

Lista de Ponteiros - Introdução à Programação C/C++

Nathan Andrade dos Santos Lobo

DRE: 120082390

Questão 2

Como p está apontando para a variável y , p inicia-se sendo 0 e x recebe o valor de p que também é 0 até o momento.

Quando x passa a ser 4, x é decrementado uma linha após o valor do ponteiro p ser incrementado. Por fim, o valor do ponteiro é somado à x e durante todas as operações que envolviam o ponteiro, a variável y também era alterada, pois p aponta para a mesma no início.

Logo,

$$x = 3$$

$$y = 4$$

$$p = 4$$

Questão 3

a.

A variável p é um ponteiro e não está apontando para o valor na memória da variável x e sim para o objeto armazenado diretamente. Para o programa funcionar corretamente seria necessário alterar a 3ª linha para:

$$p = \&x$$

b.

Como i e j são ponteiros, $temp$ está apontando para o valor de i e não para o valor na memória de i . Uma forma de resolver este problema seria dentro do bloco de código da função **troca()** fazer com que as variáveis sejam recebidas tradicionalmente e no bloco da função **main()**, seja feito a passagem por referência ao endereço na memória, da seguinte forma:

Na função **troca()**:

```
temp = a;
```

```
a = b;
```

```
b = temp;
```

Na função **main()**:

```
a = 1000;
```

```
b = 500;
```

```
troca(&a, &b);
```

c.

No modo que está sendo comparado, na verdade, não é a palavra abacate e nem a palavra uva e sim o endereço de memória da primeira posição das duas palavras.

Para que a comparação seja feita efetivamente, eu incluiria a biblioteca *string.h* e utilizaria a função **strcmp()** para fazer a comparação entre as duas strings e ao invés de verificar se os valores na memória de *b* é maior do que *a*, checaríamos se o resultado da comparação for menor do que 0, então *b* é maior do que *a*, do contrário *a* é maior do que *b*.

Questão 5

Utilizar a expressão $v + n$, de modo que *v* seja um vetor, implica que o ponteiro está apontando na memória para a posição inicial do vetor e quando somado *n* bytes, ele prossegue até a posição *n* do vetor. Ou seja, $V[n]$ é nada mais que somar *n* bytes da posição inicial do vetor *V*.

Em relação a imagem, sobre o conteúdo do vetor após os seguintes comandos a seguinte análise foi feita:

Um vetor é iniciado com 99 posições, no primeiro *for* é preenchido o vetor *a* até a sua 98ª posição.

No outro *for*, o vetor *a* está sendo "reescrito" de forma que cada posição vai receber o valor que está no vetor *a* na posição *i*, que já fora preenchido anteriormente.

Questão 9

O valor de *pp* terá o mesmo valor da variável *x*, pois *p* está apontando para o endereço de *x* na memória e $x = 100$.