

Exercício-programa 2

Minimercado

Data de entrega: 12/06/2016 - 23:55

A pequena empresa

Um minimercado de uma cidade do interior de SP está se preparando para sobreviver após a construção do Carrefour em sua cidade. Essa construção levará meio ano, e esse é o tempo que o minimercado terá para se modernizar. Nesse mercadinho só informações de produtos comprados e vendidos, impostos e tributos pagos são guardadas. No final do mês, os donos fazem o balanço com apenas essas informações não sabendo para quem vendeu seus produtos. Não há funcionários e apenas 3 irmãos, os donos, trabalham e administram tudo.

A ideia dos donos é comprar um notebook para guardar alguma informação sobre seus clientes. Só que eles não tem conhecimentos algum de tecnologia, mas sabem que facebook e whatsapp está fazendo muito sucesso e por isso há um mês passaram anotar as informações de seus clientes numa agenda.

Na reunião de balanço mensal um deles disse:

— Vamos ter que comprar uma agenda maior. Essa de 200 folhas acabou.

— Chega de papel! Tá muito confuso! — outro respondeu balançando a cabeça triste. — Só que é muito difícil para nós achar uma solução de computador com "ZAP ZAP" e Face!

— “Tamo” perdido! “Vamo” falir com o Carrefour aqui perto! — O primeiro respondeu.

— Lembrei de uma coisa. O filho do nosso vizinho faz Engenharia na POLI, lá em SP. Ele ainda não está de férias, “mas” ele tem tempo de sobra e podemos falar com ele. Ele pode nos ajudar — falou o terceiro irmão e os outros dois concordaram.

Uma semana depois eles conversaram com o futuro engenheiro que ficou de pensar em uma solução. No dia seguinte, ele sugeriu o seguinte sistema para manter o cadastro, com Face e "ZAP ZAP" (número do celular) além das compras realizadas. Os donos gostaram do sistema e contrataram o serviço do aluno.

Solução

O sistema de cadastro de clientes e produtos deverá ter a seguinte estrutura de informação:

- Cliente
 - cpf: int
 - nome: string, 70 caracteres.
 - fone: string, 20 caracteres.
 - email: string, 50 caracteres.
 - produtos: uma lista ligada de produtos.
- Produto
 - id: string de 5 caracteres.
 - nome: string, 50 caracteres.
 - marca: string, 50 caracteres.
 - preco: double.

– proximo: ponteiro para Produto

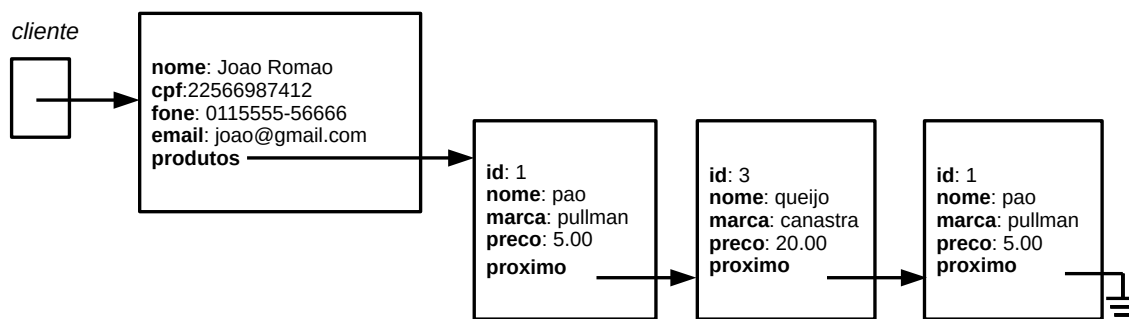
Por exemplo, um cliente

	Cliente
nome	“Joao Romao”
cpf	“22566987412”
fone	“0115555-56666”
email	“joao@gmail.com”

que adquiriu os produtos:

	Produto 1	Produto 2	Produto 3
id	1	3	1
nome	“pao”	“queijo”	“pao”
marca	“pullman”	“canastra”	“pullman”
preco	5.00	20.00	5.00

teria como estrutura na memória o seguinte esquema abaixo:



O que você deve fazer

Você deve implementar um sistema que apresenta um “menu” (*interface* com o usuário) com as seguintes opções:

1. Inserir um novo cliente: dado um cliente, inserí-lo no vetor de clientes. O sistema não deve aceitar inserir um cliente com cpf já existem. Os clientes devem ser inseridos de forma ordenada (crescente) pelo cpf;

```
void inserir_cliente(Cliente *cliente, Cliente *clientes[])
```

2. Inserir um novo produto na lista de produtos adquiridos pelo cliente: dado um cpf e um produto, inserir o produto com todas as suas informações no começo da lista de produtos adquiridos do cliente que possui aquele cpf;

```
void inserir_produto(Produto *produto, int cpf, Cliente *clientes[])
```

3. Buscar o cadastro de um cliente: dado o seu cpf, apresentar o nome e telefone do cliente. Esta busca deve ser eficiente ($O(\log_2 n)$). Sugestão: como o vetor ficará ordenado, use a busca binária.

```
void buscar_cliente(int cpf, Cliente *clientes[])
```

4. Remover um cliente: dado o cpf de um cliente, remover o seu cadastro. Rejeitar caso o cpf não seja de um cliente. Dica: use a busca acima;

```
void remover_cliente(int cpf, Cliente *clientes[])
```

