


ESTRUTURA DE DADOS HETEROGÊNEA	ESTRUTURA DE DADOS HETEROGÊNEA
Qual a diferença entre as estruturas de dados Homogêneas para as Heterogêneas ?	<p>Estruturas de Dados Homogêneas são estruturas de dados onde é possível armazenar variáveis simples ou vetores, em uma estrutura de dados que permite uma coleção de dados do mesmo tipo.</p> <p>Já Estruturas de Dados Heterogêneas são estruturas onde conseguimos armazenar uma coleção de dados de tipos diferentes.</p>
Como as variáveis que contém o tipo de Estrutura de Dados Heterogênea são conhecidas ?	Nas Linguagens de programação esse tipo de variável que contém estrutura de dados Heterogênea são chamadas de "Estrutura" (Struct) ou "Registro" .
Como podemos definir uma Estrutura ou Registro ?	<p>Estruturas ou Registros: são uma coleção arbitrária de variáveis logicamente relacionadas agrupadas sob um único nome, de forma a facilitar a sua referência. As estruturas podem conter variáveis de qualquer tipo de dados. As variáveis que fazem parte de uma estrutura são denominadas membros ou campos que são identificadas por nomes, dessa forma:</p> 
- reservado para a questão acima -	
Como podemos declarar uma variável do tipo Estrutura na Linguagem C?	<p>Definimos uma estrutura pelo seu nome e pelo conjunto de campos que ela contém. A forma geral de definição de uma estrutura em linguagem C é:</p> <pre>struct Data { int dia; int mes; int ano; };</pre>
- reservado para a questão acima -	<p>No exemplo ilustrado acima, definimos uma estrutura denominada Data, que contém os campos dia, mes, ano, todos eles do tipo inteiro. As estruturas devem ser definidas antes de qualquer função, isso é de maneira global. Isso permite que as funções tenham acesso a estrutura.</p>
Como podemos Declarar uma variável sobre os campos de Estrutura na linguagem C?	<p>Na seção anterior, vimos como definir uma estrutura. Para a manipulação dos dados em uma estrutura é preciso declarar uma variável do tipo da estrutura. A forma geral de declaração de variáveis do tipo estrutura em linguagem C é:</p> <pre>struct <nome_estrutura> <nome_variável>;</pre> <p>Por exemplo, vamos declarar uma variável para a estrutura criada da seção anterior:</p> <pre>struct Data hoje;</pre>
Como podemos referenciar aos campos por através da variável declarada na Estrutura na linguagem C?	<p>Uma vez declarada uma variável do tipo estrutura, podemos atribuir dados a seus campos. A forma geral na linguagem C é:</p> <pre><nome_variável>.<nome_campo> = valor;</pre> <p>Por exemplo, podemos atribuir valores aos campos da variável do tipo estrutura declarada na seção anterior:</p> <pre>hoje.dia = 20; hoje.mes = 10; hoje.ano = 2013;</pre>
Como podemos ler dados da tela e atribuí-los a uma Estrutura na Linguagem C?	<p>Para isso vamos usar a função scanf(). Como em um vetor, não podemos acessar todos os dados ao mesmo tempo. Em uma estrutura, temos que acessar um campo de cada vez. O trecho de código a seguir solicita ao usuário para informar dados que preenchem a variável hoje:</p> <pre>// declaração de uma variável do tipo Data struct Data hoje; // leitura do campo dia printf("Dia: "); scanf("%d", &hoje.dia); // leitura do campo mês printf("Mes: "); scanf("%d", &hoje.mes); // leitura do campo ano printf("Ano: "); scanf("%d", &hoje.ano);</pre>
- reservado para a questão acima -	

ESTRUTURA DE DADOS HETEROGÊNEA		ESTRUTURA DE DADOS HETEROGÊNEA
Como podemos acessar dos dados de uma Estrutura e apresentá-los na tela usando a Linguagem C?		Uma vez preenchida uma estrutura, queremos manipular os seus elementos ou apresentá-los na tela. Utilizando a variável hoje preenchida na seção anterior, vamos apresentar seus dados na tela usando a função printf(). Analise o seguinte trecho de código.
- reservado para a questão acima -		<pre>// imprime o valor do campo dia printf ("Dia = %d\n", hoje.dia); // imprime o valor do campo mes printf ("Mês = %d\n", hoje.mes); // imprime o valor do campo ano printf ("Ano = %d\n", hoje.ano);</pre>
Como podemos inicializar uma Estrutura no momento da sua declaração?		Para isso, basta fornecer os valores iniciais de seus campos entre chaves e separados por vírgula. Os valores devem seguir a ordem em que os campos foram declarados na estrutura. O exemplo a seguir apresenta a inicialização da estrutura Data.
- reservado para a questão acima -		<pre>// definição da estrutura struct Data { int dia, mes, ano; }; ... // declaração e inicialização da variável hoje struct Data hoje = {21, 10, 2013};</pre>
O que são Estruturas Aninhadas ?		Os valores são atribuídos aos campos da estrutura hoje na ordem em que são fornecidos: o campo dia recebe o valor 21, o mes, 10 e o ano, 2013.
- reservado para a questão acima -		É quando nós possuímos uma Estrutura que possui uma Estrutura como valor de um dos seus campos, ou seja, temos uma Estrutura dentro de outra Estrutura. Isso é possível desde que os tipos de tais estruturas tenham sido previamente declarados no programa. A seguir, apresentamos um exemplo deste tipo de estrutura.
Ilustre graficamente como se comporta uma Estrutura Aninhada...		<pre>struct Data { int dia, mes, int ano; }; struct Pessoa { char nome[50]; struct Data dataNasc; };</pre> <p>Note que o campo dataNasc é uma estrutura aninhada. Para isso, é necessário que a estrutura Data tenha sido definida antes da estrutura Pessoa. A próxima figura apresenta a representação gráfica da estrutura Pessoa.</p>
Como podemos atribuir dados e acessar os valores de uma Estrutura Aninhada ?		Ao acessar campos em estruturas aninhadas, podemos usar o operador de seleção de campo (.) quantas vezes forem necessárias. A seguir, apresentamos um exemplo de acesso aos campos de uma variável do tipo da estrutura Pessoa...
- reservado para a questão acima -		<pre>// declaração da variável da estrutura Pessoa struct Pessoa umaPessoa; // atribuição de um nome strcpy (umaPessoa.nome, "Joao"); // acesso aos campos da estrutura aninhada umaPessoa.dataNasc.dia = 15; umaPessoa.dataNasc.mes = 5; umaPessoa.dataNasc.ano = 1976;</pre>
Como podemos iniciar uma Estrutura Aninhada no momento da declaração dela?		Podemos também declarar e inicializar uma estrutura aninhada. Veja o exemplo que usa a estrutura aninhada definida anteriormente:
		<pre>// declaração e inicialização da variável da estrutura Pessoa struct Pessoa umaPessoa = {"Joao", {15, 5, 1976}};</pre>

[illegible]