

Estrutura de Dados - Turma 23-1

Trabalho 3 - T3

Profs: Denio & Giancarlo

Neste trabalho, o estudante deverá implementar um tipo abstrato de dados para representar um vetor bidimensional e algumas operações sobre o tipo criado. O exemplo abaixo, apresenta uma sugestão de criação da estrutura para o vetor 2d.

```
struct tv2d{
    int p1; // primeira posição
    int p2; // segunda posição
}
typedef struct tv2d v2d;
```

Devem ser implementadas as seguintes operações:

- Atribuir as posições para um vetor. O estudante pode escolher entre a função retornar o tipo com os valores das posições passadas como parâmetro, ou passar a estrutura por endereço.
- Imprimir um vetor passado como parâmetro. O resultado da impressão deverá ser `<valor1, valor2>`, onde `valor1` e `valor2` são os valores das posições do vetor
- Soma entre dois vetores: a função recebe dois tipos vetores e soma as posições e retorna um novo vetor com o resultado da soma: dados os vetores `v1` e `v2`, a soma é `w = <v1.p1+v2.p1, v1.p2+v2.p2>`
- Subtração entre dois vetores: similar a soma, porém as posições são subtraídas. Dados os vetores `v1` e `v2`, a subtração é `w=<v1.p1-v2.p1, v1.p2-v2.p2>`
- Módulo de um vetor: é um inteiro que representa o módulo. Dado um vetor `v`, o módulo de `v` é:
$$\sqrt{v.p1 * v.p1 + v.p2 * v.p2}$$
- Produto interno: um inteiro que representa a multiplicação das posições do vetor. Dado um vetor `v`, o produto interno é `v.p1*v.p1+v.p2*v.p2`

Deverá ser entregue dois arquivos fontes: `vetor.h` e `vetor.c`, esses arquivos deverão ser zipados e o nome do arquivo ZIP deverá ser `matricula.zip`, onde `matricula` deverá ser substituída pela matrícula do estudante.