

LISTA DE EXERCÍCIOS 2

Resolva os exercícios a seguir utilizando a Linguagem C.

1. Crie uma estrutura chamada "Aluno" com os campos "nome", "matrícula" e "nota". Escreva um programa que leia um inteiro n e os dados de n alunos e imprima o nome, matrícula e nota de cada um. Ao final, deve ser calculada uma média a partir das notas lidas e deve ser exibida com duas casas decimais.
2. Crie uma estrutura chamada "Data" com os campos "dia", "mês" e "ano". Escreva um programa que leia duas datas e imprima a data que ocorre primeiro no formato dia/mês/ano.
3. Crie uma estrutura chamada "Ponto" com os campos x e y . Escreva um programa que leia dois pontos e calcule a distância euclidiana entre eles (imprima-a com duas casas decimais).

A distância entre dois pontos pode ser calculada a partir da fórmula:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Utilize a função `sqrt` da biblioteca `math.h` para calcular a raiz quadrada. Dica: pode utilizar a função `pow` da mesma biblioteca para calcular exponenciação.

4. Crie uma estrutura chamada "Retângulo" com os campos *base* e *altura* (do tipo real). Escreva um programa que leia os valores da base e altura de dois retângulos e calcule a área e o perímetro de cada um. A ordem de impressão deve ser *area1*, *perimetro1*, *area2*, *perimetro2*. A área de um retângulo é dada por $base * altura$ e o perímetro é dado por $2 * (base + altura)$.
5. Crie uma estrutura chamada "Livro" com os campos "título", "autor" e "ano". Escreva um programa que leia um inteiro n e os dados de n livros e imprima os livros em ordem alfabética pelo título. Para isso, você pode utilizar uma função de comparação (por exemplo, `strcmp`

da biblioteca `string.h`) para comparar os títulos dos livros e ordená-los com o Bubble Sort.

6. Crie uma estrutura chamada "Fracao" com os campos "numerador" e "denominador". Escreva um programa que receba uma fração e a imprima de forma totalmente simplificada. Para simplificar uma fração, é necessário dividir o numerador e o denominador pelo seu máximo divisor comum. Uma fração está totalmente simplificada quando verificamos que seus termos estão totalmente reduzidos a números que não possuem termos divisíveis entre si."
7. Baseado na estrutura "Fracao" do exercício anterior, escreva um programa que leia os termos de duas frações e calcule a soma entre elas. Em seguida, o programa deve simplificar a fração resultante e imprimi-la no formato "numerador/denominador".
8. Baseado na estrutura chamada "Data" com os campos "dia", "mês" e "ano", crie uma função `compara_data`, que recebe duas datas como parâmetro (`d1` e `d2`) e retorna um inteiro de acordo com a regra a seguir:

- -1, se `d1` vem antes de `d2`;
- 0, se `d1` é o mesmo dia que `d2`;
- 1, se `d1` vem depois de `d2`.

```
int compara_data(Data d1, Data d2) {  
    // retorna -1, 0 ou 1 se a primeira data é menor,  
    // igual ou maior que a segunda data, respectivamente  
    // ...  
}
```

Crie, ainda, uma função `ordena_datas`, que recebe um vetor de elementos do tipo "Data", e um inteiro `n` com o número de elementos. A função deve executar o Bubble Sort e ordenar o vetor em ordem crescente de datas.

```
void bubble_sort_datas(Data datas[], int n) {  
    // ordena o vetor datas de forma crescente  
    // ...  
}
```

Por fim, a `main` deve ler o inteiro `n`, os `n` elementos do tipo `Data`, ordenar o vetor e imprimir as datas em ordem crescente, no formato `dia/mês/ano`.