

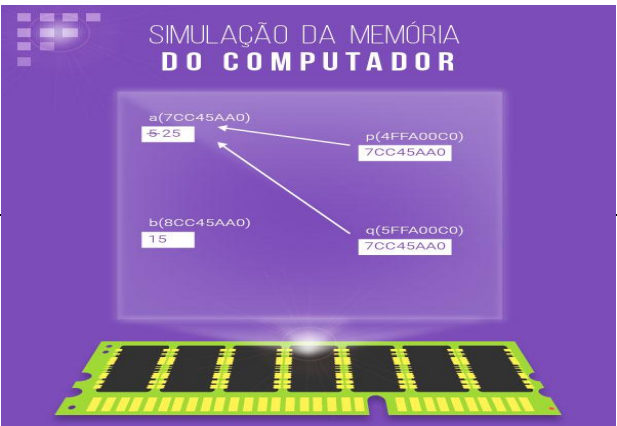


PONTEIROS	PONTEIROS
O que é a estrutura de dados Ponteiro ?	<p>Ponteiro: Na maioria das linguagens de programação, as variáveis, quando declaradas, devem ter um identificador (nome) e um tipo de dados. Quando o programa é executado, as variáveis declaradas recebem uma identificação para que elas possam ser localizadas na memória do computador e uma quantidade de bytes, esse identificador normalmente é representado por números hexadecimais e ocupam o primeiro byte alocado para elas, damos a esse identificador o nome de ponteiro.</p>
<p>Ilustre o uso de ponteiros: (Perceba que nos ponteiro eles não recebem o nome da variável e sim a própria localização da memória em hexadecimal)</p>	
- reservado para a questão acima -	
Para que finalidades geralmente utilizamos os ponteiros?	<p>Os ponteiros podem ser utilizados para diversas finalidades. Dentre elas, cita-se:</p> <p>Subprogramas (funções e procedimentos): modificação de argumentos em subrotinas, quando os parâmetros são passados por referência;</p> <p>Alocação dinâmica de memória: permitem a criação de estruturas de dados complexas, tais como listas encadeadas e árvores binárias;</p>
Como podemos definir um Ponteiro?	<p>Um ponteiro é uma variável especial que armazena endereço de memória ao invés de armazenar um dado ou valor. O mecanismo usado para isso é o endereço da variável, sendo o ponteiro a representação simbólica de um endereço. Com isso, é possível acessar o conteúdo de uma variável de forma indireta.</p>
Como podemos fazer a declaração de uma variável como Ponteiro?	<p>A única diferença na declaração de ponteiros com relação às variáveis mais comuns (int, float e char) está no fato dele armazenar endereço de memória. Para informar que a variável é um ponteiro, basta colocar o símbolo de asterisco (*) ao lado do tipo da variável. Por exemplo, se o tipo é definido como int, então a variável declarada como ponteiro só poderá armazenar endereço de memória para um número inteiro. O símbolo de asterisco é que vai indicar à linguagem C que a variável é um ponteiro e não uma variável comum.</p> <p>A forma geral de declaração de um ponteiro em linguagem C é:</p>
- reservado para a questão acima -	<p>tipo *nome-variável;</p> <p>Onde, tipo é um tipo qualquer de dados e nome-variável é o nome pelo qual o ponteiro será referenciado. Vejamos 2 exemplos:</p> <p>1º a declaração de um ponteiro denominado px, que armazenará o endereço de memória de uma variável do tipo int:</p>
- reservado para a questão acima -	<p>int *px;</p> <p>2º a declaração de um ponteiro denominado pc, que armazenará o endereço de memória de uma variável do tipo char:</p>
	<p>char *pc;</p>
O que são os Operadores de Ponteiro ? E Como podemos utilizá-los?	<p>A manipulação de variáveis declaradas como ponteiros se faz por meio da utilização de 2 operadores unários, que são os seguintes:</p> <p>* : o operador “asterisco” pode ser utilizado sob 2 formas:</p> <ul style="list-style-type: none"> - Na declaração de ponteiros, é ele que é usado para declarar uma variável como ponteiro; - E para visualizar e também modificar o conteúdo de uma variável que está sendo apontada pelo ponteiro, do contrário, se tentarmos visualizar a variável ou mudar o valor dela por métodos normais, vamos ver somente o endereço de memória, e não o dado em si;
- reservado para a questão acima -	<p>& : este operador permite acessar (referencia) o endereço de memória de uma variável, por exemplo, podemos fazer uma variável ser atribuída sobre o endereço de memória de uma variável ponteiro por através desse operador. Mas, para isso a variável ponteiro deve ser declarada antes do nome da variável que deseja imitar o seu endereço de memória;</p>

PONTEIROS	PONTEIROS
Ilustre o uso dos Operadores de Ponteiro...	<p>Temos logo abaixo é mostrado um programa que possui uma variável inteira "x" e um ponteiro "px", que contém o endereço de memória de "x".</p> <pre>#include <stdio.h> #include <stdlib.h> main(){ int x = 15; int *px; px = &x; printf("Endereco de memoria de x = %p", &x); printf("Conteudo da variavel x por meio do ponteiro px = %d", *px); printf("Conteudo da variavel px = %p", px); printf("Endereco de memoria de px = %p", &px); system ("PAUSE"); }</pre> <p>Note que: Na linha 6, a variável px é declarada como um ponteiro, utilizando-se o operador * (asterisco). Na linha 7, a variável px recebe o endereço de memória de x, utilizando-se o operador &. Percebe-se que, na linha 8, para mostrar o endereço de memória de x, é necessário o uso do operador & e o especificador de formato %p. Na linha 9, para exibir o conteúdo da variável que está sendo apontada por px (acesso indireto ao conteúdo de x), é necessário o uso do operador * (asterisco).</p> <p>A Figura abaixo, mostra uma simulação do programa acima...</p>
- reservado para a questão acima -	
- reservado para a questão acima -	
- reservado para a questão acima -	 <p>A simulação mostra uma memória virtual com duas variáveis: x(3FFA00C0) com valor 15 e px(4FFA00C0) com valor 3FFA00C0. Uma seta indica que o ponteiro px aponta para a variável x. Abaixo, há uma representação de uma placa de memória física.</p>
- reservado para a questão acima -	
Ilustre como podemos usar os operadores de ponteiro para modificar o valor de uma variável por através dos ponteiros...	<p>Podemos ver isso no exemplo de um programa com as variáveis "p" e "q" como ponteiros para número inteiro. Observe que, na linha 10, é feita uma atribuição à variável q, que passa a ter o mesmo valor (endereço de memória de a) armazenado em p. Na linha 11, ao modificarmos o conteúdo apontado por q, o valor armazenado em a será alterado, visto que a variável q também aponta para a variável a, mudando o valor dela para "25"...</p> <pre>#include <stdio.h> #include <stdlib.h> main(){ int a, b; int *p, *q; a = 5; b = 15; p = &a; q = p; *q = 25; printf("Conteudo da variavel a = %d", a); system ("PAUSE"); }</pre> <p>Veja o resultado dessa operação na prática...</p>
- reservado para a questão acima -	
- reservado para a questão acima -	 <p>A simulação mostra uma memória virtual com quatro variáveis: a(7CC45AA0) com valor 5, b(8CC45AA0) com valor 15, p(4FFA00C0) com valor 7CC45AA0 e q(5FFA00C0) com valor 7CC45AA0. Setas indicam que tanto p quanto q apontam para a variável a. Abaixo, há uma representação de uma placa de memória física.</p>
- reservado para a questão acima -	