 11 12 13 14 15
16 17 18 19 20 21
22 23 24 25 26 27

8. Crie um programa C ANSI que armazene as informações de uma data na seguinte estrutura:

```
typedef struct {
    short int dia;
    short int mes;
    int ano;
} data;
```

9. Crie um programa C ANSI como o anterior, porém que realize a leitura da data e posterior impressão em tela.
10. Crie um programa C ANSI que armazene as informações de cinco carros. Deve ser utilizada a seguinte estrutura:

```
typedef struct {
    char nome[60];
    char modelo[60];
    int anoFabricacao;
    double valor;
} carro;
```

11. Crie um programa C ANSI como o anterior, porém que realize a leitura das informações dos carros e posterior impressão em tela.
12. [LÓGICA] Crie um programa C ANSI que receba – via argumento de linha de comando – o nome de um arquivo binário com vários números inteiros. O objetivo é apontar quais e quantos números estão cercados por números maiores que ele. Considere um arranjo circular; por exemplo assumamos [7,8,3], para o 1º elemento (7) considere o último (3) e o 2º (8) e para o último elemento (3) considere o penúltimo (8) e o 1º (7).

Por exemplo:

[9, 1, 4, 5, 0, 0, 7, -2, 13, 8] → 3 (1, -2, 8)
[3, 5, -1, 4, 1, 18, 0, -5, 2, 8] → 4 (3, -1, 1, -5)