

Linguagem C

Estruturas de Dados
Heterogêneas (struct)

Estruturas (struct)

- “structs” são **Estruturas de Dados Heterogêneas**
- Uma estrutura agrupa várias variáveis numa só.
- Funciona como uma espécie de “ficha”. A ficha armazena diversos dados relacionados, porém de tipos diferente.
- A estrutura, então, serve para agrupar um conjunto de dados não similares, formando um novo tipo de dados.

Estruturas (struct + typedef)

- O comando typedef é usado para criar “sinônimo” ou um “alias” para tipos de dados existentes. Então na prática podemos dizer que estamos renomeando um tipo de dados.
- A renomeação de tipos facilita a organização e o entendimento do código.

Sintaxe:

```
typedef <nome do tipo de dado> <novo nome>;
```

Estruturas (struct)

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

int main()
{

typedef struct{
    char Primeiro[15];
    char Meio[15];
    char Sobrenome[15];
} NomeCompleto;

typedef struct{
    NomeCompleto Nome;
    int idade;
}Pessoa;
```

Estruturas (struct) continua

```
Pessoa Drausio; //Declarando a Var. Drausio como sendo do tipo Pessoa
```

```
printf("Digite seu primeiro nome: ");  
scanf("%s", Drausio.Nome.Primeiro);  
printf("Digite seu nome do meio: ");  
scanf("%s", Drausio.Nome.Meio);  
printf("Digite seu sobrenome: ");  
scanf("%s", Drausio.Nome.Sobrenome);  
printf("Digite sua idade: ");  
scanf("%d", &Drausio.idade);
```

```
printf("Nome completo: %s %s %s\nIdade: %d anos\n", Drausio.Nome.Primeiro,  
      Drausio.Nome.Meio, Drausio.Nome.Sobrenome, Drausio.idade);  
}
```

Estruturas (struct)

- Exercício 1:
 - Criar um programa que permita ao usuário digitar os dados de 3 alunos, (seus nomes, idades e registros acadêmicos). Em seguida, pedir ao usuário para digitar uma idade, e o programa deve imprimir os nomes e idades dos alunos com idade menor que a digitada;

Estruturas (struct)

- Exercício 2:
 - Criar um programa que permita ao usuário digitar os dados de 30 alunos, (seus nomes, idades, registros acadêmicos e). Em seguida, pedir ao usuário para digitar uma idade, e o programa deve imprimir os nomes e idades dos alunos com idade menor que a digitada;
 - (Para isto, será necessário criar um vetor de alunos.....)

Estruturas (struct)

- Exercício 3:
 - Criar um programa que permita ao usuário digitar os dados de 30 alunos, (seus nomes, idades, registros acadêmicos e). Em seguida, pedir ao usuário para escrever os 30 nomes e idades em ordem crescente de idade.

Estruturas (struct)

- Exercício 4:
 - Criar um programa em C que leia um nome (string) para um retângulo e 4 pontos num plano cartesiano que indiquem as posições x_1, y_1, x_2 e y_2 (inteiros) deste retângulo.

Após a leitura dos dados para o primeiro retângulo, leia novos dados para um segundo retângulo.
 - Terminado o processo de leitura, indique se existe algum tipo de sobreposição nestes retângulos, a resposta deve ser:
 - Não ocorre sobreposição
 - Sobreposição apenas sobre as bordas
 - Sobreposição de área interna

Estruturas (struct)

- Exercício 5:
 - Criar um programa em C que armazene os dados de todos os alunos da turma (5 alunos), e permita os cálculos de:
 - Idade de todos os alunos(sendo dada uma data atual);
 - Médias de todos os alunos;
 - Listagem dos alunos que ficaram para final(média entre 4 e 7);

```
typedef struct {  
    int dia, mes, ano;  
} tipo_data;
```

```
typedef struct {  
    char nome[50];  tipo_data nascimento; float notas[3];  
} tipo_aluno;
```

Estruturas (struct)

- Exercício 6:
 - Criar um programa em C que permita:
 - Cadastrar até 3 turmas, com nome e código;
 - Cadastrar até 15 alunos, não podendo ultrapassar 5 alunos por turma, o aluno deve indicar o código da turma que deseja participar;
 - Imprimir o nome da turma e dos alunos de cada turma.

```
typedef struct {  
    char nome[50];  
} tipo_aluno;
```

```
typedef struct {  
    int codigo;  
    char nome[50];  
    tipo_aluno alunos[5];  
} tipo_turma;
```