**Problema 3**

Crie um programa que permita gravar as informações de um cliente e de uma conta (memória). Uma vez aberta a conta, faça no programa uma simulação de operações de depósito, saque, e exibição de saldo. Nota: você deve considerar como regra para sacar que só é feito o saque se o cliente possuir saldo, e para depositar a regra é que o valor seja positivo.

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <locale.h>**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <string.h>**

**#define MAX\_CLIENTES 5**

**struct s\_cliente {**

**char nome[256];**

**char rg[15];**

**char cpf[15];**

**int idade;**

**float saldo;**

**char sexo;**

**};**

**Logotipo, nome da empresa

Descrição gerada automaticamente**

**struct s\_cliente clientes[MAX\_CLIENTES];**

**int i;**

**int n\_clientes = 0;**

**void cadastraCliente(struct s\_cliente cliente) {**

**clientes[n\_clientes] = cliente;**

**n\_clientes++;**

**setlocale(LC\_ALL, "Portuguese");**

**}**

**struct s\_cliente leDados() {**

**system("cls");**

**struct s\_cliente auxiliar;**

**printf("\n\n");**

**printf("Nome: ");**

**fflush(stdin);**

**fgets(auxiliar.nome, 256, stdin);**

**printf("RG: ");**

**fflush(stdin);**

**fgets(auxiliar.rg, 15, stdin);**

**printf("CPF: ");**

**fflush(stdin);**

**fgets(auxiliar.cpf, 15, stdin);**

**printf("Sexo: ");**

**fflush(stdin);**

**scanf("%c", &auxiliar.sexo);**

**printf("Idade: ");**

**fflush(stdin);**

**scanf("%d", &auxiliar.idade);**

**auxiliar.saldo = 0;**

**return auxiliar;**

**}**

**void imprimeClientes() {**

**system("cls");**

**for (i=0; i<n\_clientes; i++) {**

**printf("Codigo: %d\n", i+1);**

**printf("Nome..: %s", clientes[i].nome);**

**printf("RG....: %s", clientes[i].rg);**

**printf("CPF...: %s", clientes[i].cpf);**

**printf("Sexo..: %c", clientes[i].sexo);**

**Logotipo, nome da empresa

Descrição gerada automaticamente**

**printf("\nIdade.: %d", clientes[i].idade);**

**printf("\nSaldo.: %.2f", clientes[i].saldo);**

**printf("\n+++++++++++++++++++\n");**

**}**

**printf("\nAperte qualquer tecla para retornar ao menu principal");**

**getch();**

**}**

**int menu() {**

**fflush(stdin);**

**system("cls");**

**int opcao;**

**printf("\n\n ==== MENU DE OPCOES ====\n");**

**printf("0 - SAIR \n");**

**printf("1 - INSERIR \n");**

**printf("2 - DEPOSITAR \n");**

**printf("3 - SACAR \n");**

**printf("4 - EXIBIR \n");**

**printf("5 - EXCLUIR \n");**

**printf("\nO que deseja fazer? ");**

**scanf("%d", &opcao);**

**return opcao;**

**}**

**void depositar(posicao) {// DEPOSITAR DINHEIRO**

**int indice = posicao - 1;**

**int i = 0;**

**float valor = 0;**

**float saldo = clientes[indice].saldo;**

**if ((posicao < 1) || (posicao > n\_clientes)) {**

**printf("\nPosicao invalida, aperte qualquer tecla para voltar ao menu!\n");**

**getch();**

**return;**

**}**

**printf("digite o valor do depósito: ");**

**scanf("%f", &valor);**

**clientes[indice].saldo = saldo + valor;**

**system("cls");**

**}**

**Logotipo, nome da empresa

Descrição gerada automaticamente**

**void sacar(int posicao){ //SACAR DINHEIRO**

**int indice = posicao - 1;**

**int i = 0;**

**float valor = 0;**

**float saldo = clientes[indice].saldo;**

**if ((posicao < 1) || (posicao > n\_clientes)) {**

**printf("\nPosicao invalida, aperte qualquer tecla para voltar ao menu!\n");**

**getch();**

**return;**

**}**

**printf("digite o valor do saque: ");**

**scanf("%f", &valor);**

**// While para evitar saldo negativo**

**while (valor > saldo){**

**printf("O valor do saque não pode ser maior que o saldo, por favor digite novamente: ");**

**scanf("%f", &valor);**

**}**

**clientes[indice].saldo = clientes[indice].saldo - valor;**

**system("cls");**

**}**

**void remover(int posicao) {// EXCLUIR CONTA**

**system("cls");**

**int indice = posicao - 1;**

**int i = 0;**

**if ((posicao < 1) || (posicao > n\_clientes)) {**

**printf("\nPosicao invalida, aperte qualquer tecla para voltar ao menu!\n");**

**getch();**

**return;**

**}**

**if (posicao == n\_clientes) {**

**n\_clientes--;**

**Logotipo, nome da empresa

Descrição gerada automaticamente**

**return;**

**}**

**for (i=indice; i<n\_clientes; i++) {**

**clientes[i] = clientes[i+1];**

**system("cls");**

**}**

**printf("Cliente removido com sucesso");**

**printf("Aperte qualquer tecla para retornar ao mennu principal");**

**n\_clientes--;**

**getch();**

**}**

**int main()**

**{**

**system("cls");**

**int opcao\_selecionada;**

**int rem = 0;**

**struct s\_cliente novo\_cliente;**

**int r;**

**do {**

**opcao\_selecionada = menu();**

**switch(opcao\_selecionada) {**

**case 0: break;**

**case 1:**

**if (n\_clientes == MAX\_CLIENTES) {**

**printf("\nBANCO DE DADOS LOTADO!\n");**

**break;**

**}**

**novo\_cliente = leDados();**

**cadastraCliente(novo\_cliente);**

**break;**

**case 2:**

**printf("Digite a posicao do cliente que deseja realizar o depósito: ");**

**scanf("%d", &r);**

**depositar(r);**

**break;**

**case 3:**

**printf("Digite a posicao do cliente que deseja realizar o saque: ");**

**scanf("%d", &r);**

**sacar(r);**

**Logotipo, nome da empresa

Descrição gerada automaticamente**

**break;**

**case 4:**

**imprimeClientes();**

**break;**

**case 5:**

**printf("Digite a posicao que deseja excluir: ");**

**scanf("%d", &r);**

**remover(r);**

**break;**

**default:**

**system("cls");**

**printf("----- OPCAO INVALIDA -----");**

**}**

**}**

**while (opcao\_selecionada != 0);**

**getch();**

**return 0;**

**}**