



Lista de Exercício Avaliativa 2

Turma: Sistemas de Informação 3º Período

Objetivo: O presente exercício avaliativo tem por objetivo consolidar os conceitos sobre alocação dinâmica de memória, manipulação de arquivos e tipos definidos pelo programador.

Observações e regras de entrega:

- Esta atividade deverá ser realizada de forma **individual**.
- Terá o valor de 1,5 pontos na composição da nota da A1.
- A solução deverá ser enviada por meio da plataforma run.codes até as 23h59min do dia 07/12/2022.
- Pontos importantes a serem seguidos:
 - **Conformidade com os requisitos** aqui apresentados.
 - Uso dos conceitos de modularização.
 - Códigos bem estruturados e bem indentados.
- Caso seja detectada a ocorrência de plágio, será atribuída nota zero a **todos os envolvidos**.
- Submissões que apresentarem erros de compilação ou falhas de segmentação no run.codes **NÃO** serão corrigidas.
- O não atendimento dos **requisitos especificados neste documento** acarretará em perda de pontos.

Especificação

Gerenciando contatos

Escreva um programa em linguagem C que realize a leitura dos dados de contatos de usuários fictícios. Estes dados estão armazenados no arquivo “`contacts_dataset.dat`” fornecido juntamente com esta especificação. Neste arquivo, os dados estão dispostos linha a linha. Cada linha contém as seguintes informações:

- Id
- Nome
- Telefone
- E-mail

Ao ser iniciado, seu programa deverá ler o **caminho** para o arquivo com um simples `scanf` e posteriormente solicitar o número de contatos (**numCont**) a serem lidos, também através de um simples `scanf`.

Existem exatamente 100 registros no arquivo, portanto o valor lido deverá estar entre:
 $1 \leq \text{numCont} \leq 100$.

Nesta atividade você utilizará um vetor de estruturas (**structs**) que deverá ser alocado dinamicamente de acordo com a quantidade fornecida pelo usuário relacionada ao número de contatos a serem lidos. O vetor deverá ser do tipo (**TContato**) e servirá para armazenar os dados dos respectivos contatos.

Deverá existir uma função para cada uma das seguintes tarefas:

- Alocar o vetor e retornar um ponteiro válido para o mesmo.
- Realizar a leitura dos dados no arquivo e adicionar os contatos no vetor.
- Remover um contato do vetor. Neste caso a função deverá receber por parâmetro o id do contato a ser excluído.
- Mostrar os dados dos contatos **ordenados** por id (identificador).

Para a realizar a ordenação dos contatos no vetor, o algoritmo **quicksort** (também fornecido juntamente com esta especificação) deverá ser utilizado.

Observações importantes:

- Seu programa deverá liberar toda e qualquer memória previamente alocada de forma dinâmica.
- O usuário deverá ser informado com uma mensagem de “Arquivo nao encontrado” se fornecer um nome/caminho de arquivo inexistente. Neste caso o programa deverá ser imediatamente finalizado.

Abaixo seguem alguns exemplos de como seu programa deverá se comportar:

```
./contatos
/home/usuario/Desktop/arquivo.dat
10
Arquivo nao encontrado.
```

```
./contatos
/home/usuario/Desktop/arquivo.dat
0
O valor informado deve estar dentro do intervalo: 1 <= x <= 100.
```

```
./contatos
/home/usuario/Desktop/arquivo.dat
-1
O valor informado deve estar dentro do intervalo: 1 <= x <= 100.
```

```
./contatos
/home/usuario/Desktop/arquivo.dat
101
O valor informado deve estar dentro do intervalo: 1 <= x <= 100.
```

```
./contatos
/home/usuario/Desktop/contacts_dataset.dat
5
```

Id	Nome	Telefone	E-mail
6	Ahmed	(59)92362-7960	placerat@vel.com
24	Abraham	(79)94229-9721	rutrum@morbitristique.net
46	Allen	(85)99961-0159	nibh.enim@sodales.ca
52	Amal	(16)93639-2412	eu@massarutrummagna.net
95	Abraham	(72)97309-3868	auctor@condimentumDonecat.net

Bom Trabalho!