



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**  
**AD2 de Programação OO**  
**1º semestre de 2017**

**EXERCÍCIO (ENTREGAR OS ARQUIVOS EM MÍDIA, PARA FINS DE TESTE, JUNTAMENTE COM A AD IMPRESSA):**

Supondo que você tenha uma empresa de software, e que a FEX contrate sua empresa para escrever um programa que informe, automaticamente, os confrontos das semi-finais do campeonato do Estado X. Esses times são determinados a partir do número de pontos obtidos, e caso os times tenham o mesmo número de pontos, o critério de desempate é o número de vitórias, seguido do saldo de gols (o saldo de gols é dado pela diferença entre os gols feitos e os gols sofridos). Os quatro primeiros times deverão fazer as semi-finais, e os confrontos devem ser o primeiro contra o quatro classificados, e o segundo contra o terceiro melhores ordenados. Os dados de entrada são as datas e os placares dos jogos. Um time que ganha um jogo recebe três pontos e um time que perde não ganha pontos. Já quando os times empatam, eles recebem um ponto cada. Para o exemplo do arquivo que segue:

02.04. 16:00/Boavista/Portuguesa-RJ/1 : 1  
02.04. 16:00/Botafogo/Resende/3 : 2  
02.04. 16:00/Fluminense/Flamengo/1 : 1  
02.04. 16:00/Macae/Volta Redonda/2 : 2  
02.04. 16:00/Madureira/Bangu/3 : 1  
02.04. 16:00/Nova Iguaçu/Vasco/0 : 2  
30.03. 21:30/Vasco/Boavista/1 : 0  
30.03. 19:30/Portuguesa-RJ/Botafogo/1 : 4  
30.03. 15:30/Bangu/Macae/2 : 0  
29.03. 21:45/Volta Redonda/Flamengo/1 : 1  
29.03. 16:00/Madureira/Fluminense/2 : 2  
29.03. 15:30/Resende/Nova Iguaçu/0 : 1  
26.03. 19:00/Fluminense/Macae/3 : 0  
26.03. 18:30/Flamengo/Vasco/2 : 2  
26.03. 16:00/Bangu/Botafogo/0 : 2  
26.03. 15:30/Resende/Boavista/0 : 1  
25.03. 18:30/Volta Redonda/Nova Iguaçu/2 : 1  
25.03. 15:30/Portuguesa-RJ/Madureira/2 : 0  
23.03. 21:45/Botafogo/Fluminense/2 : 3  
23.03. 15:30/Macae/Resende/1 : 2  
22.03. 21:45/Flamengo/Bangu/3 : 0  
22.03. 19:30/Vasco/Madureira/1 : 0  
22.03. 17:00/Boavista/Volta Redonda/1 : 0  
22.03. 15:30/Nova Iguaçu/Portuguesa-RJ/0 : 0

19.03. 19:00/Volta Redonda/Madureira/3 : 2  
19.03. 18:30/Vasco/Botafogo/0 : 0  
19.03. 16:00/Fluminense/Nova Iguaçu/1 : 3  
19.03. 15:30/Portuguesa-RJ/Macae/2 : 1  
18.03. 18:30/Resende/Flamengo/0 : 1  
18.03. 15:30/Bangu/Boavista/0 : 0  
12.03. 19:00/Boavista/Fluminense/0 : 2  
12.03. 16:00/Macae/Vasco/2 : 2  
12.03. 15:30/Madureira/Resende/1 : 1  
11.03. 18:30/Flamengo/Portuguesa-RJ/5 : 1  
11.03. 15:30/Nova Iguaçu/Bangu/2 : 1

Os jogos das semi-finais serão os seguintes:

**Flamengo X Botafogo**

**Vasco X Fluminense**

**LEMBRE-SE: SEU PROGRAMA DEVE EXECUTAR COM QUAISQUER DADOS INFORMADOS COMO PARÂMETROS DE ENTRADA. SE O SEU PROGRAMA RESOLVER SOMENTE O PROBLEMA DO EXERCÍCIO SUPRACITADO, SUA QUESTÃO SERÁ TOTALMENTE DESCONTADA.**

