INF1007 - PROGRAMAÇÃO II

LISTA DE EXERCÍCIOS 14

1. Implemente um TAD Árvore Binária de Busca (ABB) para armazenar ponteiros para estruturas do tipo Aluno:

```
struct aluno {
  int matricula
  char *nome;
  float media;
};
```

Os nós da árvore binária de busca são definidos pela seguinte estrutura:

```
struct noABB {
  Aluno *info;
  struct noABB *esq;
  struct noABB *dir;
};
```

O TAD ABB deve implementar as seguintes funções:

- `abb_cria` – A função deve criar uma nova ABB vazia retornando NULL. Essa função tem o seguinte protótipo:

```
NoABB *abb_cria();
```

- `abb_insere` – A função deve criar e inserir um novo nó na ABB respeitando a ordenação por média dos nós. Essa função tem o seguinte protótipo:

```
NoABB *abb_insere(NoABB *a, int mat, char *nome, float media);
```

- `abb_imprime` – A função de imprimir na tela todas as informações (matricula, nome e média) de todos os alunos existentes na ABB em ordem crescente de média. Essa função tem o seguinte protótipo:

```
void abb_imprime(NoABB *a);
```

- `abb_libera` – a função deve liberar da memória todos os nós da ABB. Essa função tem o seguinte protótipo:

```
void abb_libera(NoABB *a);
```

Além das funções normais, o TAD ABB também deve implementar e exportar as seguintes funções:

- `abb_alunoComMaiorMedia` – A função recebe uma árvore binária de busca e imprime as informações do aluno com a maior média. A sua função deve levar em consideração a ordenação da árvore binária de busca (isto é, não percorra a árvore inteira, siga apenas o caminho que leva ao nó com a maior média). Essa função tem o seguinte protótipo:

```
void abb_alunoComMaiorMedia(NoABB *a);
```

- `abb_contaAprovados` – A função recebe uma árvore binária de busca e retorna o número de alunos aprovados (isto é, alunos com média maior ou igual a 5.0). A sua função deve levar em consideração a ordenação da árvore binária de busca (isto é, não percorra a árvore inteira se não for necessário). Essa função tem o seguinte protótipo:

```
int abb_contaAprovados(NoABB *a);
```

- `abb_imprimeAprovados` – A função recebe uma árvore binária de busca e imprime a matricula, nome e a média dos alunos aprovados em ordem DECRESCENTE. A sua função deve levar em consideração a ordenação da árvore binária de busca (isto é, não percorra a árvore inteira se não for necessário). Essa função tem o seguinte protótipo:

```
void abb_imprimeAprovados(NoABB *a);
```

- `abb_contaIntervalo` – A função recebe uma árvore binária de busca, um valor mínimo e um valor máximo e retorna o número de alunos com médias no intervalo dado (isto é, alunos com média maior ou igual ao valor mínimo E menor ou igual ao valor máximo). A sua função deve levar em consideração a ordenação da árvore binária de busca (isto é, não percorra a árvore inteira se não for necessário). A função tem o seguinte protótipo:

```
int abb_contaIntervalo(NoABB *a, float min, float max);
```

Após implementar as funções, utilize a seguinte função principal para testar o seu programa:

```c
#include <stdio.h>
#include <stdlib.h>
#include "abb.h"

int main(void) {
  NoABB *abb = abb_cria();

  abb = abb_insere(abb, 456124, "Pedro Duarte", 7.5);
  abb = abb_insere(abb, 453984, "Ana Silva", 8.8);
  abb = abb_insere(abb, 365597, "Joao Filho", 2.5);
  abb = abb_insere(abb, 687451, "Maria Gomes", 10.0);
  abb = abb_insere(abb, 364512, "Silvio Lins", 4.8);
  abb = abb_insere(abb, 984544, "Marcia Morais", 7.8);
  abb = abb_insere(abb, 698421, "Bruno Rodrigues", 2.0);
  abb = abb_insere(abb, 784512, "Thais Silva", 6.5);
  abb = abb_insere(abb, 694231, "Ricardo Costa", 9.5);
  abb = abb_insere(abb, 126411, "Julia Neves", 8.0);

  printf("Alunos em ordem crescente de media:\n");
  abb_imprime(abb);

  printf("\nAluno com maior media:\n");
  abb_alunoComMaiorMedia(abb);

  printf("\nNumero de alunos aprovados: %d\n", abb_contaAprovados(abb));

  printf("\nAlunos aprovados em ordem decrescente: \n");
  abb_imprimeAprovados(abb);

  printf("\nNumero de alunos entre 2.5 e 8.5: %d\n",
  abb_contaIntervalo(abb, 2.5, 8.5));

  return 0;
}
```