

7. Se **i** e **j** são variáveis inteiras e **p** e **q** ponteiros para **int**, quais das seguintes expressões de atribuição são ilegais?

- a) `p = &i;` b) `*q = &j;` c) `p = &*i;` d) `i = (*&)j;` e) `i = *&j;`
f) `i = *&*j;` g) `q = *p;` h) `i = (*p)++ + *q;`

8. Supor a declaração: **int mat[4], *p, x;** Quais expressões são válidas? Justifique:

- a) `p = mat + 1;` b) `p = mat++;`
c) `p = ++mat;` d) `x = (*mat)++;`

9. Qual é a saída do programa a seguir?

```
#include<stdio.h>
```

```
int main()
```

```
{
```

```
    int vet[] = {4,9,13};
```

```
    int i;
```

```
    for(i=0;i<3;i++)
```

```
        printf("%d ",*(vet+i));
```

```
    for(i=0;i<3;i++)
```

```
        printf("%X ",vet+i);
```

```
    return 0;
```

```
}
```

10. Seja a seguinte sequência de instruções em um programa C:

```
int *pti;
```

```
int i = 10;
```

```
pti = &i;
```

Qual é a afirmativa falsa?

- 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

11. Seja vet um vetor de 4 elementos: **TIPO vet[4]**. Supor que depois da declaração, vet esteja armazenado no endereço de memória 4092 (ou seja, o endereço de vet[0]). Supor também que na máquina usada uma variável do tipo char ocupa 1 byte, do tipo int ocupa 2 bytes, do tipo float ocupa 4 bytes e do tipo double ocupa 8 bytes. Qual o valor de vet+1, vet+2 e vet+3 se:

- a) vet for declarado como char?
- b) vet for declarado como int?
- c) vet for declarado como float?
- d) vet for declarado como double?

12. Responda as perguntas abaixo. Se achar necessário, faça testes no programa em C.

- a) Explique a diferença entre p++; (*p)++; *(p++);
- b) O que quer dizer *(p+10);?

13. Qual o valor de y no final do programa? Tente primeiro descobrir e depois verifique no computador o resultado. A seguir, escreva um /* comentário */ em cada comando de atribuição explicando o que ele faz e o valor da variável à esquerda do '=' após sua execução.

```
#include<stdio.h>
```

```
int main()
```

```
{
```

```
    int y, *p, x;
```

```
    y = 0;
```

```
    p = &y;
```

```
    x = *p;
```

```
    x = 4;
```

```
    (*p)++;
```

```
    x--;
```

```
    (*p) += x;
```

```
    printf ("y = %d\n", y);
```

```
    return(0);
```

```
}
```

14. Diga quais expressões abaixo são válidas ou não. Considere as declarações

```
int vetor[10];
```

```
int *ponteiro;
```

[illegible]