



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**

**Disciplina: Programação III**

**AP2 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) (3.0 pontos)**

Escreva um programa que receba o nome de um arquivo texto e retire deste texto palavras consecutivas repetidas. O seu programa deve retornar, no arquivo de saída “Q1” acrescido do nome de arquivo de entrada, a resposta desta questão.

Por exemplo, se o conteúdo de um arquivo texto for: **Isto e um texto texto repetido repetido repetido . . Com as repeticoes repeticoes fica fica sem sem sentido . Sem elas elas elas melhora melhora um um pouco . . . .**

A saída do seu programa será: **Isto e um texto repetido . Com as repeticoes fica sem sentido . Sem elas melhora um um pouco .**

**RESPOSTA:**

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

```
class no{  
    String info;  
    no prox;
```

```
    no(String info){  
        this.info = info;  
        prox = null;
```

```

    }

    public String toString(){ return info + " "; }
}

class lista{
    no prim;

    lista(){ prim = null; }

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

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

        return resp + "\n";
    }
}

public class Q1_AP2_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{
            while ((s = in.readLine()) != null){
                String vs[] = s.split(" ");
                for(int i = 0; i < vs.length; i++) l.insere(vs[i]);
            }
        }catch (Exception e){

```

```

        System.out.println("Excecao leitura\n");
    }finally{
        in.close();
    }

    BufferedWriter out = new BufferedWriter(new FileWriter("Q1-" + args[0]));

    try{
        out.write(l.toString());
    }catch (Exception e){
        System.out.println("Excecao de escrita\n");
    }finally{
        out.close();
    }
}
}
}

```

### Questão 2) (3.0 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 (levando-se em consideração, SOMENTE, o número de vitórias), deste turno.

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	VIT	
1	RJX	9	
2	SADA CRUZEIRO	8	
2	CANOAS	8	
2	MEDLEY/CAMPINAS		8
5	SESI-SP	7	
5	VIVO/MINAS	7	
7	SAO BERNARDO VOLEI	5	
8	VOLEI FUTURO	4	
8	VOLTA REDONDA	4	
10	SUPER IMPERATRIZ VOLEI	3	
11	UFJF	2	
12	FUNVIC/MIDIA FONE		1

#### RESPOSTA:

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

```
class no{
    String nome;
    int pos, vit;
    no prox;

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

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

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

void insereVitoria(String nome){
    no p = busca(nome);
    p.vit++;
}

public class Q2_AP2_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{
    while((s = in.readLine()) != 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) l.insereVitoria(vs[0]);
        else l.insereVitoria(vs[2]);
    }
}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\tVIT\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.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 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.vit == p.prox.vit))
                p.prox.pos = p.pos;
            p = p.prox;
        }
    }
}

```

```

    }
    }
    }
}

```

### Questão 3) (4.0 pontos)

Dado o método principal abaixo, defina todas as classes e interfaces necessárias, se existente, para que este programa funcione. A identificação destas deve ser feita com a análise de código abaixo. Por exemplo, para que a linha *Produto prod1 = new Produto("Cimento", "CPIII", 51.25);* execute é necessário que existe uma classe *Produto* e que esta tenha um construtor que receba 3 parâmetros, 2 strings e 1 valor decimal, nesta ordem. Na classe (ou interface) *Item* o número representa a quantidade de itens usados de um dado produto, enquanto que o número na classe (ou interface) *Colaborador* o número representa a quantidade de horas trabalhadas de uma pessoa.

```

public class AP2_2013_1_Q3 {
    public static void main(String[] args) {
        Produto prod1 = new Produto("Cimento", "CPIII", 51.25); // Cimento do tipo
        CPIII com valor R$51,25
        Produto prod2 = new Produto("Tijolo", "Comum", 2.50);
        Pessoa p1 = new Pessoa("Pedreiro", 10.00); // Pedreiro que recebe
        R$10,00 / hora
        Pessoa p2 = new Pessoa("Arquiteto", 50.00);
        List<Gasto> obra = new ArrayList<Gasto>();
        obra.add(new Item(prod1, 50));
        obra.add(new Item(prod2, 1000));
        obra.add(new Colaborador(p1, 9));
        obra.add(new Colaborador(p2, 2));
        System.out.println("O total da obra é: ");
        double soma = 0;
        for (Gasto g : obra) {
            soma += g.valor();
        }
        System.out.println(soma);
    }
}

```

### RESPOSTA:

```

interface Gasto {
    double valor();
}

class Produto {
    String nome;
    String tipo;
    double valor;

    public Produto(String nome, String tipo, double valor) {
        this.nome = nome;
        this.tipo = tipo;
        this.valor = valor;
    }
}

```



```
class Pessoa {
    String funcao;
    double valorHora;

    public Pessoa(String funcao, double valorHora) {
        this.funcao = funcao;
        this.valorHora = valorHora;
    }
}

class Item implements Gasto {
    Produto p;
    int qtd;

    public Item(Produto p, int qtd) {
        this.p = p;
        this.qtd = qtd;
    }

    public double valor() {
        return p.valor * qtd * 1.05;
    }
}

class Colaborador implements Gasto {
    Pessoa p;
    int qtd;

    public Colaborador(Pessoa p, int qtd) {
        this.p = p;
        this.qtd = qtd;
    }

    public double valor() {
        return p.valorHora * qtd;
    }
}
```