

Ficha 9

Programação Imperativa

Árvores binárias

Use, se achar necessário, o projecto <https://codeboard.io/projects/243300> para responder aos problemas propostos.

Considere o seguinte tipo para representar árvores binárias de inteiros.

```
typedef struct nodo {
    int valor;
    struct nodo *esq, *dir;
} * ABin;

ABin newABin (int r, ABin e, ABin d) {
    ABin a = malloc (sizeof(struct nodo));
    if (a!=NULL) {
        a->valor = r; a->esq = e; a->dir = d;
    }
    return a;
}
```

1. Apresente definições das seguintes funções sobre árvores.
 - (a) `int altura (ABin a)` que calcula a altura de uma árvore.
 - (b) `int nFolhas (ABin a)` que calcula o número de folhas (i.e., nodos cujas sub-árvores são **ambas** vazias).
 - (c) `ABin maisEsquerda (ABin a)` que calcula o nodo mais à esquerda de uma árvore.
 - (d) `void imprimeNivel (ABin a, int l)` que escreve no ecrã os elementos da árvore que estão ao nível `l` (Considere que a raiz de uma árvore se encontra no nível 0).
 - (e) `int procuraE (ABin a, int x)` que testa se `x` ocorre na árvore. Não assuma qualquer propriedade sobre a árvore `a`.
2. Apresente definições sobre as seguintes funções sobre árvores binárias **de procura**.
 - (f) `struct nodo *procura (ABin a, int x)` que procura um elemento numa árvore. A função deverá retornar o endereço da célula onde o elemento ocorre ou `NULL` caso o elemento não exista na árvore.
 - (g) `int nivel (ABin a, int x)` que calcula o nível a que o elemento ocorre na árvore (-1 caso não exista).
 - (h) `void imprimeAte (ABin a, int x)` que imprime no ecrã, por ordem crescente, todos os elementos da árvore que são (estritamente) menores do que um dado valor.