



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**

**Disciplina: Programação III**

**AP3 1º semestre de 2011.**

**Nome –**

**Assinatura –**

---

Observações:

1. Prova sem consulta e sem uso de máquina de calcular.
  2. Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
  3. Você pode usar lápis para responder as questões.
  4. Ao final da prova devolva as folhas de questões e as de respostas.
  5. Todas as respostas devem ser transcritas nas folhas de respostas. As respostas nas folhas de questões não serão corrigidas.
- 

**Questão 1) (5.0 pontos)**

Supondo que você tenha uma empresa de software, e que a Federação Internacional de Voleibol contrate sua empresa para escrever um programa que informe, automaticamente, os nomes dos semifinalistas do Campeonato Mundial de Voleibol. Esta competição é dividida em pelo menos duas fases: uma preliminar e uma final. Na fase preliminar, os times são organizados em duas chaves. Cada time realiza uma partida contra todos os outros times de sua chave. Quando todas as partidas da fase preliminar foram disputadas, os dois melhores times em cada chave qualificam-se para a fase final. As semifinais devem ser disputadas em cruzamento olímpico, isto é, a equipe melhor classificada de um grupo joga contra a segunda equipe melhor classificada do outro grupo.

Para classificar as equipes dentro de um grupo será levado em consideração, em primeiro lugar, o número de vitórias conquistadas por cada equipe. Equipes com mesmo número de vitórias terão, como critério de desempate, a maior diferença entre sets ganhos e sets perdidos.

Os dados de entrada são o número de times em cada grupo e os placares dos jogos. Para o exemplo do arquivo que segue:

**4**

**Brasil/3x0/Tunisia**

**Espanha/2x3/Cuba**

**Tunisia/0x3/Cuba**

**Brasil/3x1/Espanha**

**Espanha/3x1/Tunisia**

**Brasil/2x3/Cuba**

**Polonia/3x0/Canada**

Alemanha/0x3/Servia  
Canada/3x1/Servia  
Polonia/3x2/Alemanha  
Alemanha/3x1/Canada  
Servia/1x3/Polonia

O seu software deve informar os seguintes jogos de semifinal:

Polonia x Brasil  
Cuba x Servia

## RESPOSTA:

```
import java.io.*;

class time{
    String nome;
    int win, num_vit, loss;
    time (String n){
        nome = n;
        num_vit = win = loss = 0;
    }
    public String toString(){ return nome; }
}

public class teste{
    public static void Ordena(time[] chave){
        int i, j;
        time temp;
        for(i = 0; i < chave.length; i++){
            for(j = i + 1; j < chave.length; j++){
                if((chave[i].num_vit < chave[j].num_vit) ||
                    ((chave[i].num_vit == chave[j].num_vit) &&
                     ((chave[i].win - chave[i].loss) < (chave[j].win - chave[j].loss))))
                {
                    temp = chave[i];
                    chave[i] = chave[j];
                    chave[j] = temp;
                }
            }
        }
    }

    public static boolean busca (time[] chave, String n){
        for(int i = 0; chave[i] != null; i++)
            if(n.equals(chave[i].nome)) return true;
        return false;
    }

    public static void insere(time[] chave, String n, int win, int loss){
        boolean r = busca(chave, n);
        int i;
        if(r){
            for(i = 0; !n.equals(chave[i].nome); i++);
            chave[i].win += win;
            chave[i].loss += loss;
            if(win > loss) chave[i].num_vit++;
        }else{
```

```

        for(i = 0; chavez[i] != null; i++);
        time q = new time(n);
        q.win = win;
        q.loss = loss;
        if(win > loss) q.num_vit = 1;
        chavez[i] = q;
    }
}

public static void main(String[] args) throws IOException {
    BufferedReader in = new BufferedReader(new FileReader(args[0]));
    time chavez1[], chavez2[];
    try{
        String s = in.readLine();
        int n = Integer.parseInt(s);
        chavez1 = new time[n];
        int total = (n * (n - 1)) / 2;
        String vs[], placar[];
        int aux_g1, aux_g2;
        while (total > 0){
            total--;
            s = in.readLine();
            vs = s.split("/");
            placar = vs[1].split("x");
            aux_g1 = Integer.parseInt(placar[0]);
            aux_g2 = Integer.parseInt(placar[1]);
            insere(chavez1, vs[0], aux_g1, aux_g2);
            insere(chavez1, vs[2], aux_g2, aux_g1);
        }
        chavez2 = new time[n];
        total = (n * (n - 1)) / 2;
        while (total > 0){
            total--;
            s = in.readLine();
            vs = s.split("/");
            placar = vs[1].split("x");
            aux_g1 = Integer.parseInt(placar[0]);
            aux_g2 = Integer.parseInt(placar[1]);
            insere(chavez2, vs[0], aux_g1, aux_g2);
            insere(chavez2, vs[2], aux_g2, aux_g1);
        }
        in.close();
        Ordena(chavez1);
        Ordena(chavez2);
        System.out.println(chavez1[0].toString() + " x " +
chavez2[1].toString());
        System.out.println(chavez2[0].toString() + " x " +
chavez1[1].toString());
    }catch (Exception e){
        System.out.println(e);
    }finally{
        in.close();
    }
}
}

