



MINISTÉRIO DA EDUCAÇÃO  
**UNIVERSIDADE FEDERAL DE ITAJUBÁ**

Criada pela Lei nº 10.435 – 24/04/2002

## **COM111- Algoritmos e Estrutura de Dados I**

1. Seja a função troca, cuja assinatura (protótipo) está definida abaixo, uma função que troque o valor de duas variáveis do tipo float, ambas passadas por referência.  
void troca(float \*p, float\*q ).

Implemente a função troca e elabore um programa (somente a função principal) que troque os valores das variáveis float x, y e z, de forma que no final  $x \leq y \leq z$ . A função principal deve chamar a função troca.

2. Um número é considerado um número perfeito se a soma dos seus divisores, exceto o próprio número, é igual ao número. Por exemplo:
  - 6 é perfeito  
Divisores de 6, exceto o 6: 1, 2, 3 (soma = 6)
  - 14 não é perfeito  
Divisores de 14, exceto o 14: 1, 2, 7 (soma 10)
  - 28 é perfeito  
Divisores de 28, exceto o 28: 1,2,4,7,14 (soma 28)

Elabore uma função com a seguinte assinatura  
void (int num, int \*e\_perfeito)

A função deve receber um número inteiro positivo (parâmetro num) e verificar se o número é perfeito. Caso seja, o parâmetro e\_perfeito deve receber 1, caso contrário, deve receber 0. Use a função no programa principal para verificar se um número digitado pelo usuário é perfeito.

### **Leia com atenção as observações abaixo:**

- (A) 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
```



MINISTÈRIO DA EDUCAÇÃO  
**UNIVERSIDADE FEDERAL DE ITAJUBÁ**

Criada pela Lei nº 10.435 – 24/04/2002

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

- (B) 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:

**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;
```



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

Criada pela Lei nº 10.435 – 24/04/2002

(C) 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;
```

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  
}
```

3. 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.

4. Seja uma estrutura para descrever os livros de uma loja, contendo os seguintes campos:

- título: string de tamanho 15
- autor: string de tamanho 20



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

Criada pela Lei nº 10.435 – 24/04/2002

- 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 e exibir os dados do livro por autor. \n");  
    printf(" Digite T para procurar e exibir os dados do livro por título. \n");  
    printf(" Digite E para procurar e exibir os dados do 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;  
            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 do-while.



**MINISTÉRIO DA EDUCAÇÃO**  
**UNIVERSIDADE FEDERAL DE ITAJUBÁ**

Criada pela Lei nº 10.435 – 24/04/2002

- d) Escrever as funções que estão faltando. Para cada função, verifique o objetivo descrito na função menu e defina quais parâmetros serão necessários.

5. Considere o código descrito na Figura 1.

```
void alteraValor(float valor, float novovalor){ valor = novovalor;}
int main(){
    1. float valor, novovalor;
    2. printf("\nDigite o valor original e o novo valor: ");
    3. scanf("%f%f", &valor, &novovalor); // usuário digita 10 e 50
    4. printf("\nAntes da chamada da função: ");
    5. printf("\nValor: %f, Novo Valor: %f\n", valor, novovalor);
    6. alteraPreco(valor,novovalor);
    7. printf("\nDepois da chamada da função: ");
    8. printf("\nValor: %f, Novo Valor: %f\n", valor, novovalor);
    9. system("pause");}
```

- a. Supondo que o usuário digite os valores especificados no comentário da linha 3, quais valores serão impressos nas linhas 5 e 8?
- b. Justifique sua resposta

6. Considere o código descrito na Figura 2.

```
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");}
```

- c. Supondo que o usuário digite os valores especificados nos comentários das linhas 4 e 8, quais valores serão impressos nas linhas 6 e 11?
- d. Justifique sua resposta

7. 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:



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

Criada pela Lei nº 10.435 – 24/04/2002

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

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.

8. Modifique a função principal do exercício 7 de forma que o usuário possa digitar a quantidade de caminhos que quiser. Neste caso, voce 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).
9. 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.
10. Escrever um algoritmo que leia uma matriz A de elementos inteiros de dimensão 50 e, utilizando a função do exercício 10, 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.