



MINISTÈRIO DA EDUCAÇÃO  
**UNIVERSIDADE FEDERAL DE ITAJUBÁ**  
Criada pela Lei nº 10.435 – 24/04/2002

## Lista de Exercício

### Leia com atenção as observações abaixo:

- (1) Considere a struct ponto definida abaixo e um vetor de estruturas struct ponto, denominado vet, de tamanho 2

```
struct ponto{  
    int x, y;  
}
```

Na declaração **e somente na declaração**, vet pode ser inicializado das seguintes maneiras:

```
struct ponto vet[2] = { 10,20,30,40 } ou struct ponto vet[2] = { { 10,20 }, { 30,40 } }
```

Nas duas formas, teremos os seguintes valores para os campos x e y dos elementos dos vetores:

```
vet[0].x = 10 e vet[0].y = 20
```

```
vet[1].x = 30 e vet[1].y = 40
```

- (2) Em C, é possível redefinir um tipo existente atribuindo um novo nome ao tipo. Isto é feito usando o comando typedef.

Veja o código abaixo. O tipo **int** foi renomeado para **inteiro** e o tipo **float** para **decimal**. Tendo feito esta renomeação, as variáveis dos tipos int e float podem ser declaradas usando os novos identificadores de tipos criados: inteiro e decimal, respectivamente.

```
typedef int inteiro;  
typedef float decimal;  
int main () {  
    inteiro código;  
    decimal preco;  
    printf("\nDigite o código e o preço do produto: ");  
    scanf("%d %f", &código, &preco);  
    return 0;  
}
```

Ao declarar uma estrutura, não basta usar somente o identificador da estrutura, é necessário usar também a palavra chave struct. Por exemplo, ao declarar o vetor vet, foi usado **struct ponto vet[2]**.

Para simplificar esta declaração, podemos usar o typedef para definir um novo nome (identificador) para estrutura:



MINISTÈRIO DA EDUCAÇÃO  
**UNIVERSIDADE FEDERAL DE ITAJUBÁ**  
Criada pela Lei nº 10.435 – 24/04/2002

```
struct ponto{  
    int x y;  
}  
typedef struct ponto tponto;
```

A partir da redefinição do tipo **struct ponto** para **tponto**, podemos usar este novo tipo para declarar novas estruturas do tipo struct ponto:

Veja o exemplo abaixo:

```
struct ponto{  
    int x, y;  
}  
typedef struct ponto tponto;  
int main(){  
    tponto p;  
    p.x = 10; p.y=20;  
}
```

Os dois blocos de instruções abaixo são idênticos:

**Bloco 1:**

```
struct ponto{  
    int x, y;  
}  
typedef struct ponto tponto;
```

**Bloco 2:**

```
typedef struct ponto{  
    int x, y;  
} tponto;
```

- (3) Uma estrutura pode ser usada para definir um campo de outra estrutura, por exemplo:

```
struct figura {  
    char nome[50];  
    tponto posicao;  
};
```

Veja que a struct ponto foi usada para definir um campo da struct figura.

Agora, usando o typedef:

```
struct ponto {int, x, y;}  
typedef struct ponto tponto;  
struct figura {char nome[50]; tponto posicao;};  
typedef struct figura tfigura;
```



**MINISTÈRIO DA EDUCAÇÃO**  
**UNIVERSIDADE FEDERAL DE ITAJUBÁ**

Criada pela Lei nº 10.435 – 24/04/2002

Na função principal ou nas demais funções implementadas em um programa, os campos da estrutura figura são acessados da seguinte forma:

```
int main(){
    tfigura f;
    strcpy(f.nome, "Quadrado");
    f.posicao.x = 10; //acesso ao campo x do campo posição da figura
    f.posicao.y = 20; //acesso ao campo y do campo posição da figura
}
```



**MINISTÈRIO DA EDUCAÇÃO**  
**UNIVERSIDADE FEDERAL DE ITAJUBÁ**

Criada pela Lei nº 10.435 – 24/04/2002

1. Seja uma estrutura para descrever os carros de uma determinada revendedora, contendo os seguintes campos:
  - marca: string de tamanho 15
  - ano: inteiro
  - cor: string de tamanho 10
  - preço: real
  - a) Escrever a definição da estrutura carro.
  - b) Criar um novo tipo a partir da estrutura elaborada com o comando typedef.
  - c) Declarar o vetor vetcarros do tipo criado na letra b, de tamanho 20;
  - d) Crie uma função para preencher a lista de carros com informações fornecidas pelo usuário;
  - e) Crie uma função que receba a lista de carros, um preço e imprima os carros (marca, cor e ano) que tenham preço igual ou menor ao preço recebido.
  - f) Crie uma função que receba a lista de carros, uma marca, ano e cor e informe se existe ou não um carro com essas características. Se existir, retornar o preço do primeiro carro encontrado. Caso contrário, retorne 0.
2. Elaborar um programa que auxilie no controle de uma fazenda de gado. A base de dados é formada por um conjunto de estruturas (registros) contendo os seguintes campos referentes a cada cabeça de gado:
  - código: código da cabeça de gado,
  - leite: número de litros de leite produzido por semana,
  - alim: quantidade de alimento ingerida por semana - em quilos,
  - nasc: data de nascimento - mês e ano,
  - abate: 'N' (não) ou 'S' (sim)

Crie uma estrutura data para representar a data de nascimento. A estrutura deve ter dois campos: mês e ano. Use a estrutura data para definir o campo data nasc (data de nascimento) da estrutura que representará as informações do gado. O número de cabeças de gado que deverão ser armazenadas deve ser fornecido pelo usuário. Use o typedef nos dois casos (para criação da estrutura para data e para o gado).

