

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 = &num;`

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 <stdio.h>
```

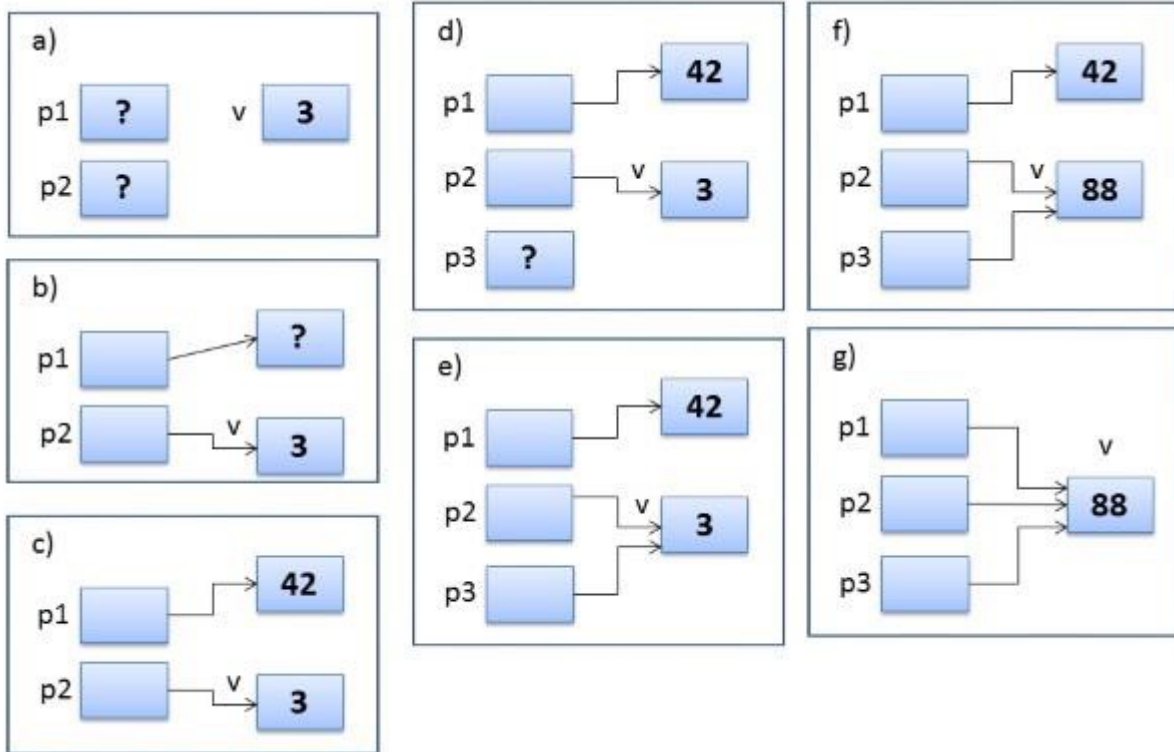
```
int main() {  
    int *p1, *p2, *p3, v, x;  
  
    printf("V ...: ");  
    scanf("%d",&v);  
    printf("X ...: ");  
    scanf("%d",&x);  
  
    p3 = malloc();  
    p1 = &v;  
    p2 = &x;  
    *p3 = v + *p2;  
    printf("\n\n%d",*p1);  
    printf("\n%d",*p2);  
    printf("\n%d",*p3);
```



```
    free(p3);  
    *p3 = p1;  
    printf("\n\n%d",*p1);  
    printf("\n%d",*p2);  
    printf("\n%d",*p3);  
    p3 = p2;  
    printf("\n\n%d",*p1);  
    printf("\n%d",*p2);  
    printf("\n%d",*p3);  
    *p3 = 8;  
    printf("\n\n%d",*p1);  
    printf("\n%d",*p2);  
    printf("\n%d",*p3);  
    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 <stdio.h>
int main()
{
    int i=5, *p, **l;
    p = &i;
    l = &p;
    printf("i .....: %d \n", i);
    printf("p = &i \n");
    printf("&i .....: %x \n", &i);
    printf("p .....: %x \n", p);
    printf("*p+2 .....: %d \n", *p+2);
    printf("&p .....: %x \n", &p);
    printf("*&p .....: %d \n", *&p);
    printf("**&p .....: %d \n", **&p);
    printf("3*p .....: %d \n", 3*p);
    printf("**&p+4 ...: %d \n", **&p+4);
    printf("l .....: %x \n", l);
    printf("*l .....: %x \n", *l);
    return 0;
}
```