```

## Questão 2) (5.0 pontos)

Suponha que queiramos manipular polígonos no plano 2D, os quais são compostos de um conjunto de pontos (coordenadas x e y). Uma operação que se deseja realizar sobre polígonos é o cálculo do seu perímetro (soma dos lados). Resolva este problema de forma fragmentada, ou seja, crie um método para cálculo da distância entre 2 pontos e, depois, para a soma de todos os lados do polígono. Implemente também uma operação para o cálculo do “centróide”, a qual retorna um ponto no plano 2D cujas coordenadas são as médias das respectivas coordenadas dos pontos do polígono.

### RESPOSTA:

```
package br.cederj.comp.ano2011;

import java.util.ArrayList;
import java.util.List;

/**
 * Classe que representa um ponto
 * @author carlosbazilio
 */

class Ponto {
    protected double x, y;

    public Ponto(double x, double y) {
        this.x = x;
        this.y = y;
    }

    // Método que calcula a distância entre 2 pontos
    public double getDistancia (Ponto p) {
        return Math.sqrt(Math.pow(p.x - this.x, 2) + Math.pow(p.y -
this.y, 2));
    }
}

/**
 * Classe que representa um poligono, ou seja, uma lista de pontos
 */

class Poligono {
    private List<Ponto> pontos;

    public Poligono() {
        this.pontos = new ArrayList<Ponto>();
    }

    public Poligono(List<Ponto> pontos) {
        this.pontos = pontos;
    }

    public int getNumPontos() {
        return pontos.size();
    }
}
```

```

    }

    // Obtém o perímetro do polígono
    public double getPerimetro() {
        double perimetro = 0;
        int pos = 0;
        while (pos < pontos.size() - 1) {
            Ponto p = pontos.get(pos);
            Ponto p2 = pontos.get(pos + 1);
            perimetro += p.getDistancia(p2);
            pos++;
        }
        Ponto p = pontos.get(0);
        return perimetro + p.getDistancia(pontos.get(pontos.size() - 1));
    }

    // Obtém o centróide que é o ponto médio das coordenadas do polígono
    public Ponto getCentroide() { // 2D
        Ponto retorno = null;
        double x = 0, y = 0;
        // Percorre os pontos do polígono
        for (Ponto paux : pontos) {
            x += paux.x;
            y += paux.y;
        }
        // Calcula a média para cada dimensão do ponto
        retorno = new Ponto(x / this.getNumPontos(), y /
this.getNumPontos());
        return retorno;
    }

    public void adicionaPonto (Ponto p) {
        pontos.add(p);
    }
}

/**
 * Classe de teste para a aplicação (NÃO FOI PEDIDO NA QUESTÃO)
 */

public class AP3_2011_1_Q2 {
    public static void main(String[] args) {
        Poligono figura = new Poligono();
        figura.adicionaPonto(new Ponto(0,0));
        figura.adicionaPonto(new Ponto(0,1));
        figura.adicionaPonto(new Ponto(1,1));
        figura.adicionaPonto(new Ponto(1,0));
        System.out.println("Perimetro da figura: " +
figura.getPerimetro());
        System.out.println("Centroide da figura: (" +
figura.getCentroide().x + ", " + figura.getCentroide().y + ")");
    }
}

```