

**Universidade Federal de Minas Gerais**  
**Departamento de Engenharia Elétrica**  
**ELE078 – Programação Orientada a Objetos**

**Lista de Exercícios 1:**

**Assunto: Ambientação com a Linguagem Java/C++**

**Exercícios para Programadores em JAVA e em C++:**

1. Construa um programa que receba da linha de comando, com a qual o programa é executado, três valores em ponto flutuante, e informe, na saída padrão, se estes podem ou não formar os lados de um triângulo, e qual tipo de triângulo seria: Equilátero, Isósceles ou Escaleno.
2. Escreva um programa em que leia um conjunto de 4 valores i, a, b, c, onde i é um valor inteiro e positivo e a, b, c, são quaisquer valores reais e os escreva:
  - (a) Se i=1 escrever os três valores a, b, c em ordem crescente.
  - (b) Se i=2 escrever os três valores a, b, c em ordem decrescente.
  - (c) Se i=3 escrever os três valores a, b, c de forma que o maior entre a, b, c fique dentre os dois.
3. Desenvolva um método (função) que recebe como parâmetros um vetor de números inteiros e um número inteiro a ser pesquisado dentro do vetor. O método deve retornar “True” se o número pesquisado se encontra no vetor e “False”, caso contrário. Para testá-lo, crie um método main que execute os seguintes passos:
  - Cria e inicializa um vetor de 100 números inteiros aleatórios variando de 100 a 199.
  - Leia da entrada padrão vários números a serem pesquisados. Para cada um deles imprima um resultado dizendo se o número se encontra ou não no arranjo.
  - Imprima o vetor de números inteiros criado.
4. Faça um programa em que inicialize um vetor de 10 elementos inteiros com valores aleatórios entre 12 e 21. Em seguida, elimine todos os elementos repetidos e imprima apenas os elementos restantes.

**- Exemplo:**

Vetor gerado

12	14	18	14	12	13	14	14	21	17
----	----	----	----	----	----	----	----	----	----

Vetor resultante

12	14	18	13	21	17	?	?	?	?
----	----	----	----	----	----	---	---	---	---

O programa deve possuir três métodos:

- `int eliminaRepetidos (int vet[])` – elimina os elementos repetidos do vetor `vet`. Retorna o número de elementos restantes no vetor após a eliminação.

- `void imprime (int vet[], int numElem)` – imprime na saída padrão os elementos do vetor de inteiros `vet`. `numElem` é o número de elementos do vetor.
- `void main ()` – cria e inicializa o vetor, elimina os elementos repetidos através da chamada do método `eliminaRepetidos` e imprime o vetor resultante usando o método `imprime`.

5. Faça um programa que inicialize uma matriz de 10 linhas e 10 colunas de elementos inteiros com valores aleatórios entre 0 e 4. Em seguida, imprima os elementos da matriz e uma mensagem informando se a matriz é ou não simétrica. Uma matriz é simétrica se for quadrada (o número de linhas é igual ao número de colunas) e todos os elementos das posições  $i, j$  são iguais aos elementos das posições  $j, i$ , onde  $i$  representa o número da linha e  $j$  representa o número da coluna.

O programa deve possuir três métodos estáticos:

- Método `main`, para criar, inicializar e imprimir a matriz, e imprimir uma mensagem informando se a matriz é ou não simétrica. O método `main` deve usar os dois métodos definidos abaixo.
  - Método para imprimir uma matriz. Este método deve receber como parâmetro a matriz a ser impressa.
  - Método para verificar se uma matriz é simétrica. Este método deve receber como parâmetro a matriz a ser verificada e retornar `true` se a matriz é simétrica e `false`, caso contrário. O método não deve imprimir nenhuma mensagem.
6. Escreva um programa que simule a jogada de dois dados. O programa deve usar um método de geração de números aleatórios (ex. `rand()`) para lançar os dois dados. A soma dos valores obtidos pelos dois dados deve então ser calculada. Como cada dado pode mostrar um valor inteiro entre 1 e 6, a soma dos dois dados irá variar de 2 a 12, com 7 sendo a soma mais frequente e 2 e 12 sendo as somas menos frequentes. A Figura a seguir mostra as 36 combinações possíveis dos dois dados. Seu programa deve lançar os dados 34.000 vezes. Use um vetor unidimensional para registrar o número de vezes que é obtida cada soma possível. Imprima os resultados em formato de tabela. Além disso, determine se as somas são razoáveis, isto é, há seis maneiras de se obter 7, portanto, aproximadamente um sexto do total de jogadas deve ser 7.

	1	2	3	4	5	6
1	2	3	4	5	6	7
2	3	4	5	6	7	8
3	4	5	6	7	8	9
4	5	6	7	8	9	10
5	6	7	8	9	10	11
6	7	8	9	10	11	12

7. Faça um programa para embaralhar um baralho de 52 cartas. Após embaralhar, o programa deve mostrar todas as cartas embaralhadas, da primeira até a última. Para tanto, use uma matriz **baralho** de 4 x 13 para representar o baralho. As linhas devem corresponder aos naipes: a linha 0 corresponde a **copas**, a linha 1 a **ouros**, a linha 2 a **paus** e a linha 3 a **espadas**. As colunas correspondem aos valores de face das cartas: 0 até 9 correspondem a **às** até **dez**, respectivamente, e as colunas 10 até 12 correspondem a **valete**, **dama** e **rei**, respectivamente. Deve-se ter também um vetor **naipes**, do tipo **string**, representando os naipes e o vetor **faces**, também do tipo **string**, representando os treze valores das cartas. A Figura a seguir ilustra a matriz **baralho**.

