



## Laboratório 04

1º Considerando o arquivo texto apresentado, o que o trecho de código abaixo faz?

A leitura de arquivos pode ser realizada por caractere,  
por palavras ou por linhas.

```
#include <stdio.h>
#include <stdlib.h>
```

```
int main()
{
    FILE *fin;
    fin = fopen("intro.txt", "r");

    char ch;
    fscanf(fin, "%c", &ch);
    fclose(fin);

    printf("%c\n", ch);
    return 0;
}
```

Modifique o programa para ler do arquivo texto:

- a) O oitavo caractere
- b) Uma palavra
- c) A quinta palavra
- d) Uma linha

2º Considere o arquivo texto apresentado abaixo:

```
Ontem, às 12 horas, faziam 40 graus ao sol e 38 à sombra.  
Eu andei 100 metros antes de alcançar um lugar à sombra.  
Foram 5 minutos de sofrimento.
```

Escreva um programa para ler todos os números do texto e gravá-los em outro arquivo.

```
12  
40  
38  
100  
5
```

3º Construa um programa que leia um número desconhecido de peixes de um arquivo texto, chamado “pescado.txt”, e exiba a quantidade total de quilos de peixe pescado. O arquivo é formatado como no exemplo abaixo: o nome do peixe, o peso em gramas e o comprimento em centímetros.

```
Dourado 250 20  
Tilapia 300 25  
Carpa 400 40  
Sardinha 50 10  
Atum 150 15
```

4º Construa um programa que leia um arquivo texto contendo o nome e as três notas de vários alunos (uma quantidade indefinida de alunos). Escreva em outro arquivo texto o nome e a situação do aluno (aprovado, quarta prova ou reprovado).

```
Wilson 8.2 7.5 5.0  
Silvio 3.2 7.5 7.2  
Luis 6.0 7.0 9.0  
Pedro 3.0 3.0 1.0
```

5º Escreva um programa que abra um arquivo texto, leia caractere a caractere até o fim do arquivo e exiba na tela o número total de caracteres, o número de vogais, o número de consoantes, e a quantidade de outros caracteres presentes no texto.

6º Escreva um programa que peça ao usuário para digitar um nome de arquivo texto e uma palavra que ele deseja procurar neste arquivo. O programa deve dizer se a palavra está ou não presente no arquivo.

```
Nome do arquivo: texto.txt
Palavra: homens
A palavra "homens" está presente no texto.
```

7º Uma palavra é um palíndromo se a sequência de letras que a forma é a mesma quando lida da esquerda para a direita ou da direita para a esquerda (ex: raia). Escreva uma função que seja capaz de descobrir se uma palavra é um palíndromo. Nesta verificação desconsidere maiúsculas e minúsculas (i.e. Ana é um palíndromo).

O programa deve ler um número indefinido de palavras de um arquivo texto e, para cada palavra, verificar se ela é ou não um palíndromo. Exiba a quantidade de palíndromos encontrados no texto, bem como as palavras que são palíndromos, como mostrado abaixo:

```
No raia do dia Ana observou que esse radar estava quebrado
```

```
Arquivo: palindromos.txt
raia
ana
esse
radar
Foram encontrados 4 palíndromos neste arquivo.
```

8º Uma escola deseja fazer uma competição intercalasses com seus alunos. A secretaria da escola montou um arquivo texto com a lista dos alunos interessados em participar da competição. A listagem contém o nome do aluno e um código que indica o turno (manhã ou tarde) e a série (6ª, 7ª ou 8ª), como no exemplo abaixo:

Anita Torcato	M8	Marcela Arantes	M6
Bárbara Borja	T6	Vânia Horta	T8
Estevan Prado	T7	Óscar Neiva	M7
Eudes Salvado	M6	Amélia Reino	M8
Geovana Castro	M7	Cláudio Reine	T6
Pedro Vieira	T7	Fábria Candeias	T8

Crie um registro para representar um aluno, leia as informações do arquivo e guarde em um vetor de alunos. Em seguida use laços e testes condicionais para

separar e exibir os alunos agrupados por turno e série, como no exemplo abaixo.

Matutino 6a Série		
-----		
Eudes Salvado		M6
Marcela Arantes		M6
Matutino 7a Série		
-----		
Geovana Castro		M7
Óscar Neiva		M7
Matutino 8a Série		
-----		
Anita Torcato		M8
Amélia Reino		M8
Vespertino 6a Série		
-----		
Bárbara Borja		T6
Cláudio Reine		T6
Vespertino 7a Série		
-----		
Estevam Prado		T7
Pedro Vieira		T7
Vespertino 8a Série		
-----		
Vânia Horta		T8
Fábia Candeias		T8

**Sugestão:** tente também listar os alunos separados apenas por turno.

a) Seria possível separar os alunos por sexo usando os dados fornecidos no programa? Descreva sua solução.

b) Se fosse possível adicionar novas informações aos dados, qual seria a forma mais fácil de conseguir fazer essa separação por sexo?

9º Use o registro peixe definido abaixo.

```
struct peixe
{
    char nome[20];
    unsigned peso;
    float comp;
};
```

a) Construa um programa que leia do usuário os dados de um único peixe e salve-os em um arquivo binário. Cada execução do programa deve acrescentar um peixe ao arquivo.

b) Modifique o programa obtido no item anterior para ler e exibir os peixes cadastrados no arquivo binário antes de permitir que o usuário faça um novo cadastro.

