Questão Extra. [1 ponto] Suponha que implementamos árvores binárias de busca (ABBs) com elementos do tipo `No`, definido assim:

```
typedef struct no_struct {
  int chave;
  int bal;
  struct no_struct *f_esq;
  struct no_struct *f_dir;
} No;
```

Escreva uma função em `C` de protótipo

```
int calcule_balanco(No *raiz);
```

que se comporta como segue:
(i) esta função recebe um ponteiro para a raiz de uma ABB, cujos nós têm campo `bal` indeterminado;
(ii) esta função deve calcular, para cada nó desta árvore, a diferença entre as alturas das subárvores esquerda e direita deste nó, e deve armazenar este valor no campo `bal` correspondente. Mais precisamente, após a execução da função, se `ptr` aponta para um nó arbitrário desta árvore, deve valer que

```
ptr->bal = altura da ABB de raiz ptr->f_esq
             - altura da ABB de raiz ptr->f_dir.
```

Observação (apenas por curiosidade): existem umas estruturas conhecidas como *árvores AVL* que mantêm este campo `bal` para manter a árvore sempre balanceada. Vocês provavelmente encontrarão estas árvores em MAC323, *Estruturas de Dados*. Se vocês quiserem, podem encontrar uma discussão detalhada sobre estas árvores no vol. 3 de Knuth. Naturalmente, nada disto é necessário para se fazer esta questão!