O método de embaralhar deve proceder da seguinte forma: inicialmente, a matriz **baralho** é inicializada com zeros. Então, **uma linha e uma coluna** são escolhidas ao acaso. O número 1 é então inserido no **baralho** na posição **baralho[linha][coluna]** para indicar que esta carta vai ser a primeira a ser retirada do baralho (carta que fica por cima). Este procedimento continua com os números 2,3,4..., 52 sendo aleatoriamente inseridos na matriz **baralho** para indicar quais cartas vão ser colocadas na 2ª, 3ª, 4ª, ..., 52ª posição. À medida que a matriz **baralho** começa a ser preenchida com as posições das cartas, é possível que uma carta seja selecionada mais de uma vez. Se a carta já encontra-se posicionada (não-zero) então a nova seleção deve ser ignorada. Em algum momento, os números 1 a 52 ocuparão todas as posições da matriz **baralho**. Neste momento, o processo de embaralhamento estará completo. Após o término do procedimento de embaralhar as cartas, o programa deve mostrar as cartas da 1ª até a 52ª. A saída do programa deve ser similar a:

“seis de paus”  
“às de espadas”  
“dama de paus”  
“sete de ouros”  
etc.

## Exercícios para Programadores JAVA:

8. O programa abaixo recebe valores numéricos passados através dos argumentos de chamada do método *main* e imprime na saída padrão o maior valor fornecido. Altere-o para imprimir também o menor valor, a soma e a média aritmética dos valores fornecidos.

```
public class MaiorNumero {  
  
    public static void main (String[] args){  
  
        if (args.length > 0) {  
            double valor;  
            double maior = Double.parseDouble(args[0]);  
  
            for (int i=1; i < args.length; i++) {  
                valor = Double.parseDouble(args[i]);  
                if (valor > maior)  
                    maior = valor;  
            }  
            System.out.println("Maior valor = " + maior);  
        }  
    }  
}
```

```

    }
    else System.out.println("Entre com os valores na
forma: java MaiorNumero n1  n2 n3 ...");
    }
}

```

9. Mostre qual o resultado será impresso pelo programa se ele for executado.

```

public class Parametro {
    public static void main(String[] args) {
        int[] v = new int[4];
        v[0] = (int)5.25;
        v[1] = 1;
        v[2] = 2;
        v[3] = 3;
        troca1(v[0],v[1]);
        imprime(v);
        troca2(v);
        imprime(v);
    }
    public static void troca1(int a, int b) {
        int aux = a;
        a = b;
        b = aux;
    }
    public static void troca2(int[] v) {
        int aux;
        for (int i=0; i<v.length/2; i++) {
            aux = v[i];
            v[i] = v[v.length-i-1];
            v[v.length-i-1] = aux;
        }
    }
    public static void imprime(int[] v) {
        for(int i=0; i<v.length; i++)
            System.out.print(v[i]+" ");
    }
}

```

## Exercícios para Programadores C++:

10. O programa abaixo recebe valores numéricos passados através dos argumentos de chamada do método *main* e imprime na saída padrão o maior valor fornecido. Altere-o para imprimir também o menor valor, a soma e a média aritmética dos valores fornecidos.

```

// program maiorNumero.cpp

int main(int argc, char*  argv[]){

    if argc > 0 {
        double valor;
        double maior = atof(argv[1]);

```

```

        for (int i=2; i < argc; i++) {
            valor = atof(argv[i]);
            if (valor > maior)
                maior = valor;
        }
        cout << "Maior valor = " << maior << endl;
    }
    else cout << "Entre com os valores na forma:
maiorNumero.exe n1 n2 n3 ..." << endl;

}

```

11. Mostre qual o resultado será impresso pelo programa se ele for executado.

```

void troca1(int a, int b) {
    int aux = a;
    a = b;
    b = aux;
}

void troca2(int* v, int length) {
    int aux;

    for (int i=0; i< length/2; i++) {
        aux = v[i];
        v[i] = v[length-i-1];
        v[length-i-1] = aux;
    }
}

void imprime(int* v int length) {

    for(int i=0; i<length; i++)
        cout << "v[" << i << "] = " << v[i] << endl;
}

int main() {
    int v[4];
    v[0] = (int)5.25;
    v[1] = 1;
    v[2] = 2;
    v[3] = 3;
    troca1(v[0],v[1]);
    int length = sizeof(v)/sizeof(v[0]);
    imprime(v, length);
    troca2(v);
    imprime(v);
}

```

12. Seja p uma variável do tipo ponteiro, explique a diferença entre  
p++; (\*p)++; \*(p++); \*p++; \*(p+10);

13. Seja a[ ] um vetor qualquer, independente de tipo e tamanho, e pa um ponteiro

para o mesmo tipo de `a[ ]`. Responda V ou F, justificando:

- a) Após a atribuição `pa = &a[0]`; `pa` e `a` possuem valores idênticos, isto é, apontam para o mesmo endereço .
- b) A atribuição `pa = &a[0]`; pode ser escrita como `pa = a`;
- c) `a[i]` pode ser escrito como `*(a+i)`
- d) `&a[i]` e `a+i` são idênticos
- e) `a+i` é o endereço do *i*-ésimo elemento após `a`
- f) `pa[i]` é idêntico a `*(pa+i)`
- g) `pa=a` é uma operação válida
- h) `pa++` é uma operação válida
- i) `a=pa` é uma operação válida
- j) `a++` é uma operação válida