**10º** Construa um programa que leia um número desconhecido de peixes de um arquivo texto (semelhante ao arquivo utilizado na primeira questão) e grave estas informações em um arquivo binário que possa ser lido pelo programa da segunda questão. Compare o tamanho dos arquivos texto e binário.

**11º** Escreva um programa que leia um valor inteiro do usuário e armazene-o em um arquivo binário chamado “integer.bin”. Ao entrar, o programa deve verificar se o arquivo “integer.bin” existe, e em caso positivo, deve ler e exibir o número na tela.

Exemplo 1:

```
Digite um inteiro: 43
Número armazenado no arquivo integer.bin.
```

Exemplo 2:

```
Arquivo integer.bin encontrado!
Ele contém o número 43.

Digite um inteiro: 52
Número armazenado no arquivo integer.bin.
```

**12º** Construa um programa que leia do usuário uma sequência de números inteiros, até que o número zero seja lido. Os números devem ser armazenados

em um arquivo binário chamado “vet.dat”. O primeiro número do arquivo deve ser a quantidade de números digitados. Na entrada do programa verifique se o arquivo existe e, caso positivo, exiba os valores do vetor.

Exemplo 1:

```
Digite números (zero para encerrar):  
50 82 45 31 94 22 0  
6 números foram armazenados em vet.dat.
```

Exemplo 2:

```
O arquivo vet.dat contém:  
50 82 45 31 94 22  
  
Digite números (zero para encerrar):  
83 40 29 0  
3 números foram armazenados em vet.dat.
```

13º O Exercício de Revisão 1 do Laboratório 6 (Tipos Inteiros) mostra como usar códigos de Escape para mudar a cor do texto exibido em um terminal. A sequência de caracteres “\033[38;5;000;48;5;154m” define a cor 000 para o texto e 154 para o fundo. Considerando que as cores são valores de 3 dígitos de 000 a 255, construa um programa para ler do usuário um texto com no máximo 80 caracteres, um código para a cor do texto e um código para a cor do fundo. Guarde essas informações em um arquivo binário. O usuário deve ter a opção de guardar uma nova frase ou exibir a frase já armazenada através de um menu, como mostrado no exemplo abaixo.

```
Mensagens Coloridas  
-----  
[A]rmazenar  
[E]xibir  
[S]air  
-----  
Opção: [A]  
  
Sua mensagem: Feliz Aniversário!  
Cor do texto: 130  
Cor do fundo: 038  
  
Texto colorido foi armazenado.
```

**14º** No exercício anterior, se o usuário digitar apenas espaços para a mensagem, o resultado será uma faixa colorida com a cor de fundo. Podemos usar isso para criar uma imagem formada apenas por caracteres de espaço coloridos.

Construa um programa que peça a largura e altura da imagem e leia do usuário uma matriz de números. Cada número de 3 dígitos representa a cor de um “bloco” da imagem. Guarde em um arquivo binário os valores de altura, largura e de cada bloco da imagem. Assim como o programa anterior, construa um menu para controlar as opções de armazenamento e exibição da imagem.

```
Imagens Coloridas
-----
[A]rmazenar
[E]xibir
[S]air
-----
Opção: [A]

Largura: 5
Altura : 5

Defina 5 cores em cada linha
#1: 220 220 220 220 220
#2: 220 000 220 000 220
#3: 220 220 220 220 220
#4: 220 000 000 000 220
#5: 220 220 220 220 220

Imagem colorida foi armazenada.
```

**15º** Crie um programa que grave os 100 primeiros números naturais ao mesmo tempo em um arquivo texto e em um arquivo binário. Compare os tamanhos dos arquivos e explique como os tipos escolhidos para guardar os números podem tornar o arquivo binário menor ou maior que o arquivo texto.

**16º** Crie um registro Soldado com os campos nome, eliminações, mortes, taxa de eliminação por morte (eliminações/mortes) e número de partidas jogadas. Na função principal leia um soldado de um arquivo binário. Se o arquivo não existir, você deve direcionar o usuário para a criação de um novo soldado. Quando o soldado for lido (seja pelo arquivo ou pelo teclado), você deve dar as seguintes opções ao usuário:

[N]ovo soldado (sobrescreve o anterior)  
[A]tualiza soldado (com os dados da última partida)  
[E]xibe soldado (atual)  
[S]air

**Novo:** deve pedir as informações de um soldado, armazená-las em memória durante a execução do programa ao final do programa gravá-las em um arquivo binário.

**Atualiza:** o usuário deve digitar as informações referentes a última partida do jogo (número de eliminações e mortes), e o programa deve atualizar as informações do soldado contabilizando as informações dessa partida.

**Exibe:** crie uma função que receba um Soldado e mostre as informações dele na tela. Chame a função passando o soldado atual sempre que essa opção for selecionada.

**17º** Crie uma união chamada Senha com os campos Alfanumérica e Numérica. Na função principal pergunte ao usuário se ele quer exibir a senha armazenada ou gravar uma nova senha no arquivo. Se o arquivo não tiver sido criado ainda, a opção de exibição não deve fazer nada, a não ser avisar o usuário que o arquivo não existe.

Use um arquivo binário para guardar a senha. Na gravação da senha, grave primeiro um número para representar o tipo da senha. O modo alfanumérico será correspondente ao número 1, e o modo simplesmente numérico será correspondente ao 2. Quando o usuário abrir o programa e selecionar a opção de exibição, o tipo da senha deve ser lido para decidir qual campo da união apresentar na tela (campo alfanumérico ou o campo numérico).