Universidade Estadual de Campinas
Instituto de Computação
Algoritmos e Programação de Computadores - MC102QRSTWY

Laboratório em Casa 04

Prof. Arnaldo Moura e Prof. Lehilton Pedrosa

Prazo para entrega: 08/05/2016 às 21:59:59

Agenda Telefônica

A senhora Marta é muito organizada com suas tarefas cotidianas, tanto profissionais como pessoais. No entanto, ela gostaria de organizar melhor sua agenda telefônica separando contatos que são pessoais e contatos profissionais.

A senhora Marta gostaria de uma agenda onde ela pudesse inserir, consultar, listar contatos pessoais, listar contatos profissionais, listar todos, excluir e alterar os dados de contatos. Os contatos devem estar organizados por tipo (pessoal/profissional).

Você deve fazer um programa que ajude a senhora Marta a gerenciar a sua agenda telefônica para no máximo 10 contatos de cada tipo. Os dados que a agenda irá armazenar são:

- **Nome:** string
- **CPF:** inteiro com 11 dígitos
- **Endereço:** string
- **Telefone Residencial/Comercial:** inteiro com 10 dígitos (inclui o DDD com dois dígitos)
- **Telefone Celular** inteiro com 11 dígitos (inclui o DDD com dois dígitos)

É obrigatório o uso de vetores para o armazenamento dos dados dos contatos. As declarações dos tipos de dados informados (strings e inteiros) também são obrigatórias.
Crie um vetor para cada tipo de dado do contato. O contato de índice i mantém o mesmo índice em todos os vetores.
ex: nomePessoal[0], cpfPessoal[0],endPessoal[0],foneRes[0], foneCel[0] pertencem ao contato no índice da primeira posição.
Os tipos de contatos devem ser armazenados em estruturas diferentes e são representados por um número inteiro, a saber:

- **Pessoal:** 1

- **Profissional:** 2

O programa deve emitir mensagens de sucesso, insucesso, agenda vazia ou agenda lotada sempre que necessário (veja na descrição das funções).

Dicas de programação

- Para leitura das variáveis de nome e de endereço use a seguinte sintaxe:
  scanf(" %[^\n]s",nome_da_variavel); //note que não tem o & e tem um espaço entre a aspa e o %

- Para manipular corretamente o vetor de strings use:
  typedef char string[60];//deve ficar fora da função principal e 60 é a quantidade máxima de caracteres
  string nome[10], endereco[10]; //deve ficar dentro da função principal e 10 é a quantidade máxima de contatos

- Passe a referência dos vetores como parâmetro nas chamadas das funções

- Estes protótipos podem ser utilizados
  - int inserir(tipo <variavel_de_nome>,tipo <variavel_de_cpf>, tipo <variavel_de_end>, tipo <variavel_de_fone1>,tipo <variavel_de_fone2>,tipo <variavel_de_tipo_de_contato>)
  - int excluir(tipo <variavel_de_nome>,tipo <variavel_de_cpf>, tipo <variavel_de_end>, tipo <variavel_de_fone1>,tipo <variavel_de_fone2>,tipo <variavel_de_tipo_de_contato>,tipo <variavel_de_nome_para_excluir>)
  - int alterar(tipo <variavel_de_nome>,tipo <variavel_de_cpf>, tipo <variavel_de_end>, tipo <variavel_de_fone1>,tipo <variavel_de_fone2>,tipo <variavel_de_tipo_de_contato>,tipo <variavel_de_nome_para_alterar>)
  - int listar(tipo <variavel_de_nome>,tipo <variavel_de_cpf>, tipo <variavel_de_end>, tipo <variavel_de_fone1>,tipo <variavel_de_fone2>,tipo <variavel_de_tipo_de_contato>)
  - int buscar(tipo <variavel_de_nome>, tipo <variavel_de_nome_para_buscarr>,tipo <variavel_de_tipo_de_contato>)

- Como boa prática, imprima as mensagens na função principal. As demais funções somente devem fazer as respectivas operações e retornar valores lógicos. Futuramente, você poderá usar estas funções em outros problemas e talvez as mensagens não sejam as mesmas, mas os valores lógicos sim.

