

**Lista de Exercícios 02**  
**Programando em C++**

Departamento de Computação  
Universidade Federal de Ouro Preto  
Prof. José Romildo Malaquias  
2015–1

Instruções

- Todos os exercícios que envolvem programas devem ser resolvidos por programas na linguagem C++.
- Na solução dos exercícios, devem ser utilizados os conceitos listados no cabeçalho desta lista.
- Eventuais dúvidas podem ser sanadas com o professor.

1. Escreva um programa que leia três números inteiros e imprima a soma, a média aritmética, o produto, o maior e o menor valor lido.
2. Escreva um programa que leia o raio de um círculo e imprima o diâmetro, a área e o comprimento da circunferência do círculo. Os resultados devem ser escritos com duas casas decimais.
3. Crie um programa que leia dois números inteiros e determine se um é múltiplo do outro.
4. Sabemos que `cout` não utiliza formatadores. Escreva um programa que lê um caractere e retorna seu código na tabela ASCII (um número inteiro). Pesquise por `static_cast`.
5. Escreva um programa que lê 15 números reais e através de uma função determine quais são os dois maiores deles, que não devem ser repetidos. Não utilize vetores e utilize **referências**.
6. Uma grande companhia paga seus vendedores com salários fixos mais comissões sobre as vendas. Um vendedor recebe R\$200,00 mais 9% de comissão sobre todas as suas vendas. Escreva um programa que lê o valor total das vendas de um vendedor e calcule seu salário. Seu programa deve computar o salário de diversos vendedores, e parar quando o valor informado for -1.
7. Escreva um programa que estima o valor da constante  $e$  pela fórmula

$$e = \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots$$

O usuário deve informar qual é o número de termos deste somatório, o que define a precisão do resultado.

8. Crie um `procedimento` que receba um número e determine o valor absoluto do mesmo. Utilize **referências**.
9. Crie um programa que leia 12 números inteiros em qualquer ordem e imprima uma listagem contendo o número lido e uma mensagem *PAR* ou *ÍMPAR* de acordo com o número. Utilize um vetor **alocado dinamicamente**, e depois **libere** a memória. Utilize os operadores `new` e `delete`.
10. Crie um programa que armazene 12 números em um vetor **alocado dinamicamente** e determine qual a porcentagem de números menores que 8 e qual a porcentagem de números maiores que 10. Utilize os operadores `new` e `delete`.
11. **Quadrado mágico**. Um quadrado mágico é uma tabela quadrada de lado  $n$ , onde a soma dos números das linhas, das colunas e das diagonais principal e secundária é constante, sendo que nenhum destes números se repete.

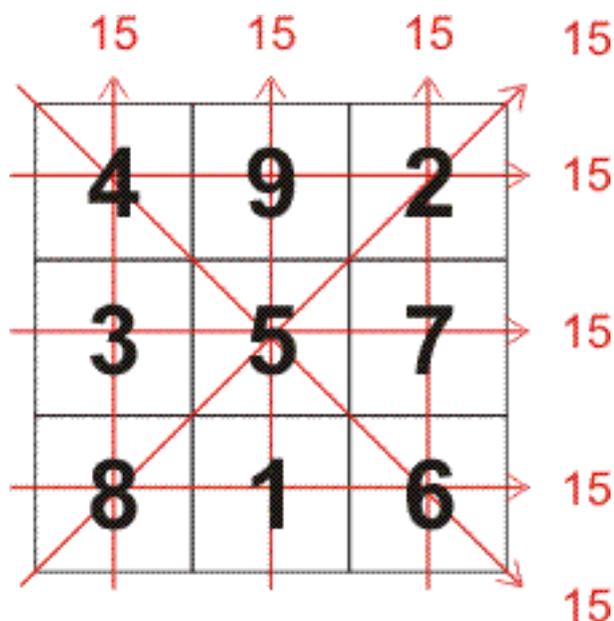


Figura 1: Exemplo de quadrado mágico.

Crie um programa para **verificar** (não confunda com resolver) se uma matriz realmente é um quadrado mágico. Utilize **alocação dinâmica** para criar a matriz e não se esqueça de liberar a memória ao final do programa. Utilize os operadores **new** e **delete**.

12. Crie um programa que **aloque dinamicamente** e leia uma matriz  $10 \times 10$  de caracteres e, através de uma função, imprima todos os elementos, **exceto os da diagonal secundária**. Libere a memória ao final do programa. Utilize os operadores **new** e **delete**.
13. Crie uma função soma para números inteiros (ex.: `soma(2, 4) = 6`). Sobrecarregue esta função para que também some números reais (ex.: `soma(4.5, 6.2) = 10,7`), vetores de caracteres (ex.: `soma("Universidade ", "Federal") = "Universidade Federal"`) e vetores de inteiros (ex.: `soma({1,2,3}, {4,5,6}) = {1,2,3,4,5,6}`). Não utilize a classe `string`. No entanto, é permitido utilizar a biblioteca `string.h`. No caso dos vetores, considere que o tamanho máximo de cada vetor será 10.