



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**

**Disciplina: Programação III**

**AP3 1º semestre de 2013.**

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)**

Suponha que a CBV (Confederação Brasileira de Vôlei) tenha contratado a sua empresa de software para desenvolver um algoritmo que dado um arquivo de entrada, com todos os jogos do primeiro turno da Superliga masculina, retorne a classificação, em ordem decrescente, deste turno. Sabe-se, analisando as regulamentações da CBV que, em caso de:

VITÓRIA ou por 3x0, ou por 3x1 – o vencedor ganha três pontos;  
DERROTA ou por 0x3, ou por 1x3 – o perdedor não ganha nenhum ponto;  
VITÓRIA por 3x2 – o vencedor ganha dois pontos; e  
DERROTA por 2x3 – o perdedor ganha um ponto.

Se times estiverem empatados em pontuação, o critério de desempate é o número de vitórias. O seu programa deve retornar, no arquivo de saída **“class”** acrescido do nome de arquivo de entrada, a classificação do turno.

Por exemplo, dados os resultados do primeiro turno deste ano (Liga 2012-2013), cujo o nome é **result-2012-2013.txt**:

**SAO BERNARDO VOLEI#3x2#SESI-SP**  
**VIVO/MINAS#0x3#SADA CRUZEIRO VOLEI**  
**FUNVIC/MIDIA FONE#2x3#VOLTA REDONDA**  
**CANOAS#3x1#SUPER IMPERATRIZ VOLEI**  
**UFJF#0x3#RJX**

MEDLEY/CAMPINAS#3x0#VOLEI FUTURO  
VIVO/MINAS#3x2#SUPER IMPERATRIZ VOLEI  
MEDLEY/CAMPINAS#3x2#SESI-SP  
SADA CRUZEIRO VOLEI#3x0#CANOAS  
VOLEI FUTURO#3x2#SAO BERNARDO VOLEI  
RJX#3x0#FUNVIC/MIDIA FONE  
UFJF#2x3#VOLTA REDONDA  
SADA CRUZEIRO VOLEI#3x2#SUPER IMPERATRIZ VOLEI  
MEDLEY/CAMPINAS#1x3#SAO BERNARDO VOLEI  
VOLTA REDONDA#0x3#RJX  
UFJF#1x3#FUNVIC/MIDIA FONE  
VOLEI FUTURO#3x1#SESI-SP  
VIVO/MINAS#3x1#CANOAS  
VOLTA REDONDA#1x3#SADA CRUZEIRO VOLEI  
FUNVIC/MIDIA FONE#2x3#VIVO/MINAS  
SESI-SP#3x0#UFJF  
SAO BERNARDO VOLEI#1x3#RJX  
CANOAS#3x1#MEDLEY/CAMPINAS  
SUPER IMPERATRIZ VOLEI#3x0#VOLEI FUTURO  
FUNVIC/MIDIA FONE#0x3#SADA CRUZEIRO VOLEI  
VOLTA REDONDA#3x1#VIVO/MINAS  
SAO BERNARDO VOLEI#3x1#UFJF  
CANOAS#3x0#VOLEI FUTURO  
SUPER IMPERATRIZ VOLEI#0x3#MEDLEY/CAMPINAS  
SESI-SP#2x3#RJX  
RJX#3x1#VOLEI FUTURO  
VIVO/MINAS#3x1#SAO BERNARDO VOLEI  
VOLTA REDONDA#3x0#SUPER IMPERATRIZ VOLEI  
UFJF#2x3#MEDLEY/CAMPINAS  
FUNVIC/MIDIA FONE#2x3#CANOAS  
SADA CRUZEIRO VOLEI#0x3#SESI-SP  
SADA CRUZEIRO VOLEI#3x0#SAO BERNARDO VOLEI  
VIVO/MINAS#1x3#SESI-SP  
VOLTA REDONDA#1x3#CANOAS  
UFJF#3x0#VOLEI FUTURO  
MEDLEY/CAMPINAS#3x2#RJX  
FUNVIC/MIDIA FONE#2x3#SUPER IMPERATRIZ VOLEI  
MEDLEY/CAMPINAS#3x2#VOLTA REDONDA  
SESI-SP#3x0#CANOAS  
UFJF#0x3#SADA CRUZEIRO VOLEI  
VOLEI FUTURO#3x1#FUNVIC/MIDIA FONE  
SUPER IMPERATRIZ VOLEI#3x2#SAO BERNARDO VOLEI  
RJX#3x1#VIVO/MINAS  
VOLEI FUTURO#3x1#VOLTA REDONDA  
UFJF#1x3#VIVO/MINAS