Descrição as funções

- **A função principal deve conter a escolha de opções através de um menu de opções:**

  Opções:
  - – inserir
  - – excluir
  - – alterar
  - – buscar
  - – listar pessoais
  - – listar profissionais
  - – listar todos
  - – Sair

- **Na opção INSERIR deve conter:**
  - Leitura do tipo de contato que será inserido
  - Leitura dos dados a serem inseridos (assuma que todas as entradas são válidas)
  - Verificação se a agenda esta lotada ou não
  - Ao fim da inserção deve ser emitido uma mensagem de sucesso ou de agenda lotada
  - Mensagens:
    - **Inserido com sucesso!**
    - **Desculpe, agenda lotada!**
- **Na opção EXCLUIR deve conter:**
  - Leitura do tipo de contato que será excluído
  - Leitura do nome que deseja excluir
  - Verificação se a agenda está vazia ou não
  - Ao fim da exclusão deve ser emitido uma mensagem de sucesso ou insucesso
  - Mensagens:
    - **Desculpe, contato <nome_do_contato> nao existe!**
    - **Excluido com sucesso!**
- **Na opção ALTERAR deve conter:**
  - Leitura do tipo de contato que será alterado
  - Leitura do nome para alteração dos dados
  - Verificação se a agenda está vazia ou não
  - Exceto o *Nome*, a alteração deve ser feita em todos os outros dados (CPF, Endereço...)
    - Na leitura dos novos dados, assuma que todos os dados são válidos
  - Ao fim da alteração deve ser emitido uma mensagem de sucesso ou insucesso;
  - Mensagens:
    - **Desculpe, contato <nome_do_contato> nao existe!**
    - **Alterado com sucesso!**
- **Na opção BUSCAR deve conter:**
  - Leitura do tipo de contato que será buscado
  - Leitura do nome para busca dos dados
  - Verificação se a agenda está vazia ou não
  - Impressão de todos os dados do contato na seguinte ordem (cada dado em uma linha):
    - Nome
    - CPF
    - Endereço
    - Telefone Residencial/Comercial
    - Telefone Celular
  - Ao fim da busca deve ser emitido uma mensagem de sucesso ou insucesso
  - Mensagens:

    - **Desculpe, contato <nome_do_contato> nao existe!**
    - **Buscado com sucesso!**

Na opção LISTAR PESSOAIS deve conter:

- Verificação se a agenda está vazia ou não
- Listar todos os dados de todos os contatos pessoais
- Impressão de todos os dados dos contatos na seguinte ordem (cada dado em uma linha):
    - Nome
    - CPF
    - Endereço
    - Telefone Residencial
    - Telefone Celular
- Ao fim da listagem deve ser emitido uma mensagem de sucesso ou de agenda vazia
- Mensagens:
    - **Listado com sucesso!**
    - **Desculpe, agenda vazia!**

Na opção LISTAR PROFISSIONAIS deve conter:

- Verificação se a agenda está vazia ou não
- Listar todos os dados de todos os contatos profissionais
- Impressão de todos os dados dos contatos na seguinte ordem (cada dado em uma linha):
    - Nome
    - CPF
    - Endereço
    - Telefone Comercial
    - Telefone Celular
- Ao fim da listagem deve ser emitido uma mensagem de sucesso ou de agenda vazia
- Mensagens:
    - **Listado com sucesso!**
    - **Desculpe, agenda vazia!**

Na opção LISTAR TODOS deve conter:

- Verificação se cada tipo de agenda está vazia ou não
- Listar todos os dados de todos os contatos na seguinte ordem
    - Pessoais
    - Profissionais
- Impressão de todos os dados dos contatos na seguinte ordem (cada dado em uma linha):
    - Nome
    - CPF
    - Endereço
    - Telefone Residencial/Comercial

    - Telefone Celular
- Ao fim de cada listagem deve ser emitido uma mensagem de sucesso ou de agenda vazia
- Mensagens para cada tipo de agenda:
    - **Listado com sucesso!**
    - **Desculpe, agenda vazia!**

Na opção SAIR deve conter:

- O encerramento do programa
- Mensagem:
    - **Obrigado!**

ENTRADA

Obs.: Somente a primeira linha da entrada é padronizada e contém um número inteiro com a primeira opção. As demais linhas de entrada serão aleatórias, no entanto, elas seguem uma lógica de acordo com o fluxo da opção. Você pode assumir que todas as opções tem fluxo de entradas válidas.

