

## Aula Prática 08 – Pesquisa Sequencial e Binária

- Data de entrega: 28/02/2016 até 23:55. O que vale é o horário do Moodle, e não do *seu*, ou do *meu*, relógio!!!

- 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!

## Questão 01

Diferencie busca binária e busca sequencial apontando as principais características, vantagens e desvantagens de cada um destes métodos. Aponte situações em que cada um dos métodos é mais adequado.

## Questão 02

Implemente um TAD denominado `TContato` , que possua os campos *nome*, *telefone* e *aniversário*. Em seguida, implemente e 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 armazenados por *array*. Cada TAD de pesquisa deverá possuir as seguintes operações:

- **Construir:** *contrói* a estrutura interna para armazenamento dos contatos, que inicialmente estará vazia.
- **Inserir:** insere um novo elemento no conjunto de contatos.
- **Pesquisar:** realiza a pesquisa por um determinado contato utilizando o algoritmo correspondente. Recebe como entrada o número de telefone do contato e retorna a posição no vetor do contato encontrado e o número de *comparações de chaves* feitas. Caso o contato não seja encontrado o valor de retorno é -1.

### 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 (sem espaços em branco), aniversário (**MMDD** , dois dígitos para o *mês* e outros dois para o *dia*) e telefone (apenas os números). 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 e, nas duas linhas seguintes seguintes os resultados das duas pesquisas realizadas: nome , aniversário , caso o método tenha encontrado o contato, ou **Nao encontrado** caso contrário, a posição no vetor e o número de comparações feitas por cada método de pesquisa.

A tabela a seguir apresenta um exemplo do padrão de entrada e saída. **O resultado para cada algoritmo foi definido ao acaso, portanto, não condiz com a realidade.**

Entrada	Saida
2	3188883333 Fulano3 0323 10 100
5 2	Fulano3 0323 11 101
Fulano1 0128 3188881111	
Fulano2 0225 3188882222	3188886666
Fulano3 0323 3188883333	Nao encontrado -1 150
Fulano4 0420 3188884444	Nao encontrado -1 200
Fulano5 0518 3188885555	
3188883333	3188885555
3188886666	Fulano5 0128 25 110 Fulano5 0128 20 20
4 2	
Fulano4 0420 3188884444	3188888888
Fulano5 0128 3188885555	Nao encontrado -1 153
Fulano6 0225 3188886666	Nao encontrado -1 206
Fulano7 0323 3188887777	
3188885555	
3188888888	