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

LABORATÓRIO 14

PONTEIROS

EXERCÍCIOS DE REVISÃO

VOCÊ DEVE ACOMPANHAR PARA OBTER INFORMAÇÕES COMPLEMENTARES

1. Suponha que **`litros`** é uma variável de tipo double. Declare um ponteiro que aponte para **`litros`** e use o ponteiro para mostrar o valor de **`litros`**.

```
O valor de litros é 3.4
```

2. Suponha que **`torque`** é um vetor de 10 floats. Declare um ponteiro que aponte para o primeiro elemento de **`torque`** e use o ponteiro para mostrar o primeiro e o último elemento do vetor.

```
Torques: 2.5, 8.1, 3.4, 9.2, 5.7, 9.6, 6.3, 8.0, 5.4, 4.9

Primeiro: 2.5
Último: 4.9
```

3. Defina um registro que descreva um **`peixe`**. O registro deve incluir o **tipo** (string), o **peso** (ponto-flutuante) e o **comprimento** (inteiro) do peixe. Em seguida mostre:
   a. Como criar uma variável de tipo **peixe**
   b. Como criar um ponteiro para uma variável de tipo **`peixe`**.

4. Construa uma função que receba um peixe e exiba o seu conteúdo.
   a. Faça uma versão utilizando um parâmetro tipo peixe
   b. Faça uma versão utilizando um parâmetro tipo ponteiro para peixe

```
Sem ponteiro: Piaba com 6.2g e 5cm
Com ponteiro: Piaba com 6.2g e 5cm
```

5. Descubra o que acontece ao tentarmos acessar um ponteiro que contém um endereço inválido. Para isso tente mostrar o conteúdo apontado por um ponteiro recém-criado:

```
int * ptr = (int*) 0x01;
cout << *ptr;
```

EXERCÍCIOS DE FIXAÇÃO

VOCÊ DEVE FAZER OS EXERCÍCIOS PARA FIXAR O CONTEÚDO

1. Modifique o programa abaixo para que ele mostre o número 10 na tela usando o ponteiro q:

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

int main()
{
    int x, * p, ** q;
    p = &x;
    q = &p;
    x = 10;
    cout << q << endl;   // não está mostrando o número 10
}
```

2. Crie uma variável do tipo char inicializada para o caractere 'A' e um ponteiro que aponte para esta variável. Modifique a variável criada usando o ponteiro, de forma que seu conteúdo agora seja 'B'. Por fim, mostre o conteúdo da variável e o conteúdo apontado pelo ponteiro.

3. Construa duas funções que realizem o incremento de um número em uma unidade. A função Mais deve receber um número inteiro através de um ponteiro. A função Incrementa deve receber um valor inteiro, sem usar ponteiros, e retornar o valor incrementado em uma unidade. Utilize as duas funções como no exemplo abaixo:

```
Digite um valor: 7

Resultado após Mais: 8
Resultado após Incrementa: 9
```

Dica: observe que a função Mais pode modificar diretamente o valor da variável recebida, enquanto a função Incrementa precisa retornar o valor porque ela tem acesso apenas a uma cópia da variável.

4. Uma cor pode ser representada pela combinação de 4 valores de intensidade para R (Red), G (Green), B (Blue) e A (Alpha). Esses valores podem ser guardados em um registro com 4 inteiros de 8 bits (0-255) ou por um valor inteiro de 32 bits codificado com os 4 valores. Construa uma união para armazenar uma cor. Em seguida construa uma função para ler do usuário uma cor no formato RGBA e outra para ler uma cor no formato inteiro de 32 bits. Ambas as funções devem receber o endereço de uma variável do tipo cor e modificar a variável recebida, sem retornar valor.

```
Digite uma cor no formato
RGBA : 38 38 38 0
Int32: 640034304
```

EXERCÍCIOS DE APRENDIZAGEM

VOCÊ DEVE ESCREVER PROGRAMAS PARA REALMENTE APRENDER

1. Declare um registro "Tigela" com os campos estado (cheia ou vazia) e tipo de alimento (sopa ou canja). Crie uma função "Fome" que recebe um ponteiro para uma Tigela e altera o seu estado para "vazia". Na função principal crie uma tigela cheia, crie um ponteiro que aponta para essa tigela e então mostre como a tigela estava antes da janta. Depois chame a função Fome com o ponteiro que aponta para a tigela e ao fim mostre a tigela depois da janta.

2. Declare um registro Horário com os campos horas e minutos. Crie uma função MostrarHorario que deve receber um ponteiro para um Horário e mostrá-lo no formato HH:MM. Na função principal, declare uma variável do tipo Horário e um ponteiro que aponta para ela. Peça que o usuário digite o horário atual e guarde-o na variável. Usando o ponteiro, incremente o horário recebido em uma hora e em seguida mostre o horário corrigido com MostrarHorario.

```
Que horas são? 9:50
Seu relógio está atrasado, o horário correto é 10:50.
```

3. Descubra qual é a saída do seguinte trecho de código, sem auxílio do computador. Depois rode o programa passo a passo com o depurador para verificar se conseguiu chegar na resposta certa.

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

int main()
{
    int valor = 10, *temp, soma = 0;
    temp = &valor;
    *temp = 20;
    temp = &soma;
    *temp = valor;
    cout << "valor: " << valor << "\nsoma: " << soma << endl;
}
```

Sugestão: observe como as variáveis se alteram com a execução do programa.

4. Crie um registro Imagem que contenha campos para nome, altura, largura e tipo, sendo o tipo um dos seguintes valores: JPG, PNG ou BMP. Use uma enumeração para guardar o tipo da imagem. No programa principal inicialize uma variável do tipo Imagem e passe seu endereço para uma função Detalhes. A função deve receber o endereço de uma Imagem e exibir os seus dados no formato do exemplo abaixo.

```
A imagem “backg.png” com tamanho 1920x1080 tem formato PNG.
```