**Instituto de Computação da UNICAMP**

**MC102: Algoritmos e Programação de Computadores**

**Segundo Semestre de 2010 - Turmas C e D**

**Laboratório Nº 07**

|  |  |
| --- | --- |
| Docente*:* | *Leonardo P. Tizzei* |
| Monitores*:* | *Elizabeth C. A. Sousa* |
|  | *Alex Brandt* |

Escreva um programa em linguagem C que gerencia uma agenda de contatos e permite buscas por nome e por telefone, listar todos os contatos ordenados por nome e adiciona uma nova funcionalidade: remoção por telefone.

**Instruções:**

O programa poderá usar o mesmo código do laboratório 05, porém com as seguintes alterações:  
\* A ordenação por nome deverá ser feita com uso do **algoritmo recursivo Quick Sort**.  
\* A busca por nome deverá ser feita com uso de **busca binária recursiva**.  
\* A opção de remoção de um contato pelo telefone deve existir.  
  
Seguem as instruções como no laboratório 05 com as devidas alterações:  
O programa receberá comandos com seus possíveis parâmetros como descrito abaixo.

i' '<telefone>' '<nome>

: faz a inserção de um novo contato na agenda, telefone e nome são separados por espaço.

r' '<telefone>

: faz a busca e remoção de um contato com o telefone dado, este é separado por espaço.

n' '<nome>

: faz a busca por um contato com o nome dado, este é separado por espaço.

t' '<telefone>

: faz a busca por um contato com o telefone dado, este é separado por espaço.

p

: faz a impressão de todos os contatos em ordem alfabética.

f

: Finaliza a execução do programa.

**Entrada:**

A entrada será composta por uma sequência de comandos, um em cada linha, terminada por um comando 'f'.

**Exemplo de entrada:**

i 32095631 Camila Yukie Takiyama  
p  
n Camila Yukie Takiyama  
n Alex Fernando Brant  
t 32090232  
r 32090232  
t 32095631  
f

Para mais exemplos vide testes.

**Saída:**

A saída será constituída pela execução de cada um dos comandos da entrada.  
Após a execução de cada comando, uma linha deverá ser pulada, exceto na inserção.  
  
A impressão de cada contato da agenda é igual nas três opções em que isso ocorre e é formatada da seguinte forma:

printf("%-40s %d\n", <nome>, <telefone>);

Quando o contato não for encontrado a seguinte mensagem deve ser impressa:

"Contato nao encontrado!\n\n"

Para o comando de remoção, a saída tem duas opções:  
\* O contato não existe na agenda, sendo assim a mensagem abaixo deve ser impressa:

"Nao foi possivel remover: Contato inexistente!\n\n"

\* O contato existe na agenda, sendo assim a mensagem abaixo deve ser impressa e o contato removido:

"Contato: %-40s %d\nFoi removido com sucesso!\n\n"

Note a existência da quebra de linha na resposta!Esta aparece repetida para que uma linha seja pulada.   
  
Para a impressão da agenda completa, deve ser adotada a seguinte formatação:  
\* Todo conjunto de nomes com mesma inicial deve ser precedido por:

"----%c----\n"

Onde %c é a inicial do conjunto de nomes que se segue.  
Apenas devem ser impressas as iniciais de nomes presentes na lista.  
\* A impressão dos contatos segue a mesma formatação descrita acima.

**Observações gerais:**

* É obrigatório o uso de funções recursivas para os casos especificados.
* O tamanho máximo dos nomes armazenados é 40 e o tamanho máximo da agenda é 70.
* Observar atentamente a formatação da entrada/saída.
* Para testar seu programa baixe os testes e siga as instruções de uso do **diff**.
* O nome do arquivo submetido deve ser **agenda2.c**
* **Fazer** o algoritmo **antes** de programar(OBRIGATÓRIO).
* Lembre-se de indentar seu código, e o faça **enquanto** escreve o código.
* Escolha apropriadamente os nomes das variáveis.
* Comente adequadamente seu código (escreva comentários para facilitar a leitura do código - evite comentar o óbvio).
* Escreva sempre um cabeçalho (como comentário) no seu código.
* Não submeta seu código sem testar antes. O número máximo de submissões é 10.

**Avaliação:**

Na correção da atividade serão considerados:

* Corretude dos resultados.
* Indentação.
* Comentários (sobre a lógica utilizada, e não em comandos simples como printf e scanf).
* Algoritmo utilizado (resultados 100% corretos não necessariamente refletem um código bom e correto).
* A ocorrência de Warnings na compilação implicará desconto de nota.