



Definição: Um **ponteiro** é uma constante ou variável que **contém um endereço** que pode ser usado para acessar dados.

Para declarar um ponteiro, utilize a notação sintática de compilação (*), para indicar que aquela variável ou constante é um ponteiro.

```
char a;      // define uma variável (a) do tipo caractere
int i;       // define uma variável (i) do tipo inteiro
float f;     // define uma variável (f) do tipo número com ponto flutuante

char* pa;    // define um ponteiro (pa) para um caractere
int* pi;     // define um ponteiro (pi) para um inteiro
float* pf;   // define um ponteiro (pf) para um número de ponto flutuante
```

Operador de endereço (&) - extrai o endereço de uma variável ou constante.

```
&a          // extrai o endereço de memória onde a variável a está armazenada

pa = &a;    // atribui o endereço de memória da variável a para o ponteiro pa.
```

Operador de indireção (*) - utilizado para acessar o conteúdo apontado pelo ponteiro. Permite inspecionar e/ou alterar o conteúdo armazenado.

```
*pi        // acessa o conteúdo da variável apontada pelo ponteiro pi.

pi = &i;
*pi = 10;   // altera o conteúdo armazenado na variável i para 10.
```

Código de Conversão (%p) - utilizado para imprimir os endereços de memória armazenados nos ponteiros.

```
printf("%p", &f) // imprime o endereço de memória onde a variável f está
                  // armazenada.

printf("%p", pf) // imprime o endereço de memória armazenado no ponteiro pf, ou
                  // seja, o endereço onde a variável f está armazenada na memória.
```

Exercícios em linguagem C:

- 1) Escreva um programa que possua três variáveis (dos tipos **char**, **int** e **float**). Solicite ao usuário para digitar valores para estas três variáveis e imprima os endereços de memória onde estas variáveis estão armazenadas. Execute várias vezes este programa. O resultado exibido na tela é igual em todas as execuções? Por que?

- 2) Insira três ponteiros ao código elaborado no exercício 1 (um ponteiro para **char**, um ponteiro para **int** e um ponteiro para **float**). Associe cada uma das variáveis aos respectivos ponteiros. Altere seu código para receber os valores digitados pelo usuário através dos ponteiros. Após receber os valores, imprima o conteúdo das variáveis na tela.
- 3) Faça um programa que possua três variáveis do tipo **char** (a, b, c) e um ponteiro para **char** (pc). Solicite ao usuário para informar os valores para as três variáveis do tipo character. Imprima o conteúdo de cada uma das variáveis utilizando apenas o ponteiro pc.
- 4) Faça um programa que receba dois números inteiros (a, b) e apresente a soma destes valores (res). Associe os inteiros (a,b, res) aos ponteiros (pa, pb, pres). Faça a entrada dos dois números e o cálculo da soma (que deve ser armazenado em res) usando apenas os ponteiros (pa, pb, pres). Apresente o resultado da soma armazenado em res.
- 5) Faça um programa que possua um vetor (vi) com 10 números inteiros. Gere valores aleatórios para preencher o vetor vi. Na sequência, defina um vetor de ponteiros para números inteiros (vponteiro), também com 10 posições, e armazene o endereço de cada um dos elementos de vi neste vetor de ponteiros vponteiro. Faça um laço para apresentar os endereços de memória de cada um dos 10 valores e, também, o conteúdo de cada posição dos vetores.