**RJX#3x2#SADA CRUZEIRO VOLEI**  
**MEDLEY/CAMPINAS#3x0#FUNVIC/MIDIA FONE**  
**SUPER IMPERATRIZ VOLEI#1x3#SESI-SP**  
**SAO BERNARDO VOLEI#0x3#CANOAS**  
**SAO BERNARDO VOLEI#3x1#VOLTA REDONDA**  
**SESI-SP#3x1#FUNVIC/MIDIA FONE**  
**VIVO/MINAS#3x2#VOLEI FUTURO**  
**SADA CRUZEIRO VOLEI#0x3#MEDLEY/CAMPINAS**  
**CANOAS#3x0#RJX**  
**SUPER IMPERATRIZ VOLEI#0x3#UFJF**  
**SESI-SP#3x1#VOLTA REDONDA**  
**SAO BERNARDO VOLEI#3x0#FUNVIC/MIDIA FONE**  
**SADA CRUZEIRO VOLEI#3x1#VOLEI FUTURO**  
**VIVO/MINAS#3x0#MEDLEY/CAMPINAS**  
**CANOAS#3x0#UFJF**  
**SUPER IMPERATRIZ VOLEI#2x3#RJX**

A saída (no arquivo **class-result-2012-2013.txt**) será:

POS	NOME	PTOS	V	D	
1	RJX	25	9	2	
2	SADA CRUZEIRO	24	8	3	
3	SESI-SP	24	7	4	
4	CANOAS	23	8	3	
5	MEDLEY/CAMPINAS		20	8	3
6	VIVO/MINAS	18	7	4	
7	SAO BERNARDO VOLEI	16	5	6	
8	VOLEI FUTURO	12	4	7	
9	VOLTA REDONDA	11	4	7	
10	SUPER IMPERATIZ VOLEI	10	3	8	
11	UFJF	8	2	9	
12	FUNVIC/MIDIA FONE		7	1	10

**RESPOSTA:**

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

```

class no{
    String nome;
    int pos;
    int vit, der, ptos;
    no prox;

    no(String nome){
        this.nome = nome;
        ptos = der = vit = pos = 0;
        prox = null;
    }
}

```

```
public String toString(){
    return pos + "\t" + nome + "\t" + ptos + "\t" + vit + "\t" + der + "\n";
}
}
```

```
class lista{
    no prim;

    lista(){ prim = null; }

    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;
        }
    }

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

    void insereVitoria(String nome, int pontos){
        no p = busca(nome);
        if(p != null){
            p.vit++;
            p.ptos += pontos;
        }
    }

    void insereDerrota(String nome, int pontos){
        no p = busca(nome);
        if(p != null){
            p.der++;
        }
    }
}
```

```

        p.ptos += pontos;
    }
}

```

```

public String toString(){
    String resp = "";
    no p = prim;
    while(p != null){
        resp += p.toString();
        p = p.prox;
    }
    return resp;
}
}

```

```

public class Q1_AP3_2013_1{
    public static void main(String[] args) throws IOException {
        BufferedReader in = new BufferedReader(new FileReader(args[0]));
        String s;
        lista l = new lista();

        try{
            s = in.readLine();
            while(s != null){
                String vs[] = s.split("#");
                l.insere(vs[0]);
                l.insere(vs[2]);
                String placar[] = vs[1].split("x");
                if(placar[0].compareTo("3") == 0){
                    if(placar[1].compareTo("2") == 0){
                        l.insereVitoria(vs[0], 2);
                        l.insereDerrota(vs[2], 1);
                    }else{
                        l.insereVitoria(vs[0], 3);
                        l.insereDerrota(vs[2], 0);
                    }
                }else{
                    if(placar[0].compareTo("2") == 0){
                        l.insereVitoria(vs[2], 2);
                        l.insereDerrota(vs[0], 1);
                    }else{
                        l.insereVitoria(vs[2], 3);
                        l.insereDerrota(vs[0], 0);
                    }
                }
            }
        }
    }
}

