

Lista 4 – Introdução à POO (INE5603) – 2017s2
Sistemas de Informação – Universidade Federal de Santa Catarina

1. Escreva um programa que leia um valor n e depois leia uma sequência de n números inteiros e os armazene em um vetor. Posteriormente, seu programa deve determinar o maior e menor elemento deste vetor.
2. Escreva um programa que inverta um vetor v de n , isto é, troque o valor $v[0]$ com $v[n-1]$, $v[1]$ com $v[n-2]$, e assim por diante.
3. Escreva um programa que leia uma palavra pelo teclado (colocando-a em um String, conforme o exemplo abaixo) e determine se é um palíndromo.

```
import java.util.Scanner;
public class Teste
{
    public void exemplo()
    {
        System.out.print("Informe a palavra: ");
        Scanner ler = new Scanner(System.in);
        String s = ler.next();
        System.out.println("\nPalavra informada: " + s);
    }
}
```

4. Escreva um programa que leia uma frase pelo teclado (para isto, troque o `next`, do programa do exercício anterior, por `nextLine`). Entre as palavras que compõem a frase pode haver um ou mais espaços em branco ou sinais de pontuação “,” e “.”. O programa deve escrever, na tela, cada palavra da frase em uma linha distinta.
5. Implemente os seguintes métodos estatísticos sobre um vetor de inteiros:

x_0	x_1	x_2	x_3	x_4	\dots	x_{n-1}
0	1	2	3	4		$n-1$

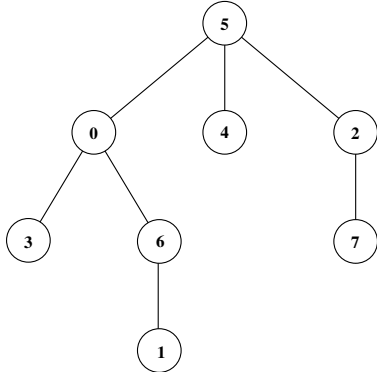
- Média (\bar{x})

$$\bar{x} = \frac{1}{n} \sum_{i=0}^{n-1} x_i = \frac{1}{n} (x_0 + x_1 + x_2 + \dots + x_{n-1})$$

- Desvio padrão (s)

$$s = \sqrt{\frac{1}{n-1} \sum_{i=0}^{n-1} (x_i - \bar{x})^2}$$

6. Em uma empresa, trabalham n pessoas, e cada uma recebe um número de identificação único, de 0 a $n - 1$. A empresa encontra-se organizada em níveis hierárquico, de forma que cada pessoa possui exatamente um chefe (exceto o presidente). A figura abaixo representa a hierarquia de uma empresa com 8 pessoas. Por exemplo, a pessoa 3 tem como chefe a pessoa 0. Por sua vez, a pessoa 0 tem como chefe a pessoa 5. A pessoa 5 é a única que não possui chefe, sendo, portanto, o presidente.



As informações de chefia podem ser resumidas em tabelas, como exemplificado abaixo. Note que o “chefe do presidente” está representado por -1.

Pessoa	0	1	2	3	4	5	6	7
Chefe	5	6	5	0	5	-1	0	2

Com as informações de chefia, pode-se determinar o nível de cada pessoa na empresa, sendo o presidente colocado no nível 1, as pessoas logo abaixo (0, 4 e 2), no nível 2, as pessoas abaixo delas (3, 6 e 7), no nível 3, e assim por diante. De forma resumida, tem-se:

Pessoa	0	1	2	3	4	5	6	7
Nível	2	4	2	3	2	1	3	3

Sua tarefa é determinar o nível de cada pessoa na empresa. Para isto, a quantidade n de pessoas deve ser lida, seguido de n valores inteiros positivos, indicando os chefes de cada pessoa, em ordem crescente de número de identificação, como na segunda linha da primeira tabela. A saída de seu programa deve gerar o vetor correspondente à segunda tabela, exibindo o nível de cada pessoa na tela.