Crie funções para:

- a) Preencher a lista com as informações sobre as cabeças de gado que serão digitadas pelo usuário (com exceção do campo abate).
- b) Preencher o campo abate de todos os gados cadastrados considerando que um gado vai para o abate se atender a uma das condições abaixo:
  - tenha mais de 5 anos ou
  - produza menos de 40 litros de leite por semana ou
  - produza entre 50 e 70 litros de leite por semana e ingira mais de 50 quilos de alimento por semana



**MINISTÈRIO DA EDUCAÇÃO**  
**UNIVERSIDADE FEDERAL DE ITAJUBÁ**

Criada pela Lei nº 10.435 – 24/04/2002

3. Considere o código descrito na Figura 1.

```
struct produto {
    int cod;
    float preco;
};

void alteraPreco(struct produto p,float novoP){
    p.preco = novoP;
}

int main(){
    1. struct produto p;
    2. float novoP;
    3. printf("\nDigite o código e o preço do produto: ");
    4. scanf("%d %f", &p.cod, &p.preco); // usuário digita cod = 10 e preço = 120.50
    5. printf("\nAntes da chamada da função: ");
    6. printf("\nCódigo: %d, Preço: %f\n", p.cod, p.preco);
    7. printf("\nDigite o novo preço do produto: ");
    8. scanf("%f", &novoP); // usuário digita novoP = 150
    9. alteraPreco(p,novoP);
    10. printf("\nDepois da chamada da função: ");
    11. printf("\nCódigo: %d, Preço: %f\n", p.cod, p.preco);
    12. system("pause");
}
```

- a. Supondo que o usuário digite os valores especificados nos comentários das linhas 4 e 8, quais valores serão impressos nas linhas 6 e 11?
  - b. Justifique sua resposta
4. Escreva uma função que recebe como parâmetros uma matriz quadrada A de números inteiros (ordem  $n < 10$ ), uma posição  $[i][j]$  da matriz, e calcule a média aritmética dos vizinhos de  $A[i][j]$ , ou seja, a média entre  $A[i-1][j]$ ,  $A[i+1][j]$ ,  $A[i][j+1]$ ,  $A[i][j-1]$ . Desconsidere os vizinhos que não pertencem a matriz (por exemplo, os vizinhos de (0, 0) são somente (0,1) e (1,0)).
5. Escreva uma função que receba como parâmetro uma matriz quadrada A de números inteiros (ordem  $n < 10$ ) e retorne uma matriz Média, onde  $media[i][j]$  é a média aritmética dos vizinhos de  $A[i][j]$ . Para isto, utilize a função da questão anterior.



**MINISTÈRIO DA EDUCAÇÃO**  
**UNIVERSIDADE FEDERAL DE ITAJUBÁ**

Criada pela Lei nº 10.435 – 24/04/2002

6. Escreva um programa que leia uma matriz quadrada A de números inteiros (ordem  $n < 10$ ,  $n$  deve ser fornecido pelo usuário) e imprima a matriz Média utilizando as funções das questões 3 e 4.
7. Escreva um programa em C cujo objetivo é receber 3 valores reais (X, Y e Z) e descobrir se esses valores formam um triângulo e, se sim, identificar o tipo do triângulo formado. O programa deverá ter duas funções: A primeira identifica se os valores informados pelo usuário formam um triângulo. Para tanto, é necessário que a seguinte propriedade seja satisfeita: o comprimento de cada lado de um triângulo é menor do que a soma do comprimento dos outros dois lados. A segunda função visa identificar o tipo do triângulo formado, observando as seguintes definições:
  - a. Triângulo Equilátero: os comprimentos dos 3 lados são iguais.
  - b. Triângulo Isósceles: os comprimentos de 2 lados são iguais.
  - c. Triângulo Escaleno: os comprimentos dos 3 lados são diferentes.

8. Os elementos  $a_{ij}$  de uma matriz inteira A de dimensão N representam os custos de transporte da cidade i para a cidade j. Exemplo: considere a matriz A abaixo:

$$\begin{pmatrix} 4 & 1 & 2 & 3 \\ 5 & 2 & 1 & 400 \\ 2 & 1 & 3 & 8 \\ 7 & 1 & 2 & 5 \end{pmatrix}$$

O custo do itinerário (caminho) 0 3 1 3 3 2 1 0 é

$$a_{03} + a_{31} + a_{13} + a_{33} + a_{32} + a_{21} + a_{10} = 3 + 1 + 400 + 5 + 2 + 1 + 5 = 417$$

O número máximo de cidades é 10. O tamanho máximo de um itinerário é 15. Elaborar uma função que receba a matriz A, um vetor contendo o itinerário, o tamanho do itinerário e retorne o custo do itinerário.

