

Leia com atenção o enunciado apresentado por cada exercício, e desenvolva o algoritmo solicitado.

1) Quais das instruções abaixo são corretas para declarar um ponteiro?

- a) `int _ptr x;` b) `*int ptr;` c) `int *ptr;` d) `*x;`

2) Assumindo que o endereço de `num` foi atribuído a um ponteiro `pnum`, quais das seguintes expressões são verdadeiras?

- a) `num = pnum;` b) `num = *pnum;` c) `pnum = *num;` d) `pnum = #`

3) 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..

4) Analise o código abaixo em C++, e informe qual das alternativas é falsa.

```
int *pti;  
int i = 10;  
pti = &i;
```

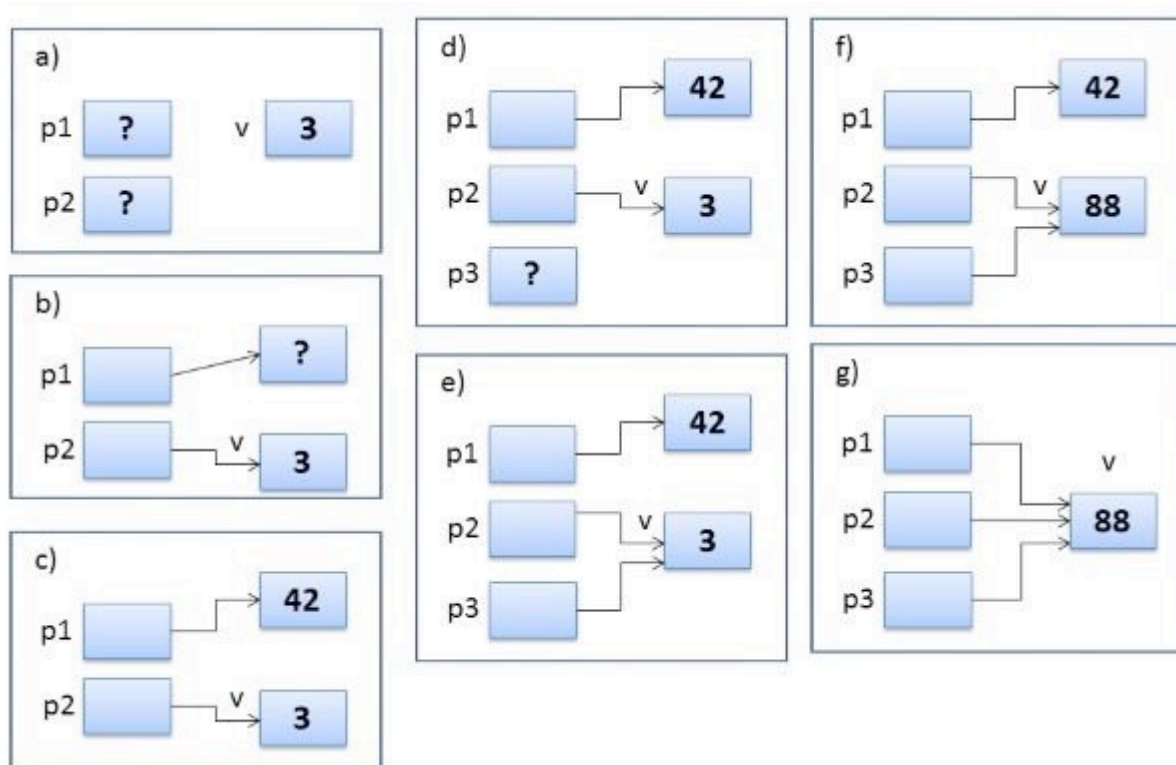
- a) `pti` armazena o endereço de `i`.
b) `*pti` é igual a 10.
c) Ao se executar `*pti = 20;` `i` passará a ter o valor 20.
d) Ao se alterar o valor de `i`, `*pti` será modificado.
e) `pti` é igual a 10.

5) O algoritmo abaixo possui alguns problemas. Olhe com atenção o programa e corrija-o.

Considere que se o usuário digitar 2 para a variável `v`, e 3 para a variável `x`, os resultados a serem exibidos são 2, 3, 5, 2, 2, 2, 2, 2, 2, 2, 8, 8

```
#include <iostream>  
using namespace std;  
  
int main() {  
    int *p1, *p2, *p3, v, x;  
  
    cout << "V ...: ";  
    cin >> v;  
    cout << "X ...: ";  
    cin >> x;  
  
    p3 = new;  
    p1 = &v;  
    p2 = &x;  
    *p3 = v + *p2;  
    cout << *p1 << endl;  
  
    cout << *p2 << endl;  
    cout << *p3 << endl;  
    delete p3;  
    *p3 = p1;  
    cout << *p1 << endl;  
    cout << *p2 << endl;  
    cout << *p3 << endl;  
    p3 = p2;  
    cout << *p1 << endl;  
    cout << *p2 << endl;  
    cout << *p3 << endl;  
    *p3 = 8;  
    cout << *p1 << endl;  
    cout << *p2 << endl;  
    cout << *p3 << endl;  
    return 0;  
}
```

6) Faça um programa utilizando ponteiros que obedeça ao teste de mesa abaixo.



7) Execute o programa abaixo e informe os valores e o que cada uma das linhas está fazendo.

```
#include <iostream>
using namespace std;

int main() {
    int i=5, *p, **l;;
    p = &i;
    l = &p;
    cout << "i .....: " << i << endl;
    cout << "p = &i \n";
    cout << "&i .....: " << &i << endl;
    cout << "p .....: " << p << endl;
    cout << "*p+2 .....: " << *p+2 << endl;
    cout << "&p .....: " << &p << endl;
    cout << "*&p .....: " << *&p << endl;
    cout << "**&p .....: " << **&p << endl;
    cout << "3**p .....: " << 3**p << endl;
    cout << "**&p+4 ..: " << **&p+4 << endl;
    cout << "l .....: " << l << endl;
    cout << "*l .....: " << *l << endl;
    return 0;
}
```