

Lista 4 - Ponteiros

1. Considere o seguinte trecho de código:

```
1 #include <stdio.h>
2
3 int main() {
4     int *p, a;
5
6     a = 10;
7     p = &a;
8     *p = *p * 2;
9     printf("a = %d\n", a);
10
11     return 0;
12 }
```

Qual o valor da variável **a** ao final da execução? (Tente primeiro responder apenas analisando o código. Confirme a sua resposta compilando e executando o código.).

2. Considere o seguinte trecho de código:

```
1 #include <stdio.h>
2
3 int main() {
4     float *q, b;
5
6     b = 3.14;
7     *q = *q + 1;
8     printf("*q = %f\n", *q);
9
10    return 0;
11 }
```

Onde está o erro neste código? O que acontecerá se você executá-lo desta maneira?

3. Declare duas variáveis **x** e **y** do tipo **int** e duas variáveis **p** e **q** do tipo **int***. Atribua o valor 2 para **x** e 8 para **y**, o endereço de **x** para **p** e o endereço de **y** para **q**. Então, imprima (usando **printf**) as seguintes informações:

- (a) O endereço de **x** e o valor de **x**;
- (b) O valor de **p** e o valor de ***p**;
- (c) O endereço de **y** e o valor de **y**;
- (d) O endereço de **p** (não o seu conteúdo!).

Obs.: Um endereço deve ser impresso utilizando o especificador de formato **%p** no **printf**.

4. Discuta, passo a passo, o efeito do seguinte fragmento de código:

```
1 int *v;
2 v = (int *) malloc (10 * sizeof(int));
```

5. Escreva um programa que solicite ao usuário o número de notas a serem entradas e aloque dinamicamente um vetor com a dimensão especificada para armazenar as entradas. Solicite as notas e chame uma função que retorne a média aritmética das notas. Após imprimir a média o programa deve liberar a memória dinâmica que fora alocada.
6. Implemente a função `troque1()` que faz uso da passagem por referência com argumentos de referência. Essa função deve realizar a troca dos argumentos `x` e `y` passados por referência. Teste na `main` se a sua função está funcionando corretamente.
7. Implemente a função `troque2()` que faz uso da passagem por referência com argumentos de ponteiro. Esta função deve realizar a troca dos argumentos `x` e `y` passados por referência (passa-se o endereço neste caso específico). Teste na `main` se a sua função está funcionando corretamente.
8. Escreva um programa que recebe três variáveis do tipo `int` (`a`, `b`, `c`) e rotacione os valores de forma que o valor de `a` vá para `b`, o valor de `b` para `c` e o valor de `c` para `a`. Utilize chamadas a função `troque` desenvolvida anteriormente (escolha uma das duas implementações da função `troque`).
9. Crie uma função chamada `print_addr(int x)` cujo propósito seja apenas imprimir o conteúdo da variável `x` e seu endereço. Declare uma variável `x` do tipo `int` na `main` e inicialize-a com o valor 3. Imprima seu conteúdo e endereço dentro da `main`. Então, chame a função `print_addr` passando a variável `x` como argumento. Compare os resultados. Este comportamento é o esperado?
10. O seguinte trecho de código calcula a soma dos n primeiros números naturais.

```

1  #include <stdio.h>
2
3  int main () {
4      int n, soma;
5
6      soma = 0;
7
8      scanf ("%d", &n);
9
10     while (n > 0) {
11         soma = soma + n;
12         n -= 1;
13     }
14
15     printf ("A soma dos %d primeiros numeros e: %d", n, soma);
16
17     return 0;
18 }
```

Reescreva o código utilizando duas funções. Uma função deve calcular a soma utilizando passagem por referência e a segunda função exibe o resultado utilizando passagem por valor.