No programa principal, leia a matriz de custo e solicite o usuário que digite o tamanho do itinerário e as cidades do caminho. Utilize a função elaborada para retornar o custo total da viagem.

9. Modifique a função principal acima de forma que o usuário possa digitar a quantidade de caminhos que quiser. Neste caso, você deverá perguntar ao usuário se deseja ou não continuar. Além disso, na nova versão do programa, o usuário não deve fornecer o tamanho do caminho. O fim do caminho deve ser indicado pelo valor -1. Vale lembrar que o tamanho máximo de um caminho é 15 (cidades).
10. Dizemos que uma matriz quadrada inteira é um quadrado mágico se a soma dos elementos de cada linha, a soma dos elementos de cada coluna e a soma dos elementos das diagonais principal e secundária são todas iguais. Elabore uma função que receba



**MINISTÈRIO DA EDUCAÇÃO**  
**UNIVERSIDADE FEDERAL DE ITAJUBÁ**

Criada pela Lei nº 10.435 – 24/04/2002

uma matriz e retorne 1, se a matriz for um quadrado mágico e 0, caso contrário. Utilize a função no programa principal.

11. Fazer uma função que recebe um vetor de elementos inteiros de dimensão N ( $N = 50$ ), um elemento inteiro X e retorne um valor que corresponde ao número de vezes que o elemento X aparece no vetor. Escrever um algoritmo que leia uma matriz A de elementos inteiros de dimensão M x N, ( $M, N = 50$ ) e, utilizando a subrotina acima, gere um vetor V cujos elementos representam quantas vezes o valor zero (0) aparece em cada linha da matriz A. Escreva a matriz A e o vetor V.
12. Seja uma estrutura para descrever os livros de uma loja, contendo os seguintes campos:
- título: string de tamanho 15
  - autor: string de tamanho 20
  - estilo: string de tamanho 10
  - código: inteiro
  - preço: real
- a) Escrever a definição da estrutura livro.
- b) Crie o tipo tlivro a partir da estrutura criada na letra a.

Seja o seguinte trecho de programa:

```
// definição da função menu( ):  
void menu( ) {  
    printf(" Digite I para incluir um livro. \n");  
    printf(" Digite L para listar o nome de todos os livros. \n");  
    printf(" Digite A para procurar livro por autor. \n");  
    printf(" Digite T para procurar livro por título. \n");  
    printf(" Digite E para procurar livro por estilo. \n");  
    printf(" Digite M para calcular a média de preço de todos os livros. \n");  
    printf(" Digite S para sair \n");  
}  
// definição da função principal  
main( ) {  
    tlivro vetlivro[50];  
    char ch;  
    int numlivros = 0; //numero de livros cadastrados  
    menu();  
    scanf("%c", &ch);  
    while (ch != 'S') {  
        switch(ch) {  
            case 'I': incluir_liv(parametros); break( );  
            case 'L': lista_liv(parametros); break;  
        }  
    }  
}
```



**MINISTÈRIO DA EDUCAÇÃO**  
**UNIVERSIDADE FEDERAL DE ITAJUBÁ**

Criada pela Lei nº 10.435 – 24/04/2002

```
        case 'A': proc_autor(parametros); break;
        case 'T': proc_titulo(parametros); break;
        case 'E': proc_estilo(parametros); break;
        case 'M': media_preco(parametros); break;
    } // fim do switch
    menu( );
    scanf("%c", &ch);

}
} // fim da função main.
```

- c) Altere a função main substituindo a estrutura de repetição while por repeat.
  - d) Escrever a funções que estão faltando. Para cada função, verifique o objetivo definido no comentário da função menu e defina quais parâmetros serão necessários.
13. Elabore uma função que receba duas matrizes A e B e suas respectivas dimensões e imprima a matriz resultante da multiplicação de A por B. Caso não seja possível multiplicar A por B, imprima mensagem de erro.
14. Dizemos que uma matriz inteira  $An \times n$  é uma matriz de permutação se em cada linha e em cada coluna houver n-1 elementos nulos e um único elemento igual a 1. Crie uma função que receba como parâmetro uma matriz quadrada de ordem 5 e retorne 1, se a matriz for de permutação ou 0, caso contrário. Utilize a função no programa principal.
15. Considere uma estrutura para armazenar os produtos do estoque de um supermercado. Para cada produto, tem-se os seguintes campos:
- nome: string de tamanho 15
  - setor: caracter
  - quantidade: inteiro
  - preço: real //preço por unidade do produto
- Crie funções para:
- a) Preencher a lista de estoque considerando que há, no máximo, 100 produtos em estoque.
  - b) Imprimir os nomes dos produtos de um determinado setor.
  - c) Retornar a quantidade de capital investido no estoque
  - d) Imprimir a lista de estoque.
  - e) Alterar a quantidade de um produto em estoque (dado o nome do produto e um novo valor de quantidade).