

Estruturas de Dados 1  
Trabalho 3 – Árvore Red Black

Enunciado:

Considere que já exista uma estrutura de árvore construída e preenchida com dados, seu trabalho é fazer uma função, que dado o ponteiro para o Nó Raiz da árvore, ela verifica se a árvore em questão é uma árvore Vermelho Preto, para isso sua função deve verificar se a árvore não viola nenhuma das propriedades de árvores Vermelho Preto.

O protótipo da função é:

```
int verifica_RB(RBTree *Arv)
```

O retorno deve ser:

0: Caso a árvore seja válida

Qualquer valor diferente de zero para caso a árvore seja inválida.

A estrutura dos Nós deve ser a seguinte:

```
#define RED 1          // Constantes a serem usadas
#define BLACK 0        // para as cores dos nós

typedef struct _NodeRB{
    int cor;            // Cor do nó (pode ser RED ou BLACK)
    int leaf;           // Diz se o nó é uma folha ou não (1 == Sim, 0 == Não )
    int key;            // Chave do nó
    struct _NodeRB *father; // Ponteiro para o pai do nó
    struct _NodeRB *left;  // Ponteiro para o filho esquerdo do nó
    struct _NodeRB *right; // Ponteiro para o filho direito do nó
}NodeRB;

typedef struct
{
    NodeRB *root;       // Ponteiro para a raiz da árvore
    NodeRB *nil;        // Armazena o Nó nulo especial
}RBTree;
```

## Árvore B

Devem ser desenvolvidas as seguintes funções de árvore B:

- 1) Uma função de Inserção
- 2) Uma função de Remoção
- 3) Uma função de Busca
- 4) Uma função que imprima os nós da árvore em ordem prefixa

Sugestão: Faça algumas inserções e remoções e imprima a árvore resultante depois das modificações. Comentem o código para melhorar a legibilidade e a compreensão do mesmo.

Os trabalhos devem ser realizados por grupos de até quatro alunos.

Devem ser enviados dois arquivos .C, um para cada questão, que possam ser compilados e testados.