UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
CURSO DE CIÊNCIA DA COMPUTAÇÃO
PROGRAMAÇÃO DE COMPUTADORES

LABORATÓRIO 28

ARMAZENAMENTO E REFERÊNCIAS

EXERCÍCIOS DE REVISÃO

VOCÊ DEVE ACOMPANHAR PARA OBTER INFORMAÇÕES COMPLEMENTARES

1. Suponha que todas as funções de um programa precisem acessar um ponteiro global chamado print. O ponteiro é definido na função principal e acessado por funções em outros arquivos. Construa a função `mensagem()` em um outro arquivo de código fonte de forma que ela utilize print para obter a saída abaixo:

```
Usar print normal [s/n]? s
Finalizando programa!
```

```
Usar print normal [s/n]? n
<<< Finalizando programa! >>>
```

```
#include <iostream>
using namespace std;

void (*print)(const char[]);  // ponteiro global

void normal(const char texto[]);
void alternativo(const char texto[]);
void mensagem();

int main()
{
    cout << "Usar print normal [s/n]? ";
    char opcao;
    cin >> opcao;

    if (opcao == 's') print = normal;
    else print = alternativo;
    mensagem();
}

void normal(const char texto[])
{ cout << texto << endl; }

void alternativo(const char texto[])
{ cout << "<<< " << texto << " >>>" << endl; }
```

EXERCÍCIOS DE FIXAÇÃO

VOCÊ DEVE FAZER OS EXERCÍCIOS PARA FIXAR O CONTEÚDO

1. Escreva uma função que mantenha um contador de quantas vezes ela foi chamada. A função deve mostrar o valor desse contador a cada chamada. Por exemplo, se a função for chamada 3 vezes, a saída do programa deve ser como mostrada abaixo:

```
1a chamada da função.
2a chamada da função.
3a chamada da função.
```

2. Considerando o modelo de registro abaixo:

```
struct Caixa
{
    char marca[40];
    float altura;
    float largura;
    float comprimento;
    float volume;
};
```

   a. Escreva uma função que receba uma referência para um registro **caixa** e mostre o valor de cada membro.

   b. Escreva uma função que receba uma referência para um registro **caixa** e modifique o membro **volume** para o produto das suas dimensões.

3. Refaça a questão anterior usando ponteiros no lugar de referências e tente observar as diferenças entre cada versão. Que vantagens a versão com referências tem sobre a verão com ponteiros?

4. Construa uma função calcular que use referências em seus parâmetros com o objetivo de retornar mais de um valor ao programa principal. A função deve receber um par de valores, definido pelo registro abaixo, e dois valores inteiros que devem ser modificados para guardar, respectivamente a soma e a multiplicação dos valores no par. Utilize const para os parâmetros que não serão modificados dentro da função.

```
struct par { int x; int y;};
```

```
Digite um par de valores:
4 5
Soma: 9
Multiplicação: 20
```

EXERCÍCIOS DE APRENDIZAGEM

VOCÊ DEVE ESCREVER PROGRAMAS PARA REALMENTE APRENDER

1. Um corredor deseja computar o tempo total de uma corrida a partir dos tempos parciais de cada volta. Para ajudá-lo, crie uma função `TempoDaVolta()` que recebe o tempo de uma volta, o acumula em uma variável local estática e retorna o total acumulado. Crie um programa que leia o tempo de cada volta e chame a função `TempoDaVolta()` com o valor lido. O programa deve rodar até que o corredor entre com um valor inválido para o tempo da volta.

```
Volta: 16.2
Tempo total = 16.2 segs.
Volta: 18.5
Tempo total = 34.7 segs.
Volta: 20.1
Tempo total = 54.8 segs.
Volta: 21.4
Tempo total = 76.2 segs.
Volta: fim
```

2. Considerando o registro tupla definido abaixo, construa uma função `trocar()` que receba duas tuplas e inverta seus valores. Escreva um programa para testar a função.

```
struct tupla
{
    int a;
    int b;
    int c;
};
```

```
Tupla A: 15 20 25
Tupla B: 40 50 60

Invertendo...

Tupla A: 40 50 60
Tupla B: 15 20 25
```

3. A função `operator<<` pode ser utilizada para ensinar ao cout como exibir variáveis de um tipo definido pelo programador. Utilize referências para construir uma função `operator<<` que exiba um par de elementos na tela. Use o mesmo tipo par definido na questão 4 dos exercícios de fixação. Agora construa um programa para ler um arquivo texto contendo um número indefinido de pares de números separados por espaço e exiba-os com cout, de forma que cada par seja exibido como mostrado abaixo:

```
Pares: [2,4] [3,7] [8,5] [9,2]
```