

Lista de Exercícios de Structs n.2

Parte 1 – Definindo Structs

1. Em uma disciplina onde os alunos possuem **duas notas**, e precisamos registrar e **calcular as médias** de todos eles, como seria um registro para representar esta situação? Implemente o tipo de dado **Aluno** em C.

Novo tipo	Campo	Tipo do campo
Aluno	matricula	Nusérico
	nome	Textual
	nota1	Nusérico
	nota2	Nusérico

2. Implemente um tipo de dado **Pessoa** que será utilizado em um sistema para cadastrar pessoas e em seguida **consultar** o IMC delas.
3. Implementar o tipo de dado **Produto** que será usado em um sistema, para um supermercado, que cadastre produtos e seus preços.

Nome do Registro: Produto

Campos: Para registrar um produto vamos precisar do seu **nome** e o seu **preço**. *Mas como identificar um produto cadastrado?* Quando vamos no supermercado e compramos alguma mercadoria no peso, o caixa do supermercado precisa fornecer um código para cadastrar o produto pesado. Geralmente ele utiliza uma tabela, onde há o nome do produto e o seu código. Para a essa aplicação vamos utilizar este mesmo **código** para identificar unicamente cada produto.

4. Um sistema bancário manipula dois dados principais: cliente e conta. Implemente esses dois tipos de dados em C.

Nomes dos Registros: Cliente e Conta.

Campos: O **nome** do cliente é uma informação relevante. O **CPF** poderá ser utilizado para **diferenciar** clientes com o mesmo nome. *Como identificar a conta do cliente?* Cada conta poderia ter um **número de conta único**, que serviria para identificar a conta do cliente. Cada conta terá um **saldo**, que será gerenciada pelo sistema. Como cada cliente pode possuir mais de uma conta bancária, junto com a conta deveremos registrar qual cliente é o dono dela. Vamos utilizar o **CPF do cliente na conta** para identificar o seu **dono**.

5. Vamos criar um sistema matemático para manipular pontos no plano cartesiano. Defina o tipo de dado **Ponto**. Para registrar um ponto no plano cartesiano basta informar os valores de suas coordenadas (x,y).

Parte 2 – Manipulando *Structs*

6. Complete o código abaixo para:

- (1) Atualizar a matricula do aluno com o valor 201328
- (2) Atualizar o nome do aluno com “Maria Bonita”
- (3) Definir a nota como 8.0
- (4) Definir a segunda nota como 9.0
- (5) Imprimir a matricula e o nome do aluno
- (6) Imprimir os demais dados do aluno.

```
#include <stdio.h>
#include <string.h>

typedef struct {
    int matricula;
    char nome[100];
    float nota1;
    float nota2;
} Aluno;

int main(){
    Aluno aluno;
    aluno.matricula =          //❶
    strcpy           //❷
    aluno.          = 8.0; //❸
    aluno.          =    ; //❹

    printf("\n%d %s %1.2f %1.2f",          , //❺

    return 0;
}
```

7. Complete o código abaixo para:

(1) Ler os valores e atribuí-los aos campos do registro

(2) Emitir a mensagem de “Produto em Promoção” caso o preço seja inferior a 4.
Caso contrário, imprimir “Produto Cadastrado”

```
#include <stdio.h>

typedef struct {
    long  codigo;
    char  nome[100];
    float preco;
} Produto;

int main(){
    Produto p;
    scanf("%ld %s %f",          ); //❶
                                //❷

    return 0;
}
```

8. Complete o código abaixo para:

- (1) Definir uma constante QUANTIDADE_DE_PONTOS com o valor 3 (será utilizada em todo o código para controlar a quantidade de pontos)
- (2) Criar o vetor “pontos” para armazenar 03 pontos (Use a constante!)
- (3) Crie um ponto na posição 0 do vetor com x=-4 e y=7
- (4) Crie um ponto na posição 1 do vetor com x=2 e y=-9
- (5) Crie um ponto na posição 2 do vetor com x=5 e y=3
- (6) Apresente na tela as coordenadas dos pontos que estejam acima do eixo X juntamente com a mensagem “Ponto acima da reta”.

```
#include <stdio.h>
#include <string.h>

typedef struct(
    int x;
    int y;
) Ponto;

# // ❶

int main(){
    Ponto // ❷

    pontos = ; pontos[ // ❸
    pontos = ; pontos[ // ❹
    pontos = ; pontos[ // ❺

    for (int i = 0; i < QUANTIDADE_DE_PONTOS ; i++){ // ❻

    }
    return 0;
}
```

9. Analise o programa que permite cadastrar o nome, a matrícula e duas notas de vários alunos. Ao final, imprime a matrícula, o nome e a média de cada um deles.

```
#include <stdio.h>

typedef struct {
    int matricula;
    char nome[100];
    float nota1;
    float nota2;
} Aluno;

#define QUANTIDADE_DE_ALUNOS 3

int main(){
    Aluno alunos[QUANTIDADE_DE_ALUNOS];

    printf("Dados: nome(sem espacos), matricula, nota1, nota2\n");
    for(int i=0; (i < QUANTIDADE_DE_ALUNOS); i++){
        printf("\nInforme os dados do aluno(%i): ", i+1);
        scanf("%s %i %f %f", alunos[i].nome, &alunos[i].matricula,
            &alunos[i].nota1, &alunos[i].nota2);
    }

    printf("\nMatricula\tNome\tMedia\n");
    for(int i=0; (i < QUANTIDADE_DE_ALUNOS); i++){
        printf("%i\t%s\t%.2f\n", alunos[i].matricula, alunos[i].nome,
            (alunos[i].nota1 + alunos[i].nota2)/2);
    }
    return 0;
}
```

Simulando a execução conforme abaixo, qual será a saída produzida ?

```
Dados do aluno: nome(sem espacos), matricula, nota1, nota2
Informe os dados do aluno(1): Jesuino 2887399 6.0 7.5
Informe os dados do aluno(2): Maria 2887398 7.0 9.0
Informe os dados do aluno(3): Virgulino 2887400 10.0 8.0
```

10. Complemente o código abaixo para que cadastre o nome, a altura, o peso, o cpf e sexo de algumas pessoas. Com os dados cadastrados, em seguida localizar uma pessoas através do seu CPF e imprimir o seu IMC.

(1) Ler o CPF a ser localizado

(2) Se o CPF for igual ao informado ..

(3) Calcular o IMC.

```
#include <stdio.h>

typedef struct{
    int    IMC;
    char  sexo; //
    float peso;
    float altura;
    long long cpf;
} Pessoa;

#define QUANTIDADE_DE_PESSOAS 3

int main() {
    Pessoa pessoas[QUANTIDADE_DE_PESSOAS];
    long long cpf localizador;

    printf("Campos:      altura, peso, cpf, sexo\n");
    for(int i=0; (i < QUANTIDADE_DE_PESSOAS); i++){
        printf("\nInforme os dados da pessoa(%i): ", i+1);
        scanf("%s %f %f %Lu %c", &pessoas[i].altura,
            &pessoas[i].peso, &pessoas[i].cpf, &pessoas[i].sexo);
    }

    printf("\nInforme o CPF da pessoa: ");
    scanf("%Lu", &cpf localizador); // ❶

    printf("\nSexo\tNome\tIMC");
    for(int i=0; (i < QUANTIDADE_DE_PESSOAS); i++){
        if (
            float imc =
        ) ( // ❷
            // ❸

        printf("\n%c\t\t\t%1.2f\n", pessoas[i].sexo, imc);
        break;
    }
    return 0;
}
```