



UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

Trabalho Prático

Disciplina: Introdução à Computação

Professores: Ricardo Franco

Quantidade máxima de alunos por grupo: 4 alunos

Instruções

O programa em questão pode parecer bastante complexo e intimidador em um primeiro momento. No entanto, podemos decompor em partes menores, de mais fácil compreensão e solução. Para isso, podemos fazer um uso intenso de funções (na linguagem C) de forma a transformá-lo em subproblemas menores que podem ser mais facilmente resolvidos.

Sempre que abordamos um problema de tamanho não trivial é importante uma estratégia de **priorização das operações mais importantes**, sem as quais não é possível passar para a etapa seguinte. Neste caso, **o cadastro de clientes é necessário para que possamos implementar a listagem de clientes**. O cadastro de contas por sua vez, depende da capacidade de poder buscar um cliente por código ou CPF/CNPJ. As operações em contas dependem da existência de um cadastro de contas. Ou seja, a partir da análise dos requisitos, funcionalidades esperadas do programa, é possível deduzir uma ordem de prioridade de implementação.

Baseado nessa filosofia, tente inicialmente implementar um cadastro e listagem de clientes funcional, e só então passe à etapa seguinte.

Problema:

Imaginem que vocês foram contratados para desenvolver um sistema de controle de transações financeiras. O sistema deverá permitir que sejam cadastrados até 100 clientes, 200 contas e 1.000 transações. Deverá haver um menu para seleção de ações e cada ação deverá estar separada em uma função.

- No início da execução e no final de cada ação, o menu abaixo deverá ser apresentado:

===== Bem vindo! =====

Digite um comando para prosseguir:

C - Gerenciar Clientes

T - Gerenciar Contas

S - Sair



UNIVERSIDADE FEDERAL DE GOIÁS INSTITUTO DE INFORMÁTICA

- Ao pressionar a tecla C no menu principal, o usuário será redirecionado para outro menu:

===== Gerenciar Clientes =====

Digite um comando para prosseguir:

C - Cadastrar um cliente
L - Listar todos os clientes cadastrados
B - Buscar cliente já cadastrado
A - Atualizar um cliente cadastrado
E - Excluir um cliente cadastrado
S - Sair

- Ao pressionar a tecla T no menu principal, o usuário será redirecionado para outro menu:

Digite um comando para prosseguir:

R - Listagem de todas as contas cadastradas.
C - Cadastrar uma conta para um cliente.
L - Listar todas as contas de um cliente.
W - Realizar um saque em uma conta.
D - Realizar um depósito em uma conta.
T - Realizar transferência entre contas.
E - Exibir extrato de uma conta.
S - Sair

Gestão de Clientes:

- Um cliente deve conter código, nome, CNPJ/CPF, telefone e endereço.

- No menu de cadastro do cliente, o sistema deve pedir para que o usuário informe os dados de um cliente (código, nome, CNPJ/CPF, telefone, endereço) e o salve (opcionalmente em arquivo, de forma que seja possível acessar suas informações posteriormente). Caso já exista um cliente com o mesmo código ou CNPJ/CPF cadastrado o sistema deverá exibir uma mensagem de erro: "Cliente já cadastrado".

- No menu de listagem de clientes, o sistema deve apresentar uma listagem contendo todos os clientes cadastrados. Caso não exista nenhum cliente cadastrado ainda, a mensagem "Nenhum cliente cadastrado." deve ser exibida. A listagem deverá estar em ordem alfabética do nome do cliente.

- No menu de busca por cliente, o usuário terá a opção de buscar por nome, código ou CPF/CNPJ. Caso encontrado, os dados do cliente em questão devem ser exibidos na tela para o usuário. Caso contrário, uma mensagem de erro deve ser impressa.



UNIVERSIDADE FEDERAL DE GOIÁS INSTITUTO DE INFORMÁTICA

- No menu de atualização de clientes, o usuário deverá informar o código ou CPF/CNPJ do cliente que deseja atualizar. Em seguida, o sistema deverá pedir todos os dados novamente para que possam ser atualizados. Caso o cliente não seja encontrado, uma mensagem de erro “Cliente não encontrado” deverá ser exibida.

- No menu de exclusão de cliente, o usuário deve informar o código ou CPF/CNPJ do cliente que deseja excluir. Caso o cliente não exista, uma mensagem de erro “Cliente não encontrado” deverá ser exibida. Caso o cliente exista, o sistema deverá imprimir os dados desse cliente e pedir confirmação quanto a exclusão. Caso o usuário aceite a exclusão, o cliente deverá ser removido do sistema de forma a não haver lacunas (linhas em branco).

Gestão de Contas:

- Uma conta deve conter agência, número da conta, cliente e saldo. Cliente deverá referenciar um cliente cadastrado.

- No menu de cadastro de contas, o usuário deverá informar o código ou CPF/CNPJ de um cliente. Caso o cliente exista, deverá pedir os dados da conta (agência e número da conta) e salvar. O saldo inicial da conta deverá ser 0. Caso já exista uma conta com a mesma agência e número uma mensagem “Conta já existente” deverá ser exibida.

- No menu para listar todas as contas, as contas deverão ser listadas em ordem alfabética do nome do cliente. Se o cliente tiver mais de uma conta, suas contas deverão ser listadas em ordem decrescente do saldo da conta. Os dados de um mesmo cliente deverão ser impressos apenas uma vez.

