

## Exercícios – structs

---

Em cada exercício abaixo, implemente e teste sua solução.

### Exercício 1

Suponha que temos um tipo de dado usado para representar retângulos:

```
typedef struct {  
    float altura;  
    float largura;  
} retangulo;
```

Implemente as funções:

```
int eh_quadrado(retangulo r); /* retorna 1 se o retangulo r eh um quadrao e 0 caso contrario */  
float area(retangulo r); /* retorna a area do retangulo */  
int retrato_ou_paisagem(retangulo r); /* Retorna 1 se o retangulo esta na posicao retrato, e 0 se esta na posicao paisagem */
```

### Exercício 2

Um problema que existe com structs é a inicialização de seus membros. Implemente as funções:

```
retangulo novo_retangulo(float nova_altura, float nova_largura); /* Retorna um novo retangulo, com altura e largura definidos nos parametros */
```

```
void modificar_retangulo(retangulo *r, float nova_altura, float nova_largura); /* Modifica o retangulo passado como parâmetro, de forma que ele passe a ter a nova altura e a nova largura recebidos como parametros */
```

### Exercício 3

Estruturas do tipo árvore podem ser usadas para representar hierarquias. As hierarquias podem ser usadas, por exemplo, para contar quantos trabalhadores são subordinados (direta ou indiretamente) a um outro.

- Implemente uma estrutura de dados do tipo árvore em que cada trabalhador é representado por um nó e os trabalhadores diretamente subordinados a ele são os nós filhos a ele.
- Implemente a função `int subordinados()` (escolha o cabeçalho da função) que conta quantos subordinados **diretos** um determinado trabalhador tem.
- Implemente a função `int subordinados_totais()` (escolha o cabeçalho da função) que conta quantos subordinados **diretos e indiretos** (ou seja: contando os subordinados dos subordinados, e assim por diante) um determinado trabalhador tem.
- (desafio): implemente uma função que conta quantos níveis hierárquicos existem na empresa.