

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

int retornar = 1; // Variavel global pra informar q o "retornar" sempre seja 1

struct produtos
{
    char nome[15];
    float valor;
    int cod;
    int Estoque;
    int Quantidade;
    float Subtotal;
};

struct produtos prod[5] =
{
    {"Pao de forma ", 7.50, 1, 0},
    {"Pao de centeio", 8.69, 2, 0},
    {"Broa de milho ", 5.00, 3, 0},
    {"Sonho          ", 4.50, 4, 0},
    {"Tubaina        ", 3.25, 5, 0}
};
//-----//
//-----//

int CadastrarEstoque()
{
    int i, opc, Qtd = -1;

    printf("\n| Posicao no vetor | Codigo do item |      Item      |\n\n"); // Tabela de cadastro de estoque
    for(i = 0; i < 5; i++) // Laço para construir a tabela
    {
        printf("|          %d          |          %d          | %s |\n", i, prod[i].cod, prod[i].nome); // Exibição da tabela
    }
    while(opc < 1 || opc > 5) // looping caso seja digitado algum numero que nao esteja entre 1 e 5
    {
        printf("\nDigite o codigo do item que deseja alterar o estoque: ");
        scanf("%d", &opc);

        if(opc < 1 || opc > 5) // Condição do looping caso seja digitado algum numero que nao esteja entre 1 e 5
        {
            printf("\n                Código inválido\n");
        }
    }
    while(Qtd < 0) // looping caso seja digitado algum numero abaixo de 0
    {
        printf("\nDigite a quantidade do estoque do item selecionado: ");
        scanf("%d", &Qtd);

        if(Qtd < 0) // Condilão do looping caso seja digitado algum numero abaixo de 0
        {
            printf("\nQuantidade invalida, a mesma deve ser maior ou igual a zero. \n");
        }
    }
    prod[opc-1].Estoque += Qtd; // Armazenando a quantidade em um vetor de estoque (Na struct)

    printf("\nEstoque atualizado com sucesso.\n\n");
}
//-----//
//-----//

int VisualizarEstoque()
{
    int i;

    printf("\n| Item (Codigo) | Nome do item      | Valor (Unidade) | Estoque\n\n"); // Tabela da visualização de estoque
    for(i = 0; i < 5; i++) // Laço para construir a tabela
    {
        printf("| %d          | %s      | R$ %.2f          | %d\n",prod[i].cod, prod[i].nome, prod[i].valor, prod[i].Estoque); // Exibição da
tabela
    }
}
//-----//
//-----//

int RealizarVenda()
{
    int i, j, opc_Troco, opc = 1, item, Qtd, opc_Pag, Num_parcelas;
    float troco, saldo, desconto, valor_parcelas, acrescimo, r = 0; // r = resultado ou valor total

    while(opc == 1)
    {
        printf("\n| Item (Codigo) | Nome do item      | Valor (Unidade) |\n\n"); // Tabela de compras
        for(i = 0; i < 5; i++) // Laço para construir a tabela
        {
            printf("| %d          | %s      | R$ %.2f          |\n",prod[i].cod, prod[i].nome, prod[i].valor); // Exibição da tabela
        }
        printf("\nDigite o código do item que deseja: ");
    }
}
```

```

scanf("%d", &item);

if(item < 1 || item > 5) // Verificar se o código de produto está correto(se o numero digitado está entre 1 e 5)
{
    printf("Codigo de produto invalido, selecione o produto novamente\n");
    continue; // fazendo o papel de "else"
}
printf("Digite a quantidade do produto: "); // Selecao de quantidade do produto
scanf("%d", &Qtd);

if(Qtd <= 0 || Qtd > prod[item-1].Estoque) // Verificar se a quantidade do produto está correto e repetir caso nao seja
{
    printf("\nQuantidade invalida!! Deseja tentar realizar a compra novamente? 1-Sim / 2-Nao: ");
    Qtd = 0;
    scanf("%d", &opc);
    if(opc == 1) // Se a opção digitada não for "1" , volta para o menu
    {
        continue; // voltar ao inicio do loop
    } else {
        return 0; // voltar ao menu inicial //
DESCULPAAAAA ;--;
    }
}
prod[item-1].Quantidade += Qtd; // Adiciona a quantidade de itens ao struct de itens vendidos
prod[item-1].Subtotal = prod[item-1].Quantidade * prod[item-1].valor; // cálculo do total das compras
prod[item-1].Estoque -= prod[item-1].Quantidade; // Subtrai o estoque comprada pela quantidade

r += prod[item-1].Subtotal; // Calculo dos valor total de todos os itens, vai ser usado na linha 144

printf("\nDeseja fazer uma nova compra? 1-Sim \ 2-Nao: ");
scanf("%d", &opc);

struct produtos temp_prod[5]; // Criação de uma struct "temp_prod" apenas para ser usada na ordenação da tabela
for (int i = 0; i < 5; i++)
{
    temp_prod[i] = prod[i];
}
// Ordenar os valores dos produtos do mais caro para o mais barato (BUBBLE SHORT)
for(i = 0; i < 5; i++)
{
    for(j = 0; j < 4 - i; j++)
    {
        if(prod[j].Subtotal < prod[j+1].Subtotal)
        {
            struct produtos temp = prod[j]; // Criação de uma struct "temp" apenas para ser usada na ordenação
            prod[j] = prod[j+1];
            prod[j+1] = temp;
        }
    }
}
//
printf("NOTA FISCAL\n");
printf("|Codigo      | Nome do item      | Valor (Unidade) | Quantidade | Subtotal\n");
for(i = 0; i < 5; i++)
{
    if(prod[i].Quantidade > 0) // apresenta apenas os itens comprados
    {
        printf("| %d          | %s      | R$ %.2f          | %d          | R$ %.2f\n",i+1, prod[i].nome, prod[i].valor, prod[i].Quantidade,
prod[i].Subtotal);
    }
}
printf("-----\n");
printf("|                                |Total      | R$ %.2f\n", r);
printf("-----");

