

// Integrantes: Caio Ferreira Trindade, Leonardo dos Santos Figueiredo, Pedro Henrique Santos Sant'Anna, Rafael Vianna Souza Grillo, Renan Xavier Mello

```
#include <ctype.h>
```

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

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

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

```
/*Estrutura do codigo*/
```

```
struct clientes {
```

```
    char nome[100];
```

```
    char sexo;
```

```
    int idade;
```

```
};
```

```
typedef struct clientes clientes;
```

```
struct vendas {
```

```
    clientes clientes;
```

```
    int numItens;
```

```
    char nomeVendedor[50]; // Luiza ou Juca
```

```
    float valorTotal;
```

```
};
```

```
typedef struct vendas vendas;
```

```
/*Funções*/
```

```
void cadastrarVendas(clientes clientes, vendas *vendas);
```

```
void VendaEspecifica(clientes clientes, vendas *vendas);
```

```
void verTodasVendas(clientes clientes, vendas *vendas);
```

```
/*Menu*/
```

```
int main(void) {  
  
    clientes clientes;  
    vendas vendas;  
    int opcao = 0;  
  
    do {  
  
        printf("\n== Seja Bem-Vindo à CasacoCraft ==\n");  
        printf("1) Cadastrar novas vendas\n");  
        printf("2) Ver informações de uma venda específica\n");  
        printf("3) Ver informações de todas as vendas\n");  
        printf("4) Encerra programa\n");  
        printf("\nQual serviço deseja? ");  
        printf("\nDigite aqui: ");  
        scanf("%i", &opcao);  
  
        switch (opcao) {  
  
            case 1:  
                cadastrarVendas(clientes, &vendas);  
                break;  
            case 2:  
                VendaEspecifica(clientes, &vendas);  
                break;  
            case 3:  
                verTodasVendas(clientes, &vendas);  
                break;  
            case 4:  
                system("cls"); // ou clear para limpar o terminal do VsCode  
                printf("\nEncerrando o programa...\n");  
                break;  
            default:  
                printf("\nOpção inválida tente novamente\n");  
                break;  
        }  
    } while (opcao != 4);
```

```
return 0;
}

void cadastrarVendas(clientes clientes, vendas *vendas) {
    int qtdvendas = 0;
    printf("Digite a quantidade de vendas que deseja cadastrar: ");
    scanf("%i", &qtdvendas);
    getchar(); // limpa o buffer

    /* alocação de memória das vendas */
    vendas = malloc(qtdvendas * sizeof(*vendas));
    if (vendas == NULL) {
        printf("Erro: Memória insuficiente!\n");
        return;
    }

    /* Criação do arquivo de texto */
    FILE *arq = fopen("loja.txt", "a");
    if (arq == NULL) {
        printf("Erro ao abrir o arquivo!\n");
        free(vendas);
        return;
    }

    /* Cadastro das vendas */
    for (int i = 0; i < qtdvendas; i++) {
        int nomeTam;

        printf("\n--- Cadastro da venda %d ---\n", i + 1);

        /* Nome do cliente */
        do {
            printf("Digite o nome do cliente: ");
            scanf(" %[^\n]", vendas[i].clientes.nome);
            nomeTam = strlen(vendas[i].clientes.nome);
```

```

if (nomeTam < 3) {
    printf("Nome inválido (mínimo 3 caracteres).\n");
}

} while (nomeTam < 3);

/* Sexo do cliente */
do {
    printf("Digite o sexo do cliente\n(M) Masculino, (F) Feminino, (N) Não desejo responder: ");
    scanf(" %c", &vendas[i].clientes.sexo);

    vendas[i].clientes.sexo = tolower(vendas[i].clientes.sexo);

    if (vendas[i].clientes.sexo != 'm' && vendas[i].clientes.sexo != 'f' && vendas[i].clientes.sexo != 'n') {
        printf("Opção inválida!\n");
    }
} while (vendas[i].clientes.sexo != 'm' && vendas[i].clientes.sexo != 'f' && vendas[i].clientes.sexo != 'n');

/* Idade */
do {
    printf("Digite a idade do cliente: ");
    scanf("%i", &vendas[i].clientes.idade);

    if (vendas[i].clientes.idade < 0) {
        printf("Idade inválida!\n");
    }
} while (vendas[i].clientes.idade < 0);

/* Nome do vendedor */
do {
    printf("Digite o nome do vendedor (Luiza ou Juca): ");
    scanf(" %49[^\\n]", vendas[i].nomeVendedor);

    if (strcmp(vendas[i].nomeVendedor, "Luiza") != 0 && strcmp(vendas[i].nomeVendedor, "Juca") != 0) {
        printf("Nome inválido. Digite Luiza ou Juca.\n");
    }
} while (strcmp(vendas[i].nomeVendedor, "Luiza") != 0 && strcmp(vendas[i].nomeVendedor, "Juca") != 0);

