

## Exercício 10 - Árvore Binária de Busca

### Descrição

Seja a interface do TAD **ABB** como definido abaixo. Desenvolva a implementação do TAD para a resolução do problema apresentado.

```
#ifndef _ARVORE_BINARIA_DE_BUSCA_H
#define _ARVORE_BINARIA_DE_BUSCA_H

#include "item.h"

typedef struct abb ABB;

ABB *abb_criar (void);
bool abb_inserir (ABB *T, ITEM *item);
void abb_imprimir (ABB *T);
void abb_apagar (ABB **T);
bool abb_remove (ABB *T, int chave);
ITEM *abb_busca (ABB *T, int chave);
bool abb_vazia (ABB *T);
#endif
```

### Entrada

A primeira linha da entrada contém dois inteiros **N** e **M**, indicando a quantidade de elementos que serão inseridos na ABB e a quantidade de operações, a próxima linha contém **N** inteiros para inserir na árvore. Após terão **M** linhas, cada linha contendo dois inteiros, o primeiro inteiro representa o tipo de operação, sendo 1 para busca e 2 para remoção, e o segundo inteiro representando a chave que será utilizada na operação.

### Saída

Imprima “Item encontrado.” ou “Item não encontrado.”, caso seja uma operação de busca, e “Item removido.” ou “Item não removido.”, caso seja uma operação de remoção.

Exemplo de Entrada	Exemplo de Saída
--------------------	------------------

```
5 10
4 5 3 1 10
1 5
2 5
1 5
1 15
1 4
1 1
1 10
2 3
2 4
1 1
```

```
Item encontrado.
Item removido.
Item não encontrado.
Item não encontrado.
Item encontrado.
Item encontrado.
Item encontrado.
Item removido.
Item removido.
Item encontrado.
```

### Observações:

- Somente as bibliotecas *stdio.h*, *stdlib.h* e *stdbool.h* podem ser utilizadas.