

Estruturas Compostas – *Struct*

1. Considere o seguinte programa.

```
/*...*/

struct coordenadas_ponto{
    float x, y;
};
typedef struct coordenadas_ponto ponto;

// subprograma escreve_ponto
// escreve um ponto com as coordenadas x e y na forma (x,y)
/*...*/

// subprograma funcao
void funcao(ponto a, ponto b)
{
    char h[15], v[15];

    if(a.x < b.x)
        strcpy(h,"esquerda");
    else strcpy(h,"direita");

    if(a.y < b.y)
        strcpy(v,"baixo");
    else strcpy(v,"cima");

    printf("O primeiro ponto está à %s e em %s do segundo ponto!\n",h,v);
}

int main()
{
    // declarar pontos p1 e p2;
    /*...*/

    // pedir ao utilizador e ler as coordenadas dos pontos p1 e p2
    /*...*/

    // chamar o subprograma escreve_ponto para mostrar os pontos lidos
    /*...*/

    // chamar o subprograma funcao aplicado aos pontos p1 e p2
    /*...*/

    return 0;
}
```

- 1.1. Complete o programa seguindo as indicações dos comentários:

- Indique o que faz o subprograma *funcao*.
- Crie um subprograma *escreve\_ponto* que apenas escreve as coordenadas de um ponto na forma usual.  
(Exemplo: o ponto com coordenadas 1 e 2 é apresentado como (1,2).)
- Complete o *main* como indicado nos comentários.

1.2. No programa considere uma nova estrutura retângulo:

- a) Defina a estrutura *retangulo* que tem 4 campos do tipo real: *xmin*, *xmax*, *ymin* e *ymax*.
- b) Crie um subprograma *area* que, dado um *retangulo* devolva a sua área.
- c) Crie um subprograma *esta\_dentro* que, dado um *ponto* e um *retangulo*, escreve se o *ponto* está dentro ou fora do *retangulo*. Considere que os pontos do perímetro *estão dentro* do retângulo.

2. Defina uma estrutura que permita representar e efetuar operações com números racionais. Esta estrutura deverá ter dois campos inteiros: um para o numerador e outro para o denominador. Relembre que o denominador não pode ser 0.

Elabore funções que desempenhem as seguintes tarefas:

- a) Ler uma fração;
- b) Somar duas frações;
- c) Subtrair duas frações;
- d) Multiplicar duas frações;
- e) Dividir duas frações;
- f) Calcular o valor real de uma fração;
- g) Determinar a potência (com expoente inteiro) de uma fração.

### Tabelas de *structs*

3. Uma pequena empresa familiar pretende organizar os dados dos seus funcionários: número, nome, tarefa, salário.

3.1. Defina uma estrutura de dados que permita representar cada funcionário.

3.2. Declare uma tabela que permita armazenar informação pretendida.

3.3. Elabore subprogramas para

- a) introduzir os dados de um funcionário na tabela;
- b) listar todos os funcionários e respetivos dados;
- c) listar os funcionários com salário superior a 500€;
- d) procurar e listar todos os dados de um funcionário, usando o seu nome (caso o funcionário não exista, deverá ser devolvida uma “estrutura vazia”: números a zero e *strings* vazias);
- e) atualizar os dados de um funcionário (usando o seu número);
- f) ordene a tabela por ordem crescente dos números dos funcionários;
- g) ordene a tabela por ordem alfabética dos nomes dos funcionários.