```

```
/* Número de itens */
do {
    printf("Digite a quantidade de itens: ");
    scanf("%i", &vendas[i].numItens);
    if (vendas[i].numItens <= 0) {
        printf("Quantidade inválida!\n");
    }
} while (vendas[i].numItens <= 0);

/* Valor total */
do {
    printf("Digite o valor total da compra: R$ ");
    scanf("%f", &vendas[i].valorTotal);
    if (vendas[i].valorTotal <= 0) {
        printf("Valor inválido!\n");
    }
} while (vendas[i].valorTotal <= 0);

printf("\nVenda cadastrada com sucesso!\n");

/* Escreve no arquivo */
fprintf(arq, "%s;%c;%d;%s;%d;%f\n",
    vendas[i].clientes.nome,
    vendas[i].clientes.sexo,
    vendas[i].clientes.idade,
    vendas[i].nomeVendedor,
    vendas[i].numItens,
    vendas[i].valorTotal);

}

fclose(arq);
free(vendas);
```

```
}

void VendaEspecifico(clientes clientes, vendas *vendas) {
    getchar(); /*limpeza do scanf*/

    char opcao;
    char nome[100];
    char linhas[100];

    /*busca pelo nome do cliente*/
    do {

        int encontrado = 0;
        float somaTotal = 0;
        printf("\nDigite o nome: ");
        scanf("%[^\\n]", nome);
        printf("\n\n");

        /*abriu o arquivo de novo*/
        FILE *arq;
        arq = fopen("loja.txt", "r");
        if (arq == NULL) {
            printf("Erro ao abrir o arquivo");
            return;
        }

        while (fgets(linhas, sizeof(linhas), arq) != NULL) {
            char *t = strtok(linhas, " ");
            if (t != NULL && strcmp(t, nome) == 0) {
                printf("nome: %s\n", t);

                t = strtok(NULL, ";"); // SEXO
                printf("sexo: %s\n", t);
            }
        }
    }
}
```

```

t = strtok(NULL, ";"); // idade
printf("idade: %s\n", t);

t = strtok(NULL, ";"); // nome vendedor
printf("Nome vendedor: %s\n", t);

t = strtok(NULL, ";"); // itens
printf("itens: %s\n", t);

t = strtok(NULL, ";"); // valor total
printf("valor total: %s\n", t);

somaTotal += atof(t);

t = strtok(NULL, " "); // soma total
encontrado = 1;
}

}

if (encontrado) {
    printf("\nSoma total das compras do cliente '%s': R$ %.2f\n", nome, somaTotal);

} else{
    printf("Cliente não encontrado!\n");
}

fclose(arq);
printf("Deseja pesquisar novamente? (s/n): ");
scanf(" %c", &opcao);
getchar();
} while (opcao == 's' || opcao == 'S');

void verTodasVendas(clientes clientes, vendas *vendas) {

//Abrir o arquivo
FILE *arq;

