

LABORATÓRIO 12

REGISTROS

EXERCÍCIOS DE REVISÃO

VOCÊ DEVE ACOMPANHAR PARA OBTER INFORMAÇÕES COMPLEMENTARES

1. A combinação da luz vermelha, da luz verde e da luz azul é capaz de gerar qualquer cor no espectro visível. Essa combinação é conhecida pela sigla RGB, do inglês, Red, Green e Blue. Ela é muito utilizada no computador para representar a cor dos pixels em arquivos de imagens, vídeos e no monitor. Uma intensidade de cor vermelha, verde ou azul pode ser representada por valores inteiros na faixa de 0 a 255. Com essa representação, uma cor pode ser descrita por 3 números. Por exemplo, (255,255,255) resulta na cor branca, (0,0,0) é a cor preta, (255,128,0) é uma cor laranja. Construa um registro Color para representar cores RGB usando números inteiros. Forneça funções para ler, exibir, somar e multiplicar cores.

$$(R_x, G_x, B_x) + (R_y, G_y, B_y) = (R_x + R_y, G_x + G_y, B_x + B_y)$$
$$(R_x, G_x, B_x) * (R_y, G_y, B_y) = (R_x * R_y / 255, G_x * G_y / 255, B_x * B_y / 255)$$

Na soma, caso o resultado ultrapasse o valor 255, ele deve ficar limitado a 255. Construa um programa para testar as funções, como no exemplo abaixo:

```
Digite duas cores:  
RGB(0,163,215)  
RGB(255,170,0)  
Soma = RGB(255,255,215)  
Mult = RGB(0,108,0)
```

2. Altere o programa da questão anterior para que a leitura/escrita de uma cor seja feita com cin/cout e a soma e multiplicação com os operadores de + e *, como mostrado no exemplo abaixo:

```
int main()  
{  
    cout << "Digite duas cores:\n";  
    Color a, b;  
    cin >> a;  
    cin >> b;  
  
    cout << "Soma = " << a + b << endl;  
    cout << "Mult = " << a * b << endl;  
}
```

EXERCÍCIOS DE FIXAÇÃO

VOCÊ DEVE FAZER OS EXERCÍCIOS PARA FIXAR O CONTEÚDO

1. O registro **chocolate** contém três membros:

- O primeiro guarda a marca da barra de chocolate
- O segundo membro guarda o peso da barra (possui uma parte fracionária)
- O terceiro membro guarda o número de calorias (um valor inteiro)

Escreva um programa que declare tal registro e crie uma variável chamada **lanche**, inicializando seus membros para “Charge”, 2.3, e 350, respectivamente. A inicialização deve ser parte da declaração da variável. Finalmente, o programa deve mostrar o conteúdo da variável **lanche**.

2. O registro **chocolate** contém três membros, como descrito na questão anterior. Escreva um programa para criar um vetor capaz de armazenar três **chocolates**. Inicialize-os para valores de sua escolha, e então mostre o conteúdo de cada registro.

3. Crie um registro para descrever uma data. O registro deve ser capaz de armazenar o dia, o mês (em formato numérico) e o ano (no formato numérico AAAA). Em seguida faça um programa que receba a data de nascimento de uma pessoa e a data atual. Calcule e mostre quantos dias de vida ela tem.

Use o novo tipo criado para armazenar as datas e uma **função** para calcular o número de dias de vida. A função deve receber dois registros do tipo data e retornar um número inteiro referente ao número de dias.

- Considere que todos os meses têm 31 dias
- Desconsidere os anos bissextos
- Não utilize instruções de decisão (if)

Data de nascimento: **01/01/1990**
Data de hoje: **26/10/2010**
Você tem 7604 dias de vida.

4. Altere o programa anterior de forma que cin e cout possam ser usados para ler e exibir datas. Além disso, a subtração de variáveis tipo data deve resultar no número de dias entre as duas datas. Refaça o programa principal para usar cin, cout e subtração com datas.

EXERCÍCIOS DE APRENDIZAGEM

VOCÊ DEVE ESCREVER PROGRAMAS PARA REALMENTE APRENDER

1. Defina um registro Conta Bancária, com os campos identificador, nome do cliente e saldo. Na função principal, use o novo tipo Conta Bancária para criar uma variável e peça ao usuário para entrar com os dados da conta. O saldo inicial deve ser zero. Mostre a conta depois de criada e pergunte quanto o usuário deseja depositar na conta, modificando o seu saldo.
2. Defina um registro Palavra para armazenar 3 strings, uma para cada tradução de uma palavra, em português, inglês e espanhol. Em seguida crie um vetor do tipo Palavra chamado dicionário, com espaço para dez palavras. Inicialize o vetor com 2 palavras de sua escolha, lembrando que cada palavra deve estar nas três línguas. Por fim, peça para que o usuário entre com as informações para mais uma palavra, armazene-a no vetor e mostre todas as palavras do dicionário.
3. Crie uma função para exibir os dados de uma Conta Bancária e outra para exibir uma Palavra, definidas de acordo com as questões 1 e 2. Modifique os programas das questões anteriores de forma que eles usem estas novas funções.
4. Defina um registro Livro e um registro Jogo, com informações pertinentes a cada um. Na função principal, crie um vetor de jogos emprestados e um vetor de livros emprestados. Pergunte ao usuário se ele quer um livro emprestado ou um jogo. Se ele quiser um livro, peça para que ele digite 1, se ele quiser um jogo, peça para que ele digite 0. Use o valor digitado para decidir em qual vetor de empréstimo o novo elemento deve ser adicionado.
5. Crie um registro para definir uma data (com dia, mês e ano) e outro para definir um horário (hora, minutos e segundos). Em seguida crie um registro para definir um evento, composto por uma data, um horário e um local. Peça ao usuário que entre com os dados de um evento e armazene-os em uma variável do tipo evento.
6. Crie uma função para ler, exibir e subtrair elementos do tipo horário, definidos como na questão anterior. Peça ao usuário para entrar com o horário de início e fim da sua aula de programação. Exiba quantos minutos de aula existem entre os dois horários, como no exemplo abaixo:

Início: **13:55**
Fim: **15:45**

Entre 13:55 e 15:45 existem 110 minutos, isto é, 1h e 50 minutos.

Sugestão: faça uma versão com funções normais e depois faça outra versão que utiliza os operadores para cin, cout e subtração no lugar das funções.