

Fonte: Núcleo de Ensino à Distância da UFMG

<http://www.ead.cpdee.ufmg.br/>

Exercícios sobre ponteiros

1. Explique a diferença entre:

`p++`; `(*p)++`; `*(p++)`;

-O que quer dizer `*(p+10)`;

-Explique o que você entendeu da comparação entre ponteiros.

2. Qual o valor de y no final do programa? Tente primeiro descobrir e depois verifique no computador o resultado. A seguir, escreva um /* comentário */ em cada comando de atribuição explicando o que ele faz e o valor da variável à esquerda do '=' após sua execução.

```
int main()
{
    int y, *p, x;
    y = 0;
    p = &y;
    x = *p;
    x = 4;
    (*p)++;
    x--;
    (*p) += x;
    printf("y = %d\n", y);
    return(0);
}
```

3. Verifique o programa abaixo. Encontre o seu erro e corrija-o para que escreva o numero 10 na tela.

```
#include <stdio.h>
int main()
{
    int x, *p, **q;
    p = &x;
    q = &p;
    x = 10;
    printf("\n%d\n", &q);
    return(0);
}
```

4. Escreva um programa que declare uma matriz 100x100 de inteiros. Você deve inicializar a matriz com zeros usando ponteiros. Preencha depois a matriz com os números de 1 a 10.000 usando ponteiros.

5. Sabendo-se que valor de uma variável ou expressão do tipo vetor é o endereço do elemento zero do vetor. Seja `a[]` um vetor qualquer, independente de tipo e tamanho, e `pa` um ponteiro para o mesmo tipo de `a[]`. Responda V ou F, justificando:

☐ Após a atribuição `pa=&a[0]`; `pa` e `a` possuem valores idênticos, isto é, apontam para o mesmo endereço.

☐ A atribuição `pa=&a[0]`; pode ser escrita como `pa=a`;

☐ `a[i]` pode ser escrito como `*(a+i)`

- ☐ `&a[i]` e `a+i` são idênticos
- ☐ `a+i` e' o endereço do i-ésimo elemento do vetor `a`
- ☐ `pa[i]` e' idêntico a `*(pa+i)`
- ☐ `pa=a` e' uma operação válida
- ☐ `pa++` e' uma operação válida
- ☐ `a=pa` e' uma operação válida
- ☐ `a++` e' uma operação válida

6. O que está errado com os programas abaixo? Descubra e indique a solução para consertá-los. Execute-o no computador para ver se o erro foi resolvido.

a)

```
void main() /* esse programa esta errado */
{
int x, *p;
x = 10;
*p = x;
}
```

Obs: por se tratar de um programa muito pequeno, talvez a execução deste não mostrará de imediato o erro. Mas ele existe e é sério.

b)

```
void main() /* esse programa esta errado */
{
int x, *p;
x = 10;
p = x;
printf ("%d", *p);
}
```