```

```
arq = fopen("loja.txt", "r"); // na função r para ler o arquivo
if (arq == NULL) {
    printf("Erro ao abrir o arquivo\n");
    return;
}

int maior = 0;
char maiornome[100];
int qtdSemSexo = 0;
int menorIdade = 9999;
int qtditens = 0;
int qtdtotalitens = 0;
int qtdvalortotal = 0;
int vendasLuiza = 0;
float qtdtotalvalor = 0.0;
float valorVenda = 0.0;
int cont = 0;
float mediaTotal = 0.0;
float maiorCompraHomem = 0.0;
float valor = 0.0;
char opcao;

float totalMulheres = 0.0;
do{
    /*pega os menores valores dos itens*/
    do {
        printf("\nDigite o valor: ");
        scanf("%f", &valor);
        if (valor < 0) {
            printf("Valor inválido\n");
        }
    } while (valor < 0);
```

```
char linhas[200];
while (fgets(linhas, sizeof(linhas), arq) != NULL) {
```

```
    char *t = strtok(linhas, ";"); //Nome
```

```
    if (t != NULL) {
```

```
        int nome = strlen(t);
```

```
        if (nome > maior) {
```

```
            strcpy(maiorNome, t);
```

```
            maior = nome;
```

```
}
```

```
    t = strtok(NULL, ";"); // SEXO
```

```
    char sexo = tolower(t[0]);
```

```
    if (t != NULL && strcmp(t, "n") == 0) {
```

```
        qtdSemSexo++;
```

```
}
```

```
    t = strtok(NULL, ";"); // idade
```

```
    int idade = atoi(t);
```

```
    if (idade < menorIdade) {
```

```
        menorIdade = idade;
```

```
}
```

```
    t = strtok(NULL, ";"); // nome vendedor
```

```
    if (t != NULL && strcmp(t, "Luiza") == 0) {
```

```
        vendasLuiza++;
```

```
}
```

```
t = strtok(NULL, ";"); // itens

qtdtotalitens += atoi(t);

if (atoi(t) == 1) {
    qtditens++;
}

t = strtok(NULL, ";"); // valor total

valorVenda = atof(t);

qtdtotalvalor += valorVenda;

if (valorVenda < valor) {
    qtdvalortotal++;
}

if (sexo == 'f') {
    totalMulheres += valorVenda;
} else if (sexo == 'm' && valorVenda > maiorCompraHomem) {
    maiorCompraHomem = valorVenda;
}

t = strtok(NULL, ";"); // Média

}

cont++;

}

mediaTotal = qtdtotalvalor / cont;

printf("\n--- Estatísticas Gerais ---\n");

printf("\nTotal de vendas com valor menor que R$ %.2f: %d\n", valor, qtdvalortotal);

printf("Quantidade de vendas com apenas 1 item: %d\n", qtditens);
```

```
printf("Quantidade de vendas realizadas por Luiza: %d\n", vendasLuiza);
printf("Quantidade de clientes que não desejaram informar o sexo: %d\n", qtdSemSexo);
printf("Valor total comprado por mulheres: R$ %.2f\n", totalMulheres);
printf("Quantidade total de itens vendidos: %d\n", qtdtotalitens);
printf("Quantidade total de vendas: %d\n", cont);
printf("Valor total de todas as vendas: R$ %.2f\n", qtdtotalvalor);
printf("Média de vendas: R$ %.2f\n", mediaTotal);
printf("Maior compra feita por um homem: R$ %.2f\n", maiorCompraHomem);
printf("Nome do cliente com maior quantidade de caracteres: %s (%d caracteres)\n", maiornome, maior);

printf("Menor idade entre os clientes: %d\n", menorIdade);

fclose(arq);
printf("\nDeseja pesquisar novamente? (s/n): ");
scanf(" %c", &opcao);
getchar();

} while (opcao == 's' || opcao == 'S');

}
```