

Lista de exercícios

– Ponteiros –

1. Considerando o programa a seguir:

```
char i = 'T', j = 'S', k = 'U';  
char *p1 = &i;  
char *p2 = &j;  
char *p3;
```

Suponha que o endereço de **i** seja 6940, o endereço de **j** seja 9772 e o endereço de **k** seja 2224. O que será impresso quando as seguintes declarações forem executadas sequencialmente?

```
p3 = &k;  
printf("p3 = %c\n", *p3); // (1)  
printf("p3 = %d\n", p3); // (2)
```

```
p3 = p1;  
printf("p3 = %c\n", *p3); // (3)  
printf("p3 = %d\n", p3); // (4)
```

```
p3 = p2;  
printf("p3 = %c\n", *p3); // (5)  
printf("p3 = %d\n", p3); // (6)
```

2. Considerando o programa a seguir:

```
int *p;  
int i;  
int k;  
i = 42;  
k = i;  
p = &i;
```

Qual das seguintes alternativas irá alterar o valor de **i** para 75?

- (a) `k = 75;`
- (b) `*k = 75;`
- (c) `p = 75;`
- (d) `*p = 75;`
- (e) Duas ou mais alternativas vão alterar **i** para 75.

3. Utilize suas palavras para descrever o(s) erro(s) do programa a seguir:

```
char c = 'A';  
double *p = &c;
```

Para os programas a seguir, forneça as saídas:

4.

```
int x, y, z, w;  
int *pa, *pb, *pc, *pd;  
  
x = 10; y = 20; z = 30;  
pa = &x;  
pb = &y;  
pc = pb;  
  
printf("%d, %d, %d\n", *pa, *pb, *pc); // (1)  
  
pb = pa;  
printf("%d, %d, %d\n", *pa, *pb, *pc); // (2)  
  
pa = &z;  
printf("%d, %d, %d\n", *pa, *pb, *pc); // (3)  
  
*pa = *pb;  
printf("%d, %d, %d\n", *pa, *pb, *pc); // (4)
```
5.

```
void F(int x, int y) {  
    int temp = x;  
    x = y;  
    y = temp;  
}  
  
int main() {  
    int data1 = 10, data2 = 20;  
    int p1 = data1;  
    int p2 = data2;  
  
    F(p1, p2);  
    printf("p1 = %d, p2 = %d\n", p1, p2); // (1)  
}
```
6.

```
void F(int* x, int* y) {  
    int temp = *x;  
    *x = *y;  
    *y = temp;  
}  
  
int main() {  
    int data1 = 10, data2 = 20;  
    int *p1 = &data1;  
    int *p2 = &data2;  
  
    F(p1, p2);  
    printf("*p1 = %d, *p2 = %d\n", *p1, *p2); // (1)  
}
```

7. Houve diferença entre os valores nas saídas dos exercícios 5 e 6? Por que?

Nos exercícios a seguir, corrija os erros:

8.

```
void quadrado(int *p) { // (1)
    *p = *p * *p;      // (2)
}                       // (3)
                        // (4)
int main() {           // (5)
    int x = 13, *p;    // (6)
    p = x;              // (7)
    quadrado(*p);      // (8)
}                       // (9)
```
9.

```
void quadrado(int *p) { // (1)
    *p = *p * *p;      // (2)
}                       // (3)
                        // (4)
int main() {           // (5)
    int x = 13, *p;    // (6)
    p = *x;             // (7)
    quadrado(p);       // (8)
}                       // (9)
```
10.

```
void quadrado(int *p) { // (1)
    *p = *p * *p;      // (2)
}                       // (3)
                        // (4)
int main() {           // (5)
    int x = 13, p;     // (6)
    p = &x;            // (7)
    quadrado(p);       // (8)
}                       // (9)
```
11.

```
void quadrado(int *p) { // (1)
    *p = *p * *p;      // (2)
}                       // (3)
                        // (4)
int main() {           // (5)
    int *x = 13, *p;   // (6)
    &p = *x;            // (7)
    quadrado(*p);      // (8)
}                       // (9)
```

12. Forneça para o programa a seguir o valor da variável **temp** em cada instrução de atribuição. Para isso, suponha que as linhas sejam executadas sequencialmente.

```
int main() {
    char blocos[3] = { 'A', 'B', 'C' };
    char *ptr = &blocos[0];
    char temp;

    temp = blocos[0];      // (1)
    temp = *(blocos + 2);  // (2)
    temp = *(ptr + 1);     // (3)
    temp = *ptr;           // (4)

    ptr = blocos + 1;
    temp = *ptr;           // (5)
    temp = *(ptr + 1);     // (6)

    ptr = blocos;
    temp = ++*ptr;         // (7)
    temp = ++*ptr;         // (8)
    temp = *ptr++;         // (9)
    temp = *ptr;           // (10)
}
```

Respostas:

Exercício 1:

- 1) *p3 = U
- 2) p3 = 2224
- 3) *p3 = T
- 4) p3 = 6940
- 5) *p3 = S
- 6) p3 = 9772

Exercício 2:

Alternativa D

Exercício 3:

O tipo de variável do ponteiro deve ser o mesmo da variável para qual está apontado. Deve-se substituir a variável do tipo double por char.

Exercício 4:

- 1) 10, 20, 20
- 2) 10, 10, 20
- 3) 30, 10, 20
- 4) 10, 10, 20

Exercício 5:

- 1) p1 = 10, p2 = 20

Exercício 6:

- 1) *p1 = 20, *p2 = 10

Exercício 7:

Sim, haja vista que em 5 não há ponteiros para realizar a troca dos valores através de um procedimento. Em 6, a utilização dos ponteiros permite a troca sem necessidade de retorno no procedimento **TROCAR**.

Exercício 8:

Linha 7: O ponteiro deve apontar para o endereço da variável **x**.

Correto: `p = &x;`

Linha 8: Deve ser passado o ponteiro para o procedimento **quadrado**, e não o valor do ponteiro.

Correto: `quadrado(p);`

Exercício 9:

Linha 7: O ponteiro deve apontar para o endereço da variável **x**.

Correto: `p = &x;`

Exercício 10:

Linha 6: A variável **p** deve ser um ponteiro.

Correto: `int x = 13, *p;`

Exercício 11:

Linha 6: A variável **x** não deve ser um ponteiro.

Correto: `int x = 13, *p;`

Linha 7: O ponteiro deve apontar para o endereço da variável **x**.

Correto: `p = &x;`

Linha 8: Deve ser passado o ponteiro para o procedimento **quadrado**, e não o valor do ponteiro.

Correto: `quadrado(p);`

Exercício 12:

- 1) A 2) C
- 3) B 4) A
- 5) B 6) C
- 7) B 8) C
- 9) C 10) C