

Suponha que as seguintes declarações tenham sido realizadas:

```
float aloha[10], coisas[10][5], *pf, value = 2.2;  
int i=3;
```

Identifique quais dos seguintes comandos é válido ou inválido:

a. `aloha[2] = value;` R.: Válido, pois o elemento 3 do vetor “aloha” copia e guarda o conteúdo da variável “value” por meio deste comando.

b. `scanf("%f", &aloha);` R.: Válido, pois através desse comando, o programa coleta um valor real digitado pelo usuário e o armazena no endereço de memória do primeiro elemento do vetor “aloha”.

c. `aloha = "value";` R.: Inválido, pois “aloha” não é uma string, ou seja, seus elementos não podem armazenar caracteres.

d. `printf("%f", aloha);` R.: Inválido, pois o programa só irá imprimir na tela o valor do primeiro elemento de “aloha” por meio da expressão: “`printf(“%f”, *aloha);`”.

e. `coisas[4][4] = aloha[3];` R.: Válido, pois o elemento da quinta linha e da quinta coluna da matriz “coisas”, por meio desse comando, copia e armazena o conteúdo do quarto elemento do vetor “aloha”.

f. `coisas[5] = aloha;` R.: Inválido, pois para que um elemento da matriz “coisas” copie e armazene o conteúdo de um elemento do vetor “aloha”, deve ser declarado na expressão tanto a

linha quanto a coluna do elemento. Ex: `“coisas[5][3] = aloha;”`.

g. `pf = value;` R.: Inválido, pois o ponteiro `“pf”` só pode guardar o endereço de memória da variável `“value”` e assumir o seu conteúdo por meio do comando: `“pf = &value;”`.

h. `pf = aloha;` R.: Válido, pois o uso do operador `“&”` antes do nome de um vetor é desnecessário em expressões como essa para que um ponteiro guarde o endereço de memória do primeiro elemento desse vetor.