5. Atualizar o telefone de um cliente: dado o cpf e o novo telefone de um cliente, atualizar o do cadastro do cliente com o telefone.

```
void atualizar_telefone(char novo_telefone[20], int cpf, Cliente *
    clientes[])
```

6. Desfazer última aquisição: dado um cpf, remover o último produto adquirido por esse cliente. Dica como a inserção nessa lista de produtos é feita no começo, remova o primeiro elemento da lista.

```
void desfazer_ultima_compra(int cpf, Cliente *clientes[])
```

7. Imprimir a lista de produtos adquiridos por um cliente dado seu cpf;

```
void listar_produtos(int cpf, Cliente *clientes[])
```

8. Imprimir lista de clientes: listar todas as informações de cada cliente, incluindo a lista de produtos adquiridos)

```
void listar_tudo(Cliente *clientes[])
```

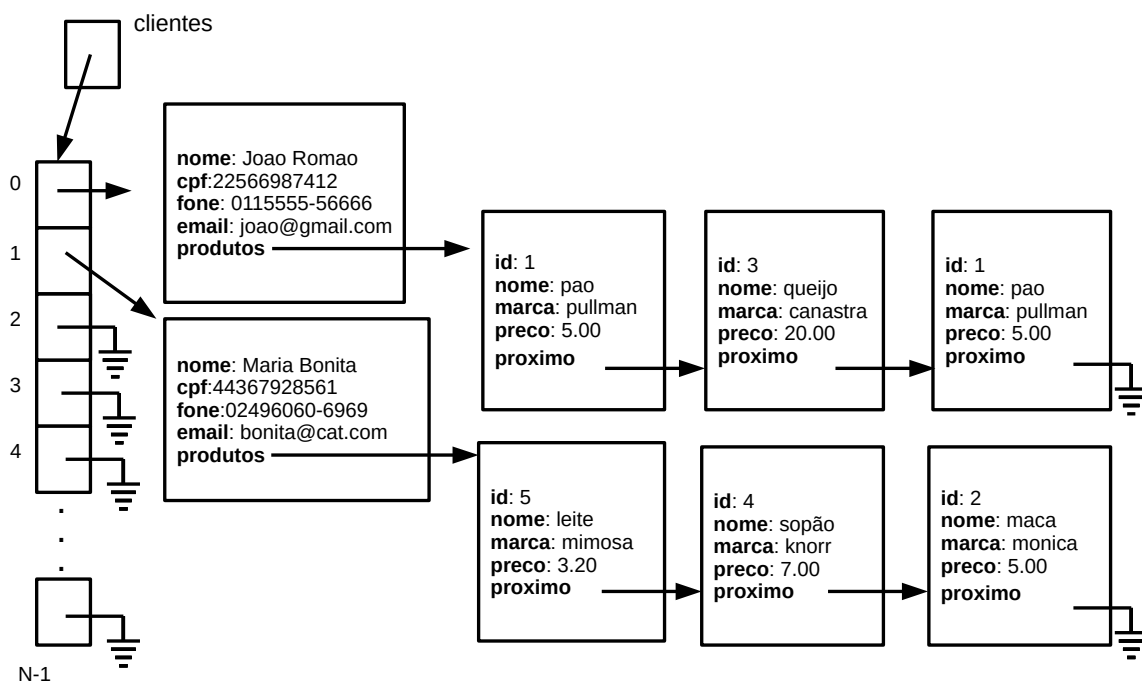
9. Sair e salvar: finalizar o programa e salvar num arquivo txt a lista de clientes com suas compras. Dica: use o listar_tudo.

Um dos donos ficou desconfiado da capacidade do aluno em fazer o sistema. Então pediu a ele para mostrar como ficaria guardado na memória do computador os dados de seus clientes após as primeiras compras.

— O João, vizinho meu, compra 1 saco de pão pullman que custa R\$5,00 pela manhã e a tarde gosta comprar 1 queijo canastra que custa R\$20,00. O nome dele é João Romão, fone 0115555-56666, email: joao@gmail.com, cpf: 22566987412. Já a Maria Bonita compra no almoço 1 saco de maçã da turma da Mônica, que custa R\$5,00, para a janta 1 sopão da Knorr que custa R\$7,00 e 1 caixa de leite Mimososa que custa R\$3,20 e seu fone é 02496060-6969, email: bonita@cat.com, e cpf: 44367928561. Antes de fechar o mercado, o João volta aqui, e compra de novo outro saco de pão pullman.

— Como fica tudo? — pergunta um dos donos.

— Fica assim — respondeu o aluno, mostrando a figura abaixo.



- E quanto vale N? - pergunta o outro dono?
- Podemos considerar que N é dado pelo usuário do sistema e deve ser lido via teclado antes de aparecer o “Menu” das opções. - respondeu o aluno.

Tarefa

Parabéns, você é o aluno da Poli contratado. Desenvolva um programa em C para gerenciar esse sistema proposto. O armazenamento dos clientes pode ser feita por vetor (ou listas ligadas) e o armazenamento de produtos por listas ligadas como visto em aula.

Critérios de avaliação

Cada uma das 9 funcionalidades vale 1 ponto. A interface e clareza do código valem 1 ponto cada também; ou seja o trabalho tem 1 ponto de bônus. Aproveite.

Entrega do trabalho

Deve ser entregue no PACA um arquivo compactado (.zip) contendo os 3 arquivos abaixo:

- clientes.h com a estrutura cliente e estrutura produto
- sistema.c com o sistema implementado
- main.c programa principal com o menu (a *interface*).