```

```

    }
    s = in.readLine();
}
}catch (Exception e){
    System.out.println("Excecao leitura\n");
}finally{
    in.close();
}
}
Ordena(l);
BufferedWriter out = new BufferedWriter(new FileWriter("class-" + args[0]));
try{
    out.write("POS\tNOME\tPTOS\tV\tD\t\n");
    out.write(l.toString());
}catch (Exception e){
    System.out.println("Excecao de escrita\n");
}finally{
    out.close();
}
}
}

```

```

static void Ordena(lista l){
    if(l.prim != null){
        for(no p = l.prim; p != null; p = p.prox)
            for(no q = p.prox; q != null; q = q.prox)
                if((p.ptos < q.ptos) || ((p.ptos == q.ptos) && (p.vit < q.vit))){
                    String n = p.nome;
                    p.nome = q.nome;
                    q.nome = n;
                    int v = p.vit;
                    p.vit = q.vit;
                    q.vit = v;
                    int d = p.der;
                    p.der = q.der;
                    q.der = d;
                    int ptos = p.ptos;
                    p.ptos = q.ptos;
                    q.ptos = ptos;
                }
        int pos = 1;
        for(no p = l.prim; p != null; p = p.prox) p.pos = pos++;
        no p = l.prim;
        while(p != null){
            if((p.prox != null) && (p.ptos == p.prox.ptos) && (p.vit == p.prox.vit))
                p.prox.pos = p.pos;
            p = p.prox;
        }
    }
}

```

```
}  
}  
}  
}
```

### Questão 2) (5.0 pontos)

Uma questão de uma prova anterior teve como gabarito o código abaixo. Este realiza a manipulação de figuras geométricas no plano 2D.

```
interface FiguraGeom {  
    double perimetro();  
    double area();  
}  
  
abstract class Quadrilatero implements FiguraGeom {  
    double lado1, lado2, lado3, lado4;  
    public Quadrilatero(double l1, double l2, double l3, double l4) {  
        lado1 = l1; lado2 = l2; lado3 = l3; lado4 = l4;  
    }  
    public double perimetro() {  
        return lado1 + lado2 + lado3 + lado4;  
    }  
}  
  
class Retangulo extends Quadrilatero {  
    public Retangulo(double b, double h) {  
        super(b, h, b, h);  
    }  
    public double area() {  
        return lado1 * lado2;  
    }  
}  
  
class Quadrado extends Retangulo {  
    public Quadrado(double l) {  
        super(l, l);  
    }  
}  
  
class Circulo implements FiguraGeom {  
    double raio;  
    public Circulo(double r) {  
        raio = r;  
    }  
    public double area() {  
        return Math.PI * Math.pow(raio, 2);  
    }  
    public double perimetro() {  
        return 2 * Math.PI * raio;  
    }  
}
```

Adicione/altere o código de forma que este possa manipular, além das figuras 2D, figuras em 3D. Por exemplo, o quadrado ou o retângulo com uma altura dão origem a um cubo (caixa). O círculo com uma altura dá origem a um cilindro. Para estas figuras só é necessário o cálculo do volume. Tanto para cubos quanto para cilindros os seus volumes são a área da base multiplicada pela altura do objeto. Garanta que todos os objetos 3D

saibam responder a este método.

RESPOSTA:

```
interface FiguraGeom3D {
    double volume();
}

class Cubo extends Retangulo implements FiguraGeom3D {
    double altura;

    public Cubo(double b, double h, double alt) {
        super(b, h);
        altura = alt;
    }

    public double volume() {
        return this.area() * altura;
    }
}

class Cilindro extends Circulo implements FiguraGeom3D {
    double altura;

    public Cilindro(double r, double alt) {
        super(r);
        altura = alt;
    }

    public double volume() {
        return this.area() * altura;
    }
}
```