Exercício 2

Preencha os espaços em cada uma das sentenças a seguir:

a) C armazena listas de valores em **Arrays ou vetores.**

b) Os elementos de um array têm em comum o fato de que eles **são do mesmo tipo.**

c) Ao se referir a um elemento do array, o número da posição contido dentro dos parênteses é um(a) **índice (ou subscrito).**

d) Os nomes dos cinco elementos do array p são **p[0], p[1], p[2], p[3] e p[4]**.

e) O conteúdo de um elemento em particular de um array é chamado **Valor** desse elemento.

f) Dar nome a um array, indicar seu tipo e especificar o número de elementos contidos nele é chamado **declarar** o array.

g) Em um array de subscrito duplo (matriz), o primeiro subscrito (por convenção) identifica a **linha** de um elemento, e o segundo subscrito (por convenção) identifica a **coluna** de um elemento.

h) Um array \*m\*-por-\*n\* contém **m** linhas, **n** colunas e **m \* n** elementos.

i) O nome do elemento do array d na linha 3, coluna 5 é **d[2][4]**.

Exercício 3

Encontre o erro em cada um dos segmentos de programa a seguir.

a)int \*number;

printf("%d\n", \*number);

**Resposta: Erro: A variável number é um ponteiro que não foi inicializado, e está sendo desreferenciada. Isso resulta em comportamento indefinido.**

**Correção:** Inicializar o ponteiro antes de usá-lo:

int value = 10;

int \*number = &value;

printf("%d\n", \*number);

b) float \*realPtr;

long \*integerPtr;

integerPtr = realPtr;

**Resposta:** **Erro:** Tipos incompatíveis: ponteiro para float não deve ser atribuído a ponteiro para long sem casting.  
**Correção:** Usar cast (com cuidado) ou corrigir tipos:

integerPtr = (long \*) realPtr;

c) int \*x, y;

x = y;

**Resposta: Erro: x é ponteiro, y é inteiro. Atribuir int para int \* está incorreto.  
Correção: ou torna y um ponteiro também, ou x uma variável inteira:**

**int \*x, \*y;**

**x = y;**

**int x, y;**

**x = y;**

d) char s[] = "um array de caracteres";

int count;

for (; \*s != '\0'; s++)

printf("%c ", \*s);

**Resposta: Erro: Está tentando modificar s, que é um array. Arrays não podem ser incrementados como ponteiros.  
Correção: Usar ponteiro auxiliar:**

**char s[] = "um array de caracteres";**

**char \*ptr = s;**

**for (; \*ptr != '\0'; ptr++)**

**printf("%c ", \*ptr);**

e) short \*numPtr, result;

void \*genericPtr = numPtr;

result = \*genericPtr + 7;

**Resposta:** **Erro:** Não se pode desreferenciar um void \* sem converter.  
**Correção:** Fazer cast antes de desreferenciar:

result = \*((short \*)genericPtr + 7);

f) float x = 19.34;

float xPtr = &x;

printf("%f\n", xPtr);

**Resposta: Erro: %f espera float, mas está passando ponteiro (xPtr).  
Correção: Desreferenciar o ponteiro:**

**printf("%f\n", \*xPtr);**

g) char \*s;

printf("%s\n", s);

**Resposta: Erro: Ponteiro s não foi inicializado — pode causar comportamento indefinido.  
Correção: Inicializar com string válida:**

**char \*s = "Olá";**

**printf("%s\n", s);**