

Prova Final de Estruturas de Dados I (INF092092) – DI UFES – 18/12/2019

Professora Patrícia Dockhorn Costa (pdcosta@inf.ufes.br)

Questão 1) Implemente um TAD Árvore Binária de Busca Genérico – AbbGen (gere os arquivos `abbgen.h` e `abbgen.c`). Considere as seguintes funcionalidades:

- Inserção (sempre como nó folha);
- Impressão (em ordem simétrica – ou seja, sempre crescente por chave de ordenação);
- Liberação (deve liberar toda a memória alocada).

Questão 2) Considere o formato de entrada de dados MATRICULA NOME CR, por exemplo:

```
1234 Patricia 97
8478 Joao 4
432 Maria 3
5400 Jose 2
840 Pedro 8
```

Implemente um cliente do TAD AbbGen (gere o arquivo `prog1.c`) que usa a árvore para organizar dados de um grupo de alunos, ordenados por **Nome** do aluno. Faça os *callbacks* necessários para utilizar as funções do AbbGen! Carregue a árvore com os dados de um arquivo “`entrada.txt`” (exemplo acima) e imprima os dados da árvore em um arquivo “`saida.txt`” - ordenados por **Nome**, no mesmo formato do arquivo de entrada. Por exemplo, a saída esperada para este arquivo de entrada seria:

```
Arvore organizada por Nome:
8478 Joao 4
5400 Jose 2
432 Maria 3
1234 Patricia 97
840 Pedro 8
```

Questão 3) Considere o mesmo formato de arquivo de entrada da questão 2. Implemente outro cliente do TAD AbbGen (gere o arquivo `prog2.c`) que usa a árvore para organizar dados de um grupo de alunos, ordenados por **Matrícula** do aluno. Faça os *callbacks* necessários para utilizar as funções do AbbGen! Carregue a árvore com os dados de um arquivo “`entrada.txt`” e imprima os dados da árvore em um arquivo “`saida2.txt`” - ordenados por **Matrícula**, no mesmo formato do arquivo de entrada. Por exemplo, a saída esperada neste caso seria:

```
Arvore organizada por Matricula:
432 Maria 3
840 Pedro 8
1234 Patricia 97
5400 Jose 2
8478 Joao 4
```

Regras Importantes:

- 1) Use os conceitos de Estruturas Genéricas, como discutido em sala de aula;
- 2) Pode-se utilizar OU não o TAD Aluno fornecido pela professora (faça o que achar mais conveniente para você);
- 3) Libere toda a memória alocada (use o valgrind!);
- 4) Utilize tipos de dados estruturados e ponteiros opacos;
- 5) Veja que estou pedindo dois programas executáveis diferentes! Para fazer as “linkagens”, na linha de comando:
 - gcc abbgen.o aluno.o prog1.o -o prog1.out (gera o executável prog1.out)
 - gcc abbgen.o aluno.o prog2.o -o prog2.out (gera o executável prog2.out)

Para sua informação, aqui está o gabarito de correção desta prova:

1. Questão 1:
 - a. Inserção na ABB Genérica (3,0)
 - b. Impressão da ABB Genérica (1,5)
 - c. Liberação de memória ABB Genérica (1,5)
2. Questão 2: Cliente que organiza os dados por Nome (leitura-escrita de arquivo, uso do TAD genérico e *callbacks*) (2,0)
3. Questão 3: Cliente que organiza os dados por Matrícula (leitura-escrita de arquivo, uso do TAD genérico e *callbacks*) (2,0)

BOA PROVA!!!