



Universidade de São Paulo – São Carlos
Instituto de Ciências Matemáticas e de Computação

Mais exercícios

Profa Rosana Braga

Resolução do exercício de estruturas

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
struct produto {
    int codigo;
    char nome[50];
    int estoque;
    float preco;
};
int main (){
    int a,c;
    char b[50];
    float d;
    struct produto p;
    printf("Digite o codigo do produto %d: ",i+1);
    scanf("%d",&a);
    p.codigo = a;
    ("%f",&d);
```

Parte 1 – sem vetor

Resolução do exercício de estruturas

```
printf("Digite o nome do produto %d: ",i+1);
scanf("%s",b);
strcpy(p.nome,b);
printf("Digite a quantidade estocada do produto %d: ",i+1);
scanf("%d",&c);
p.estoque = c;
printf("Digite o preco do produto %d: ",i+1);
scanf("%f",&d);
p.preco = d;
printf("Informacoes do Produto:"
      "\n\nCodigo: %d"
      "\nNome: %s"
      "\nQuantidade: %d"
      "\nPreco: %f",p.codigo,p.nome,p.estoque,p.preco);
system("pause");
return 0;
}
```

Resolução do exercício de estruturas

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

Parte 2 – com vetor e função

```
void gravaProd (int cod, char* nome, int quant, float preco, struct produto *p);
```

```
struct produto {
    int codigo;
    char nome[50];
    int estoque;
    float preco;
```

```
};
```

```
int main (){
    int a, c, i;
    char b[50];
    float d,soma=0.0;
    struct produto p[5];
```

Resolução do exercício de estruturas

```
for(i=0;i<5;i++){
    if(i>0) printf("\n\n");
    printf("Digite o codigo do produto %d: ",i+1);
    scanf("%d",&a);
    getchar();
    printf("Digite o nome do produto %d: ",i+1);
    gets(b);
    printf("Digite a quantidade estocada do produto %d: ",i+1);
    scanf("%d",&c);
    printf("Digite o preco do produto %d: ",i+1);
    scanf("%f",&d);
    gravaProd(a,b,c,d,&p[i]);
    soma+=p[i].preco*p[i].estoque;
}
```

Resolução do exercício de estruturas

```
printf("Informacoes dos Produtos: \n\n");
    printf("Codigo Nome                Estoque  Preco\n\n");
    for(i=0;i<5;i++){
        printf(" %d  %30s  %d  %f\n",p[i].codigo,p[i].nome,p[i].estoque,p[i].preco);
    }
    printf("\n\nPreco total em estoque: %f\n\n",soma);
    system("pause");
    return 0;
}

void gravaProd (int cod, char* nome, int quant, float preco, struct produto *p){
    (*p).codigo=cod;
    strcpy((*p).nome,nome);
    (*p).estoque=quant;
    (*p).preco=preco;
}
```

Exercício Prático

- 1) este exercício é da aula 7 que está na coteia wiki
- a – **Fazer um programa principal que apresenta o menu a seguir:**
 - 1 – Ler Valores vf e vt (vf=Fornecido e vt=Total da Conta), ambos inteiros
 - 2 – Obter troco
 - 3 – Mostrar cédulas do troco
 - 4 – Sair do programa
- b – **Use switch para chamar 3 funções diferentes, uma para cada opção. Use while para sair do programa. Use ponteiros para passar os parâmetros para as funções. As variáveis deverão estar todas declaradas no programa principal.**
- c – **Na função 3, deve-se calcular e imprimir qual é o menor número possível de notas de 50, 20, 10, 5 e 1 em que o valor do troco pode ser decomposto. Se algum valor gerado for igual a zero, não mostrar (por exemplo, se forem zero notas de 50, não mostrar esse valor). O cálculo só pode ser feito para valores positivos (conferir isso no início da função).**
- d- **Mostrar o teste de mesa para o programa acima**
- Quem já fez, fazer somente o item d e ir pro próximo

Exercício Prático

- 2) Leia 10 números, todos situados entre 10 e 100, inclusive, e imprima a lista dos números lidos, sem repetição de valor, mas com a contagem de quantas vezes ele apareceu.
- Por exemplo, se forem lidos os valores:
 - 2 4 5 7 2 8 9 8 8 7
- Será impresso:
 - 2 2
 - 4 1
 - 5 1
 - 7 2
 - 8 3
 - 9 1

- Dica: use uma matriz [10,2], onde na primeira posição voce armazena o número lido, e na segunda posição a quantidade de vezes que apareceu.

Exercício Prático

- 3) Faça um programa que leia três números d, y e a e informe se eles formam uma data válida. Uma data é válida se dia está entre 1 e 31 (confere-se também os meses com 28, 29 ou 30 dias), o mês está entre 1 e 12, e o ano está entre 1 e 2099.
- Dicas:
 - Implemente uma função para verificar se o ano é bissexto.
 - `bis = (ano%4)==0 && (ano%100)!=0 || (ano%400)==0`
 - O programa principal é bem simples e chama um função passando os parâmetro, d y e a e recebe como retorno 1 se a data é válida e 0 se a data não é válida.