

Lista 7

1. O que é um ponteiro? E para que serve um ponteiro?

Um ponteiro contém um endereço de memória de uma variável, com isso podemos atribuir um valor à variável guardada nesse endereço, ou seja, à variável apontada pelo ponteiro. Para isso, usamos o operador * (asterisco), que basicamente significa "o valor apontado por". Ex: `int a; int *p = &a; *p = 20;`

2. Quais das seguintes instruções são corretas para declarar um ponteiro?

a) `int _ptr x;`

c) `*int ptr;`

~~b) `int *ptr;`~~

d) `*x;`

3. Qual é a maneira correta de referenciar **ch**, assumindo que o endereço de **ch** foi atribuído ao ponteiro **indica**?

~~a) `*indica;`~~

d) `ch`

b) `int *indica;`

e) `*ch;`

c) `*indic;`

4. Na expressão `float *pont;` o que é do tipo float?

a) a variável `pont`.

~~c) a variável apontada por `pont`.~~

b) o endereço de `pont`.

d) nenhuma das anteriores.

5. Assumindo que o endereço de **num** foi atribuído a um ponteiro **pnum**, quais das seguintes expressões são verdadeiras?

a) `num == &pnum`

c) `pnum == *num`

~~b) `num == *pnum`~~

~~d) `pnum == &num`~~

6. Assumindo que queremos ler o valor de **x**, e o endereço de **x** foi atribuído a **px**, a instrução seguinte é correta? Por que?

`scanf ("%d", *px);`

Não, pois o elemento `*px` está referenciando o valor de `x`, e para ler uma variável utilizando o `scanf` devemos fornecer o endereço de memória da variável que queremos ler, desta forma em vez de `*px` deveríamos ter `(px)` ou então `(&x)`.

7. Qual é a instrução que deve ser adicionada ao programa seguinte para que ele trabalhe corretamente?

```
main ( ) {  
    int j, *pj;  
    *pj = 3;  
}
```

```
main () {  
    int j;  
    int *pj = &j;  
    *pj = 3;  
}
```

8. Assumindo que o endereço da variável `x` foi atribuído a um ponteiro `px`, escreva uma expressão que não usa `x` e divida `x` por 5.

9. Qual o valor das seguintes expressões:

```
int i = 3, j = 5;  
int *p = &i, *q = &j;
```

a) `p == &i`

b) `*p - *q`

c) `**&p`

10. Qual será a saída deste programa?

```
main ( ) {  
    int i = 5, *p = &i;  
    printf ( "%u %d %d %d %d \n", p, *p+2, **&p, 3**p, **&p+4);  
}
```