

Lista de exercícios sobre ponteiros

- 1) O que é e para que serve um ponteiro?
- 2) Qual a sintaxe correta para declarar um ponteiro?
- 3) Diferencie os operadores de ponteiros & e * em C.
- 4) Assumindo que o endereço da variável i foi atribuída ao ponteiro p, qual a forma correta de referenciar i através do ponteiro?
- 5) Explique qual a diferença entre um ponteiro do tipo float, int e char.
- 6) Supondo que c seja um ponteiro para char e i um ponteiro para int, explique a diferença entre realizar as operações i++ e c++.
- 7) Supondo o código abaixo:

```
int *p;  
int num;  
p = &num;
```

Diga se cada uma das comparações é verdadeira ou falsa, e justifique sua resposta:

- a) num == &p;
 - b) num == *p;
 - c) p == *num;
 - d) p == #
- 8) O trecho de código abaixo falta uma linha para funcionar. Insira essa linha no código e justifique o porquê dela ser necessária:

```
int num, *pont;  
*pont = 10;  
printf("o número é %d", num);
```

- 9) Escreva um trecho de código que divida o conteúdo de uma variável X por 2 sem utilizar o X na expressão.
- 10) Supondo que p seja um ponteiro para float, explique a diferença entre os comandos p++, *p++, (*p)++, e *(p++);

- 11) Explique qual a relação que um vetor tem com um ponteiro.
- 12) Supondo que vet seja um vetor, e p seja um ponteiro do mesmo tipo do vetor. Qual a diferença entre se fazer a atribuição `p = vet` e `p = &vet[0]`? Explique sua resposta.
- 13) Supondo que um ponteiro p esteja apontando para um vetor do tipo int. Logo após é apresentada a seguinte linha:

```
printf("%d", *(p+7));
```

O que aparecerá na tela? Por quê?

- 14) Supondo um vetor do tipo char chamado vet. A quinta posição de um vetor pode ser acessada pelo comando `*(vet+5)`? Caso contrário justifique e sugira uma solução nos mesmos moldes. E se o vetor fosse do tipo int, o que mudaria? Justifique.
- 15) Explique detalhadamente o que o trecho de código abaixo faz:

```
int vet[10], *pv, i=0;
pv = vet;
while (pv < &vet[10]){
    i++;
    *pv = i;
    pv++;
}
```

- 16) O que a função `sizeof()` faz? O que irá retornar caso pedíssemos para mostrar na tela o valor de retorno de `sizeof(int)`? e de `sizeof(char)`? Explique.
- 17) Usando a função `sizeof()`, como podemos saber qual o tamanho de um vetor qualquer, sabendo somente seu tipo e nome? Escreva a linha de código que retorna o tamanho deste vetor.
- 18) O código abaixo contém um erro. Explique-o e faça a correção:

```
int *p;
int i;
p = &i;
scanf("%d", *p);
```

- 19) Como é feita a declaração de um ponteiro para ponteiro e para que eles servem? Dê um exemplo de uso.
- 20) Analise os dois códigos abaixo:

```
int x=10, *p, *q;
p = &x;
q = p;
```

```
int x=10, *p, **pp;
```

```
p = &x;  
pp = &p;
```

Explique cada um deles, evidenciando as diferenças.

21) Supondo o código abaixo:

```
int x=10, *p, **pp;  
p = &x;  
pp = &p;  
printf("%d", *pp);
```

O que aparecerá na tela? Justifique sua resposta.

22) Observe o código abaixo:

```
int *p, *q, **r;  
int x=10, y=100;  
p = &x;  
r = &p;  
q = &y;  
*r = q;  
*p = 25;  
*q++;
```

Qual será o conteúdo das variáveis x e y após o trecho de código ter sido executado? Justifique sua resposta explicando o código detalhadamente.

23) Supondo que exista uma variável chamada var, o que apareceria na tela caso mandássemos mostrar *&var? Explique o porquê.