**RESPOSTA:**

```
import java.io.*;
```

```
class no{
    String nome;
    int pontos, gols_favor, gols_sof, vit;
    no prox;

    no(String nome){
        this.nome = nome;
        pontos = vit = gols_favor = gols_sof = 0;
        prox = null;
    }

    public String toString(){ return nome; }
}
```

```
class lista{
    no prim;

    lista(){ prim = null; }

    no busca(String nome){
        no p = prim;
        while((p != null) && (nome.compareToIgnoreCase(p.nome) != 0))
            p = p.prox;
        return p;
    }
}
```

```

void insere(String nome){
    no novo = new no(nome);
    if(prim == null)
        prim = novo;
    else{
        no p = prim;
        while(p != null){
            if(nome.compareToIgnoreCase(p.nome) == 0) return;
            p = p.prox;
        }
        novo.prox = prim;
        prim = novo;
    }
}

void insere(String nome, int gf, int gc){
    no p = busca(nome);
    if(p == null) return;
    p.gols_favor += gf;
    p.gols_sof += gc;
    if(gf > gc){ p.vit++; p.pontos += 3; }
    if(gf == gc) p.pontos += 1;
}

public String toString(){
    String resp = "";
    no p = prim;
    while (p != null){
        resp += p.toString() + " " + p.pontos + " " + p.vit + " " +
p.gols_favor + " " + p.gols_sof + "\n";
        p = p.prox;
    }
    return resp;
}
}

public class AD2_P00_2017_1{
    public static void main(String[] args) throws IOException {
        BufferedReader in = new BufferedReader(new FileReader(args[0]));
        String s;
        lista l = new lista();
        try{
            while((s = in.readLine()) != null){
                String vs[] = s.split("/");
                l.insere(vs[1]);
                l.insere(vs[2]);
                String placar[] = vs[3].split(" : ");
                l.insere(vs[1], Integer.parseInt(placar[0]),
Integer.parseInt(placar[1]));
                l.insere(vs[2], Integer.parseInt(placar[1]),
Integer.parseInt(placar[0]));
            }
        }catch (Exception e){
            System.out.println("Excecao leitura\n");
        }finally{ in.close(); }
    }
}

```

```

        Ordena(l);
        System.out.println(imprimeFinalistas(l));
    }

    static void Ordena(lista l){
        if(l.prim != null){
            for(no p = l.prim; p != null; p = p.prox){
                no maior = p;
                for(no q = p.prox; q != null; q = q.prox)
                    if (maiorEq(maior, q)) maior = q;
                if(maior != p) troca(maior, p);
            }
        }
    }

    static boolean maiorEq (no maior, no q){
        if(q.pontos > maior.pontos) return true;
        if((q.pontos == maior.pontos) && (q.vit > maior.vit)) return true;
        int saldo_maior = maior.gols_favor - maior.gols_sof;
        int saldo_q = q.gols_favor - q.gols_sof;
        if((q.pontos == maior.pontos) && (q.vit == maior.vit) && (saldo_q >
saldo_maior)) return true;
        return false;
    }

    static void troca(no maior, no p){
        String nome_aux = maior.nome; //troca nome
        maior.nome = p.nome;
        p.nome = nome_aux;
        int temp = maior.pontos; //troca pontos
        maior.pontos = p.pontos;
        p.pontos = temp;
        temp = maior.vit; //troca numero de vitorias
        maior.vit = p.vit;
        p.vit = temp;
        temp = maior.gols_favor; //troca gols a favor
        maior.gols_favor = p.gols_favor;
        p.gols_favor = temp;
        temp = maior.gols_sof; //troca gols sofridos
        maior.gols_sof = p.gols_sof;
        p.gols_sof = temp;
    }

    static String imprimeFinalistas(lista l){
        String resp = l.prim.toString() + " X ";
        resp += l.prim.prox.prox.prox.toString() + "\n";
        resp += l.prim.prox.toString() + " X ";
        resp += l.prim.prox.prox.toString();
        return resp;
    }
}

```