**LISTA 18**

1. Explique o que é:
   1. Um ponteiro

**É uma variável que armazena o endereço de memória de outra variável.**

* 1. Um ponteiro de ponteiro

**É um ponteiro que armazena o endereço de memória de outro ponteiro.**

* 1. O endereço de um ponteiro

**É a indicação do local da memória onde se encontra esse ponteiro.**

* 1. O conteúdo de um ponteiro

**É o endereço da váriavel que foi apontada.**

* 1. O endereço de uma variável

**É a indicação do local da memória onde se encontra essa variável.**

1. Explique o que acontece quando um ponteiro aponta para uma variável. Usar uma representação gráfica da memória.

|  |  |
| --- | --- |
| **A1h** | **‘a’** |
| **B2h** | **A** |

**char c = ‘a’**

**char \*p = &a**

1. Assuma a seguinte definição:

int a, b;

int \*P1, \*P2, \*\*P3; float \*P4;

Diga quais das sentenças são verdadeiras e quais são falsas (justifique as falsas) a) ( **V** ) a = 10 b) ( **F** ) P4 = P1

c) ( **F** ) b = &a d) ( **V** ) P2 = P1 = &a

e) ( **F** ) P1 = a f) ( **V** ) \*P1 = 20

g) ( **F** ) a = &P1 h) ( **V** ) \*P2 = \*P1

i) ( **F** ) \*P1 = &a j) ( **F** ) \*P3 = &P1

k) ( **F** ) &P1 = &a l) ( **V** ) P3 = &P2

m) ( **F** ) P4 = &a n) ( **V** ) \*\*P3 = \*P1

**b) P4 e P1 são de tipos direfentes, P4 é float e P1 é int;**

**c) b não pode receber um endereço, pois não é um ponteiro;**

**e)P1 é um ponteiro logo não pode receber o valor contido em a e sim o seu endereço. P1 = &a (recebe o endereço) ou \*P1 = a (A variavel apontada recebe a);**

**g)** **a não pode receber um endereço, pois não é um ponteiro;**

**i) A variável P1 precedida de um asterisco diz respeito ao conteudo do local onde ela esta apontando, logo não pode receber um endereço de memória. \*P1 = a (A variavel apontada recebe a) ou P1 = &a (recebe o endereço);**

**j)O um asterisco asterisco tranforma o ponteiro de ponteiro em ponteiro logo ele não pode receber o endereço de um ponteiro.**

**k)P1 não pode receber o endereço de a. P1 = &a;**

**m)P4 é float e a é int.**

1. Explique a diferença entre passagem de parâmetros por valor e por referência.

**Quando passamos por referencia estamos passando o endereço de memória de uma variavel, logo quando a variavel que recebeu esse valor for alterada a variavel passada como parametro tambem ira mudar, ja na passagem por valor isso não acontece.**

1. O que pode acontecer se for atribuído algum valor a um ponteiro que não tenha sido inicializado. Ex:

float \*p;

\*p = 2000;

**Vai dar erro, pois o ponteiro não esta apontando para lugar nenhum.**

1. Em C, não se pode fazer uma função que retorne dois valores, exceto com o uso de estruturas. Uma forma de solucionar esta restrição é com o uso de ponteiros, pois a função pode receber qualquer numero de variáveis por referência.

Faça um programa que, possuindo duas variáveis inteiras a=2 e b=3, chame a função void troca(int \*pa, int \*pb), a qual deve fazer a inversão dos valores de a e b. Ou seja, **b** passa a valer o que **a** valia e **a** passa a valer o que **b** valia. Após a chamada da função troca(), devem ser impressos os valores de **a** e **b**. A função troca também deve ser implementada, e deve ter tipo de retorno void.