7. Implemente um método que conte quantos valores iguais a x (passado como parâmetro) há em um vetor de inteiros.
8. Considerando que um vetor *vet* contém inteiros em um intervalo de 0 a $L - 1$, construa um vetor de saída, chamado de histograma *hist*, de L posições, onde, em cada posição i , seja calculada a quantidade de vezes que o valor i ocorre em *vet*. Segue um exemplo de histograma:

- Vetor de entrada (*vet*), considerando $L = 4$ (ou seja, os valores possíveis vão de 0 a 3):

3	2	2	0	3	0	2
0	1	2	3	4	5	6

- Vetor de entrada (*hist*) a ser produzido:

2	0	3	2
0	1	2	3

9. Uma imagem binária (preto e branco), de tamanho $m \times n$, pode ser representada por uma matriz *img*, na qual cada elemento assume ou valor 0 ou valor 1. Dado um padrão *pad*, representado por uma matriz 3×3 , também binária, escreva um programa que determine se tal padrão existe ou não na imagem.

```

int img[][] = { {0, 0, 1, 0, 0, 0, 0, 1, 0},
                {0, 1, 1, 1, 1, 1, 1, 0, 0},
                {0, 0, 1, 0, 1, 1, 1, 0, 0},
                {0, 0, 0, 0, 0, 0, 0, 0, 0},
                {0, 0, 0, 0, 1, 0, 1, 0, 0},
                {0, 0, 0, 0, 0, 1, 0, 0, 0},
                {1, 1, 1, 0, 1, 0, 1, 0, 0},
                {1, 0, 1, 0, 0, 0, 0, 0, 0},
                {1, 1, 1, 0, 0, 1, 1, 1, 0} };

int pad[][] = { {0, 1, 0},
                {1, 1, 1},
                {0, 1, 0} };

```

10. Dadas duas sequências de n e m elementos binários (0 ou 1), onde $n \leq m$, escreva um programa que verifique quantas vezes a primeira sequência ocorre na segunda. Exemplo: se a primeira for 101 e a segunda 1101010011010, então o resultado deve ser 3.
11. **Sudoku** é um quebra-cabeça, cujo objetivo do jogo é preencher os números de 1 a 9 em cada uma das células vazias em uma grade 9×9 , constituída por 3×3 subgrades chamadas regiões. Cada coluna, linha e região só pode ter um número de cada um dos números de 1 a 9. Exemplo de um jogo, à esquerda, e sua resolução, à direita (imagens obtidas do Wikipédia):

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

Escreva um programa em Java que, dado um jogo Sudoku, representado por uma classe que contém uma matriz 9×9 , verifica se o jogo está ou não correto.

12. Pesquise e implemente um método simples (*bubble sort*, *selection sort*, ...) de ordenação em vetor. Teste seu programa para um vetor de inteiros qualquer.
13. Implemente os seguintes métodos estatísticos sobre um vetor de inteiros:

x_0	x_1	x_2	x_3	x_4	...	x_{n-1}
0	1	2	3	4		$n-1$

- Moda
 - Valor mais frequente
- Mediana
 - Valor do meio, considerando a sequência ordenada

14. Sobre busca sequencial e binária:

- Escreva um programa que leia uma sequência de números positivos e os armazene em um vetor. Feito isto, seu programa deve solicitar um número ao usuário e pesquisar se o número digitado está no vetor por meio de busca sequencial.
- Escreva um programa semelhante ao anterior, mas utilizando busca binária (é necessária a ordenação prévia).
- Compare o desempenho de ambos programas, considerando o número de comparações efetuado, em cada caso, no processo de pesquisa de um número dado.

15. Escreva um programa que leia duas palavras do teclado e determina se a segunda é um anagrama da primeira. Uma palavra é um anagrama de outra se todas as letras de uma ocorrem na outra, em mesmo número, independente da posição. Exemplos: ROMA, MORA, ORAM, AMOR, RAMO são anagramas entre si.