//
    Forma de pagamentos
printf("\nForma de pagamento:\n");
printf("1- À vista\n");
printf("2- À prazo\n");
scanf("%d", &opc_Pag); // Se for digitado 1, ir para linha 180, se for 2,ir para linha 196

while(opc_Pag < 1 || opc_Pag > 2) // Verificar se a opção digitada está correta (se nao estiver entre 1 e 2, fazer looping while)
{
    printf("\nOpção de pagamento invalida!! Digite novamente\n");
    scanf("%d", &opc_Pag);
}
//
    Pagamento a vista;
if(opc_Pag == 1)
{
    if(r <= 50)
    {
        desconto = r * 0.05; // 5% de desconto
        r -= desconto; // r = valor total / valor total = valor total - desconto
    }
    else if(r < 100)
    {
        desconto = r * 0.1; // 10% de desconto
    }
}

```

```

        r -= desconto;          // r = valor total / valor total = valor total - desconto
    }
    else
    {
        desconto = r * 0.18; // 18% de desconto
        r -= desconto;        // r = valor total / valor total = valor total - desconto
    }

    printf("\nPrecisa de troco? 1-Sim / 2-Nao : ");
    scanf("%d", &opc_Troco);

    if(opc_Troco == 1)
    {
        printf("Valor de seu saldo: ");
        scanf("%f", &saldo);

        while(saldo < r) // Verifica se o saldo do cliente é suficiente para realizar a compra
        {
            printf("Saldo insuficiente!! Tente novamente\n");
            scanf("%f", &saldo);
        }
        troco = saldo - r;
        printf("Troco: R$ %.2f\n", troco);
        printf("Desconto de : R$ %.2f\n", desconto);
    }
}
//          Pagamento a prazo
else if(opc_Pag == 2)
{
    printf("\nO Valor total da compra: R$%.2f\n", r);
    printf("Digite o número de parcelas: ");
    scanf("%d", &Num_parcelas);

    while(Num_parcelas <= 0) // Verificar se o numero de parcelas está correto (se não é igual ou menor que 0)
    {
        printf("\nNúmero de parcelas inválido, tem que ser igual ou acima de 1 parcela.");
        scanf("%d", &Num_parcelas);
    }

    if(Num_parcelas > 3) // Se numero de parcelas maior que 3, vai ter 8% de acréscimo
    {
        acrescimo = r * 0.08; // 8% de acréscimo
        r += acrescimo; // r = valor total / valor total = valor total + acrescimo
    }
    else // Caso contrário, 5% de acréscimo (se for menor ou igual a 3 parcelas)
    {
        acrescimo = r * 0.05; // 5% de acréscimo
        r += acrescimo; // r = valor total / valor total = valor total + acrescimo
    }

    valor_parcelas = r / Num_parcelas; // Calcular valor de cada parcela e apresentar numero de parcelas e acréscimos
    printf("\n%d parcelas de: R$ %.2f\n", Num_parcelas, valor_parcelas); // Apresentar número de parcelas e seus respectivos valores
    printf("Acrescimo de: R$ %.2f\n", acrescimo); // Apresentar o acréscimo que foi incluído
    printf("O valor final será de: R$ %.2f\n", valor_parcelas + acrescimo); // Apresentar o valor final incluindo com o acrescimo
}

}
//-----//
//-----//
int main()
{
    setlocale(LC_ALL, "portuguese"); //cod para escrever com ç e acento
    int opc;

    while(retornar == 1)
    {
        // Apresentar menu inicial
        printf("\n1 - Cadastrar estoque\n");
        printf("2 - Visualizar estoque\n");
        printf("3 - Realizar venda\n");
        printf("4 - Sair\n");

        printf("\nDigite a opção: ");
        scanf("%d", &opc);

        switch(opc) // caso seja digitado dentre as opções 1,2 e 3, vão ser levadas as suas respectivas funções
        {
            case 1:
                CadastrarEstoque();
                break;
            case 2:
                VisualizarEstoque();
                break;
            case 3:
                RealizarVenda();
                break;
            case 4:
                // Relatório de itens que foram vendidos

```

```
printf("| Nome do item      | Itens vendidos\n"); // Tabela do relatório
for(int i = 0; i < 5; i++)      // Laço para construir a tabela
{
    printf("| %s      | %d\n", prod[i].nome, prod[i].Quantidade);    // Exibição da tabela
}
printf("\nSaindo do programa...\n");
return 0;
default:
    printf("\nOpção inválida\n");
    break;
}
}
return 0;
}
```