

BCC202 – Estruturas de Dados I (2013-02)

Departamento de Computação - Universidade Federal de Ouro Preto - MG

Professor: **Reinaldo Fortes** (www.decom.ufop.br/reinaldo)

Estagiária docente: **Josiane Rezende**

Monitores: **Bruno H. M. dos Santos**



Aula Prática 11 – Pesquisa Sequencial e Binária

- **Data de entrega: 24/01/2014 até 23:55. O que vale é o horário do Moodle, e não do *seu*, ou do *meu*, relógio!!!**

- **Decréscimo por atraso de até: 12h = 30%, 24h = 40%; 36h = 60%; 48h = 70%; Acima de 48h = 100%.**

- **Procedimento para a entrega:**

1. Crie uma pasta para a questão: **PrimeiroNome-UltimoNome-Questao** (exemplo: reinaldo-fortes-1).
 2. Crie os arquivos de código fonte necessários para a solução da questão (arquivos **.c** e **.h**)
 3. Compile na linha de comando usando **gcc *.c -o prog.exe -lm**.
 4. Execute usando redirecionamento de entrada: **./prog.exe < entrada.txt**.
 5. Apague os arquivos gerados na compilação e o arquivo de entrada (**mantenha apenas os arquivos .c e .h**).
 6. Compacte a pasta criada no item 1.
 7. Faça a entrega do arquivo compactado no moodle, na tarefa destinada à prática e questão correspondentes.
- A cada etapa, verifique se o resultado está conforme o esperado.
 - **Não utilize caracteres acentuados ou especiais para nomes de pastas e arquivos.**

- **Bom trabalho!**

Avaliação Curta

Acesse o Moodle e resolva a avaliação curta 05. **Atenção para acessar a avaliação correspondente ao horário da aula frequentada.** São três opções: **15:20, 17:10 e 19:00.**

Questão 01

Baseado no enunciado do TP03, corresponde aos dois primeiros métodos de pesquisa definidos no TP.

Utilizando-se da TAD `TContato` definida na *Prática 04*, implemente as seguintes TADs de pesquisa:

1. Pesquisa sequencial.
2. Pesquisa binária.

As TADs de pesquisa terão como parte de seus dados um conjunto de contatos que deverá ser representado internamente pela estrutura de dados que melhor convier para cada algoritmo de pesquisa (*array* ou *lista encadeada*, por exemplo).

Cada TAD de pesquisa deverá possuir pelo menos duas operações:

- **Construir:** *contrói* a estrutura interna para armazenamento dos contatos. Recebe como argumento de entrada um vetor de contatos.
- **Pesquisa:** realiza a pesquisa por um determinado elemento utilizando o algoritmo correspondente. Recebe como entrada a chave do elemento procurado e retorna se a pesquisa encontrou o elemento procurado e o número de *comparações feitas*.

0.1 Entrada e Saída

A **entrada** inicia com o número de casos de teste (**C**) na primeira linha. Em seguida, separados por uma linha em branco, a definição de cada caso de teste. Um caso de teste é definido inicialmente pelo número de contatos a serem considerados (**N**) e o número de pesquisas a serem feitas (**P**), ambos na primeira linha de definição do caso de teste. A seguir, serão definidas **N** linhas contendo as informações de contato: nome, aniversário e telefone (conforme padrão adotado na *Prática 04*). Logo após a definição dos contatos, seguirão **P** linhas contendo os números de telefone a serem pesquisados. Não deverão existir chaves repetidas na entrada.

A **saída** consistirá do número de telefone procurado, seguido de duas linhas com o resultado dos métodos de pesquisa, na ordem em que eles foram definidos anteriormente, contendo as seguintes informações: **0**, caso o método tenha encontrado o elemento, **1** caso contrário, seguido pelo número de comparações feitas.

A tabela a seguir apresenta um exemplo do padrão de entrada e saída. Os números de comparações feitas foram definidos ao acaso, portanto, não condizem com a realidade.

Entrada	Saída
2	31-8888-1111
	1 100
5 2	1 101
Fulano1 0128 31-8888-1111	
Fulano2 0225 31-8888-2222	31-8888-6666
Fulano3 0323 31-8888-3333	0 10
Fulano4 0420 31-8888-4444	0 20
Fulano5 0518 31-8888-5555	
31-8888-3333	31-8888-5555
31-8888-6666	1 110
	1 121
4 2	
Fulano4 0420 31-8888-4444	31-8888-8888
Fulano5 0128 31-8888-5555	0 15
Fulano6 0225 31-8888-6666	0 26
Fulano7 0323 31-8888-7777	
31-8888-5555	
31-8888-8888	