- No menu de listagem de contas, o usuário deverá informar o código ou CPF/CNPJ de um cliente. Caso o cliente exista, deverá exibir os dados de todas as contas daquele cliente. Se não existir nenhuma conta cadastrada para tal cliente, uma mensagem “Nenhuma conta cadastrada” deverá ser exibida. A listagem deverá estar sempre em ordem decrescente do saldo da conta.

- Uma transação deve conter o tipo (com as possibilidades: DEBITO e CREDITO), valor, data, conta (referenciando uma conta cadastrada) e descrição. A descrição é um campo texto de até 100 posições.

- No menu de saque em uma conta, o sistema deverá pedir ao usuário a agência e número da conta. Caso a conta exista, deverão ser exibidos os dados do cliente da conta e saldo da mesma. Em seguida, o sistema deverá pedir para o usuário informar o valor que deseja sacar. O sistema deve verificar se o valor informado é menor que o saldo da conta. Caso positivo, o sistema deverá pedir ao usuário uma descrição para tal operação. Em seguida, o saldo da conta deverá ser atualizado, subtraindo o valor de saque, e uma transação correspondente a esse saque deverá ser registrada. A transação deverá ser de DEBITO, para a conta selecionada, na data atual e com o valor e descrição informados



UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

pelo usuário. Por fim, o sistema deverá imprimir quais notas, e a quantidade de cada nota, foram entregues para o usuário. Considerar notas com os valores: 2, 5, 10, 20, 50, 100 e 200 reais.

- No menu de depósito em uma conta, o sistema deverá pedir ao usuário a agência e número da conta. Caso a conta exista, deverão ser exibidos os dados do cliente da conta. Em seguida, o sistema deverá pedir para o usuário informar o valor que deseja depositar e uma descrição para tal depósito. Em seguida, o saldo da conta deverá ser atualizado, somando o valor de depósito, e uma transação correspondente a esse depósito deverá ser registrada. A transação deverá ser de CREDITO, para a conta selecionada, na data atual e com o valor e descrição informados pelo usuário.

- No menu de transferência entre contas, o sistema deverá pedir ao usuário a agência e número da conta de origem (onde será realizado o saque). Caso a conta exista, deverão ser exibidos os dados do cliente da conta e saldo da mesma. Em seguida, o sistema deverá pedir para o usuário informar a agência e número da conta de destino (onde será realizado o depósito).

Caso a conta exista, deverão ser exibidos os dados do cliente da conta. O sistema, então, deverá pedir o valor que o usuário deseja transferir e verificar se o valor informado é menor que o saldo da conta de origem. Caso positivo, o sistema deverá atualizar o saldo das duas contas, subtraindo o valor da conta de origem e somando à conta de destino. As duas transações deverão ser registradas. A transação de DEBITO na conta origem ficará com a descrição “Transferência para conta: {agência}-{conta}” e a transação de CREDITO na conta destino ficará com a descrição “Transferência de conta {agência}-{conta}” (colocar no número da agência no lugar de {agência} e o número da conta no lugar de {conta}).

- Utilizar as funções de saque e depósito definidas nos tópicos anteriores para realização de uma transferência. Os dados só deverão ser salvos se as duas operações (saque e depósito) tiverem sido realizadas com sucesso.

- No menu para extrato de uma conta, o sistema deverá pedir ao usuário a agência e número da conta. Caso a conta exista, deverão ser exibidos os dados do cliente da conta e saldo da mesma. Em seguida, o sistema deverá pedir ao usuário para que informe a quantidade de dias que deseja exibir no extrato. O sistema, então, deverá listar todas as transações realizadas na conta para o período informado, em ordem decrescente da data da transação.

- As transações e contas podem ser salvas em arquivos diferentes (uma para clientes, outro para transações e outro para contas).

Exemplo de código para menu:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
void menu_gerenciar_cliente(void) {
// GERENCIAR CLIENTES
}

void menu_gerenciar_conta(void) {
// GERENCIAR CONTAS
}

void menu_principal(void) {
    char opcao;
    do {
        printf("\n===== Bem Vindo! =====\n");
        printf("Digite um comando para prosseguir:\n");
        printf("C - Gerenciar Clientes\n");
        printf("T - Gerenciar Contas\n");
        printf("S - Sair\n");
        printf("\nEscolha: ");
        scanf(" %c", &opcao);
        switch (opcao) {
            case 'C':
                menu_gerenciar_cliente();
                break;
            case 'T':
                menu_gerenciar_cliente();
                break;
            case 'S':
                break;
            default:
```

```
        printf("\n** Comando inválido digite C, T ou S  
para prosseguir **\n");  
    }  
    } while (opcao != 'S');  
    printf("\nAté mais!\n");  
}  
int main(void) {  
    menu_principal();  
    return 0;  
}
```

Exemplo de código para declaração de estrutura de dados
(tipo de dados) para representar o cliente

```
#define TAM_MAX_NOME 30  
#define TAM_CPF 11  
#define TAM_TELEFONE 15  
#define TAM_MAX_ENDERECO 50  
  
struct cliente {  
    int codigo;  
    char nome[TAM_MAX_NOME + 1];  
    char cpf[TAM_CPF + 1];  
    char telefone[TAM_TELEFONE + 1];  
    char endereco[TAM_MAX_ENDERECO + 1];  
};
```