

Exercícios para desenvolver em Laboratório, em grupo de no máximo 2 alunos.

O programa resultante deverá ser apresentado ao professor no fim da aula.

Nomes: _____ e _____

Tendo as seguintes estruturas:

```
typedef struct ponto {  
    int x;  
    int y;  
} Ponto;
```

```
typedef struct circulo {  
    Ponto p;  
    int raio;  
} Circulo;
```

1. Na função `main()`, crie a variável `p1` do tipo `Ponto` e faça a atribuição de valores quaisquer para seus campos `x` e `y`. Em seguida, peça para o usuário digitar esses valores.
2. Crie uma variável `c1` do tipo `Circulo` e faça a atribuição de valores quaisquer para os seus 3 campos. Em seguida, peça para o usuário digitar esses valores.
3. Crie uma variável '`Circulo *c2`'. Inicialize a variável com uma área alocada dinamicamente, e faça a atribuição de valores quaisquer para os seus 3 campos. Em seguida, peça para o usuário digitar esses valores.
4. Crie uma função `void mostraCirculo(Circulo *c)`, a qual mostra na tela os valores de todos os campos de um círculo.
5. Crie uma função `void atribuiAleatorio(Circulo *c)`, a qual atribui valores aleatórios entre 0 e 99 para os campos do círculo passado como parâmetro. Chame a função na `main`, passando como parâmetro em uma vez a variável `c1` e, em outra, a variável `c2`.
6. Crie uma variável '`Circulo *c3`' na `main`. Inicialize a variável com o endereço de `c1`. Altere os campos usando o ponteiro `c3` e depois chame a função `mostraCirculo` para mostrar os valores tanto de `c1` quanto de `c3`.
7. Crie um vetor na `main`: '`Circulo vet[10]`'; Desenvolva uma função chamada `atribuiAleatorioVetor(Circulo *c, int n)`, a qual atribui valores aleatórios entre 0 e 99 para os campos de cada elemento círculo do vetor. A chamada na `main` será feita com `atribuiAleatorioVetor(vet, 10)`;
8. Crie uma função `void mostraCirculoVetor(Circulo *c, int n)`, a qual mostra na tela os valores de todos os campos de um vetor de círculo. Tente reutilizar a função já desenvolvida no exercício 4.