

DISCIPLINA: ESTRUTURAS DE DADOS

PROGRAMA DE ARTICULAÇÃO DA FORMAÇÃO PROFISSIONAL MÉDIA E SUPERIOR (AMS)

STRUCTS

- Prof. Alexandre Ponce de Oliveira
alexandreponceo@gmail.com
Aulas 11 e 12
Lins/SP

Structs (Estruturas ou Registros)

Vetores e matrizes possuem estruturas de dados homogêneas que armazenam vários valores, mas todos de um mesmo tipo.

É comum conjunto de dados de tipos diferentes. Exemplo: ficha de cadastro do aluno.

Ficha do Aluno

Nome: string

Endereço: string

Telefone: string

RA: int

Mensalidade: float

Structs (Estruturas ou Registros)

Struct é um conjunto de variáveis de tipos distintos (ou não), agrupadas sob um único nome. As variáveis que compõem a estrutura são chamadas membros, campos ou elementos.

Exemplos:

pessoa -> nome, idade, endereço.

data -> dia, mes, ano.

Structs (Estruturas ou Registros)

Sintaxe (Linguagem C)

A palavra reservada `struct` indica ao compilador que está sendo criada uma estrutura.

Uma estrutura deve ser declarada após incluir as bibliotecas e antes da função `main`.

```
struct <identificador> {  
    tipo <membro1>;  
    tipo <membro2>;  
    ...  
    tipo <membroN>;  
};
```

Structs (Estruturas ou Registros)

No momento da declaração da estrutura não há alocação de espaço de memória. É definido um modelo de como estruturas do tipo identificador, devem ser construídas.

Existem duas formas para criar variáveis dos tipos de estrutura:

```
struct data{  
int dia;  
int mes;  
int ano;  
} amanha, feriado;
```

ou

```
int main () {  
...  
struct data hoje;  
... }
```

Structs (Estruturas ou Registros)

Para atribuir um valor para os campos da struct é utilizado o nome da variável struct, seguido de ponto, e o nome do campo.

Exemplos:

```
hoje.dia = 11;
```

```
hoje.mes = 03;
```

```
hoje.ano = 2025;
```

ou de uma forma direta:

```
struct data hoje ={11,03,2025};
```

Para obter o dia através da leitura do teclado pode-se utilizar:

```
scanf("%d", &amanha.dia);
```

Para imprimir o mês:

```
printf("%d", amanhã.mes);
```

Structs (Estruturas ou Registros)

Estruturas com estruturas

```
struct data{  
    int dia;  
    int mes;  
    int ano;  
}dtnasc;
```

```
struct ficha{  
    char nome[30];  
    int matricula;  
}aluno;
```

Nos exemplos acima foram criadas as estruturas **data** e **ficha**, e declarada as variáveis **dtnasc** e **aluno**, respectivamente para as duas estruturas.

Structs (Estruturas ou Registros)

Caso seja necessário a data de nascimento do aluno, é possível adicionar a struct data como campo da struct ficha da seguinte forma:

```
struct ficha{  
    char nome[80];  
    int matricula;  
    struct data dtnasc;  
}aluno;
```

A seguir, exemplos para atribuição de valores para os campos da struct **ficha**:

```
gets(aluno.nome);  
scanf("%d", &aluno.matricula);  
scanf("%d", &aluno.dtnasc.dia);
```


Structs (Estruturas ou Registros)

Vetores com Struct

Para declarar um vetor de struct é necessário antes definir a struct:

```
struct ficha{  
    char nome[80];  
    int matricula;  
    struct data dtnasc;  
};
```

Em seguida, declara-se o vetor do tipo da struct criada:

```
struct ficha alunos[20];
```

Structs (Estruturas ou Registros)

Vetores com Struct

Para manipular os dados do vetor, devem ser fornecidos o índice e o campo, por exemplo:

```
strcpy(alunos[0].nome, "Alexandre");
```

```
alunos[2].matricula = 123456;
```

```
scanf("%d", &alunos[5].dtnasc.dia);
```

Structs (Estruturas ou Registros)

Exercícios

- 1) Criar uma struct chamada notas, que armazena o nome, a nota1, a nota2 e a média do aluno. Criar uma variável dessa struct e ler os campos (com exceção da média que é calculada). Depois exibir o nome, a média do aluno e a palavra aprovado (se média ≥ 6) ou reprovado.
- 2) Considerando o exercício 1, criar uma variável que é um vetor da struct notas. O programa deve calcular e mostrar a média de 10 alunos e também a média geral da sala.