

Lista de Exercícios

Exercícios retirados/adaptados do livro:

BACKES, A. **Linguagem C**: Completa e Descomplicada. Rio de Janeiro (RJ): Elsevier, 2013.

Structs

1. Faça um programa que leia os dados de 10 alunos (Nome, RA, Media Final), armazenando em um vetor. Uma vez lidos os dados, divida esses dados em 2 novos vetores, o vetor dos aprovados e o vetor dos reprovados, considerando a média mínima para a aprovação como sendo 6,0. Exibir na tela os dados do vetor de aprovados, seguido dos dados do vetor de reprovados.
2. Faça um programa que leia um vetor com os dados de 5 carros: marca (máximo 15 letras), ano e preço. Leia um valor **p** e mostre as informações de todos os carros com preço menor que **p**. Repita este processo até que seja lido um valor $p = 0$.
3. Faça um programa que leia um vetor com dados de 5 livros: título (máximo 30 letras), autor (máximo 15 letras) e ano. Depois disso, procure um livro pelo autor digitado pelo usuário. Mostre todos os dados dos livros encontrados.

Funções

4. Faça uma função que receba uma data (três parâmetros inteiros) e retorne-a no formato textual por extenso. Exemplo: Data: 01/01/2000, Retornar: 1 de janeiro de 2000.
5. Escreva uma função que receba um número inteiro maior do que zero e retorne a soma de todos os seus algarismos. Por exemplo, ao número 251 corresponder à o valor 8 ($2 + 5 + 1$). Se o número lido no programa principal não for maior do que zero, o programa terminar a com a mensagem “Número inválido”.
6. Faça uma função que receba um número inteiro positivo **N** e retorne o superfatorial desse número. O superfatorial de um número **N** é definida pelo produto dos **N** primeiros fatoriais de **N**. Assim, o superfatorial de 4 é $sf(4) = 1! \cdot 2! \cdot 3! \cdot 4! = 288$.

7. Considerando a estrutura:

```
typedef struct {  
    int x;  
    int y;  
} Ponto;
```

para representar um ponto em uma grade 2D, implemente uma função que indique se um ponto **p** está localizado dentro ou fora de um retângulo. O retângulo é definido por seus vértices inferior esquerdo **v1** e superior direito **v2**. A função deve retornar **1** caso o ponto esteja localizado dentro do retângulo e **0** caso contrário. Essa função deve obedecer ao protótipo:

```
int dentroRet (Ponto v1, Ponto v2, Ponto p);
```

Ponteiros (Passagem de parâmetros por referência)

8. Faça um programa que leia 2 valores inteiros e chame uma função que receba essas 2 variáveis e troque o seu conteúdo, ou seja, esta função é chamada passando duas variáveis **A** e **B** por exemplo e, após a execução da função, **A** conterá o valor de **B** e **B** terá o valor de **A**.
9. Escreva uma função que aceita como parâmetro um vetor de inteiros com N valores, e determina o maior elemento do vetor e o número de vezes que esse elemento ocorreu no vetor. Por exemplo, para um vetor com os seguintes elementos: 5, 2, 15, 3, 7, 15, 8, 6, 15, a função deve retornar para o programa que a chamou o valor 15 e o número 3 (indicando que o número 15 ocorreu 3 vezes). A função deve ser do tipo **void**.
10. Escreva uma função que receba um vetor de inteiros **V** e os endereços de duas variáveis inteiras, **min** e **max**, e armazene nessas variáveis o valor mínimo e máximo do vetor. A leitura do vetor deve ser realizada na função **main()**.

Arquivos

11. Escreva um programa que:
 - a) crie/abra um arquivo texto de nome "arq.txt";
 - b) permita que o usuário grave diversos caracteres nesse arquivo, até que o usuário entre com o caractere '0';
 - c) feche o arquivo.Agora, abra e leia o arquivo, caractere por caractere, e escreva na tela todos os caracteres armazenados.
12. Faça um programa que receba do usuário o nome de um arquivo texto existente e um caractere. Mostre na tela quantas vezes aquele caractere ocorre dentro do arquivo.

13. Faça um programa que receba do usuário o nome de um arquivo texto existente e mostre na tela quantas vezes cada letra do alfabeto aparece dentro do arquivo.
14. Faça um programa que receba do usuário o nome de um arquivo texto existente. Crie outro arquivo texto contendo o texto do arquivo de entrada, mas com as vogais substituídas pelo caractere ' * '.
15. Faça um programa que receba como entrada o ano corrente e o nome de dois arquivos: um de entrada (existente) e outro de saída. Cada linha do arquivo de entrada contém o nome de uma pessoa e o seu ano de nascimento, separados pelo caracteres ' ; '. O programa deverá ler o arquivo de entrada e gerar um arquivo de saída em que aparece o nome da pessoa seguida por uma string que representa a sua idade.
 - Se a idade for menor do que 18 anos, escreva “menor de idade”
 - Se a idade for maior do que 18 anos, escreva “maior de idade”
 - Se a idade for igual a 18 anos, escreva “entrando na maior idade”
16. Faça um programa que leia o nome de um arquivo existente que contenha as dimensões de uma matriz (linha e coluna), a quantidade de posições que serão anuladas, e as posições a serem anuladas (linha e coluna). O programa lê esse arquivo e, em seguida, produz um novo arquivo com a matriz com as dimensões dadas no arquivo lido, e todas as posições especificadas no arquivo ZERADAS e o restante recebendo o valor 1.

Ex:

Arquivo de entrada: “matriz.txt”

```
3 3 2 /*3 e 3 dimensões da matriz e 2 posições que serão anuladas*/
1 0  /*primeira posição anulada*/
1 2  /*segunda posição anulada*/
```

Arquivo de saída: “matriz saida.txt”

```
1 1 1
0 1 0
1 1 1
```