EXEMPLO

1 *opção*
1 *tipo*
Luis Fernando Dos Santos Lacalle *nome*
12345678910 *cpf*
Rua da travessa numero 11 *endereço*
1912345678 *fone res*
19123456789 *fone cel*
1 *opção*
2 *tipo*
Mateus Joaquim *nome*
19876543210 *cpf*
Avenida Dutra numero 123 *endereço*
1132145678 *fone com*
11987654321 *fone cel*
1 *opção*
1 *tipo*
Rafael Figueredo Prudencio *nome*
65432134519 *cpf*
Travessa Assis numero 7 *endereço*
1587654321 *fone res*
15976539281 *fone cel*
2 *opção*
1 *tipo*
Luis Fernando Dos Santos Lacalle *nome*
4 *opção*
1 *tipo*
Luis Fernando Dos Santos Lacalle *nome*
3 *opção*
2 *tipo*
Mateus Joaquim *nome*
11111111111 *cpf*
Travessa Assis numero 10 *endereço*

2222222222 *fone com*
33333333333 *fone cel*
4 *opção*
2 *tipo*
Mateus Joaquim *nome*
1 *opção*
1 *tipo*
Regina Maria Alexandre F. De Oliveira *nome*
65432134519 *cpf*
Travessa Assis numero 7 *endereço*
1587654321 *fone res*
15976539281 *fone cel*
5 *opção*
6 *opção*
7 *opção*
0 *opção*

Em *vermelho* a descrição do dado!

SAIDA

Inserido com sucesso!
Inserido com sucesso!
Inserido com sucesso!
Excluido com sucesso!
Desculpe, contato Luis Fernando Dos Santos Lacalle nao existe!
Alterado com sucesso!
Mateus Joaquim
11111111111
Travessa Assis numero 10
2222222222
33333333333
Buscado com sucesso!
Inserido com sucesso!
Rafael Figueredo Prudencio
65432134519
Travessa Assis numero 7
1587654321
15976539281
Regina Maria Alexandre F. De Oliveira
65432134519
Travessa Assis numero 7
1587654321
15976539281
Listado com sucesso!
Mateus Joaquim
11111111111
Travessa Assis numero 10
2222222222
33333333333
Listado com sucesso!

Rafael Figueredo Prudencio
65432134519
Travessa Assis numero 7
1587654321
15976539281
Regina Maria Alexandre F. De Oliveira
65432134519
Travessa Assis numero 7
1587654321
15976539281
Listado com sucesso!
Mateus Joaquim
11111111111
Travessa Assis numero 10
2222222222
33333333333
Listado com sucesso!
Obrigado!

- **Observações**
  - *O programa deve ser submetido em C (.c)*
  - *Você deve incluir, no início do seu programa, uma breve descrição dos objetivos do programa, da entrada e da saída, além do seu nome, seu RA e sua turma*
  - *Faça comentários e indentação do seu código*
  - *O aluno pode assumir que todas as linhas da entrada terminam com o fim-de-linha*
  - *Todas as linhas da saída devem terminar com o fim-de-linha*
  - ***Estruturas do tipo registro (struct) não está permitida***
  - ***Obrigatório, no mínimo, o uso das funções especificadas***
  - **Se julgar necessário, você pode fazer uso de mais funções além das listadas neste enunciado**
  - *O número máximo de submissões é 15*
    - *O comando de compilação utilizado será:*

gcc -std=c99 -pedantic -Wall -lm labSemanal04.c -o labSemanal04