



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Programação III

AP1 1º semestre de 2012.

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) (4.0 pontos)

Escreva um programa que, inúmeras vezes, leia um número inteiro **N**, seguida de uma sequência de **N** números inteiros, e determine o comprimento máximo de um segmento crescente destes **N** números. Exemplos:

- Na sequência 5, 2, 4, 7, 9, 8, 5 o comprimento do segmento crescente máximo é 4.
- Na sequência 10, 8, 7, 5, 2 o comprimento de um segmento crescente máximo é 1.
- Na sequência 8, 7, 5, 2, 3 o comprimento de um segmento crescente máximo é 2.

Seu programa deve parar SOMENTE quando **N** for menor ou igual a zero.

RESPOSTA:

```
import java.util.Scanner;
```

```
public class Q1_2012_1{
```

```
    public static void main (String [] args){  
        Scanner sc = new Scanner(System.in);  
        int n = Integer.parseInt(sc.nextLine());
```

```
        while(n > 0){  
            int maior = 1, aux = 1, ant, prox, i;  
            ant = Integer.parseInt(sc.nextLine());
```

```

for(i = 2; i <= n; i++){
    prox = Integer.parseInt(sc.nextLine());
    if(prox > ant) aux++;
    else{
        if(aux > maior) maior = aux;
        aux = 1;
    }
    ant = prox;
}

if(aux > maior) maior = aux;
System.out.println ("A maior sequencia e " + maior);
n = Integer.parseInt(sc.nextLine());
}
sc.close();
}
}

```

Questão 2) (3.0 pontos)

Suponha que você foi contratado por uma empresa que fabrica diferentes tipos de ar condicionados (de parede, split – normalmente instalado no teto – e automotivos) e um sistema para controle de estoque precisa ser implementado. O sistema precisa manter informações como o número de série do aparelho, a capacidade do motor (em BTU's) e se possui ar quente. Aparelhos de parede e split, chamados também de residenciais, possuem também uma cor predominante, a voltagem do aparelho (110 ou 220V) e se possui controle remoto. Baseado nestas informações, faça:

- Defina as classes deste sistema (nome das classe, relação de herança e atributos que esta possui)
- Para cada classe, defina um construtor que inicialize todos os atributos
- No método main(), instancie objetos para os aparelhos listados na tabela abaixo

Série	BTU	Ar Quente	Cor	Voltagem	Controle Remoto
1	7500	Não	-	-	-
2	15000	Sim	-	-	-
3	5000	Não	Branca	110	Não
4	10000	Não	Preta	220	Não

RESPOSTA:

```

class ArCond {
    int numSerie;
    int capacidade;
    boolean possuiArQuente;

    public ArCond(int numSerie, int capacidade, boolean possuiArQuente) {
        this.numSerie = numSerie;
        this.capacidade = capacidade;
        this.possuiArQuente = possuiArQuente;
    }
}

class Residencial extends ArCond {
    String cor;
    int voltagem;
    boolean possuiControleRemoto;

    public Residencial(int numSerie, int capacidade, boolean possuiArQuente,
        String cor, int voltagem, boolean possuiControleRemoto) {
        super(numSerie, capacidade, possuiArQuente);
        this.cor = cor;
        this.voltagem = voltagem;
        this.possuiControleRemoto = possuiControleRemoto;
    }
}

// Classe de teste para a classe Telefone acima
public class AP1_2012_1_Q2 {
    public static void main(String[] args) {
        ArCond ar1 = new ArCond(1, 7500, false);
        ArCond ar2 = new ArCond(2, 15000, true);
        Residencial ar3 = new Residencial(3, 5000, false, "branca", 110, false);
        Residencial ar4 = new Residencial(4, 10000, false, "preta", 220, false);
    }
}

```

Questão 3) (3.0 pontos)

Na Matemática, um intervalo é um conjunto de números reais limitados por 2 números. Por exemplo, $[-3, 7.5)$ representa o intervalo que compreende os valores de -3 a 7.5, incluindo o -3 e excluindo o 7.5. Intervalos são elementos fundamentais na matemática intervalar, uma técnica de computação numérica que garante resultados, mesmo na presença de incertezas e/ou aproximações.

Suponha a definição da classe Limite, que representa 1 limite de um intervalo, assim como da classe IntervaloSimples abaixo, as quais representam o conceito de intervalo fornecido:

```

class Limite {
    private float valor;
    private boolean aberto;

    public Limite(float li, boolean a) {
        valor = li;
        aberto = a;
    }

    public float getValor() {
        return valor;
    }

    public boolean isAberto() {

```

```

        return aberto;
    }

    public boolean equals(Limite l) {
        return (this.valor == l.valor && this.aberto == l.aberto);
    }
}

class IntervaloSimples {
    private Limite limiteInf;
    private Limite limiteSup;

    public IntervaloSimples(float linf, boolean abertolinf, float lsup, boolean
abertolsup) {
        limiteInf = new Limite(linf, abertolinf);
        limiteSup = new Limite(lsup, abertolsup);
    }
}

```

- Implemente na classe *IntervaloSimples* o método *boolean contem(int v)*, o qual indica se o valor *v* pertence a um intervalo
- Implemente na classe *IntervaloSimples* o método *float media()*, o qual retorna o valor médio entre os limites do intervalo
- Implemente na classe *IntervaloSimples* o método *String toString()*, o qual retorna uma string com formato dado no enunciado: “[-3, 7.5)”

RESPOSTA:

```

class IntervaloSimples {
    private Limite limiteInf;
    private Limite limiteSup;

    public IntervaloSimples(float linf, boolean abertolinf, float lsup, boolean
abertolsup) {
        limiteInf = new Limite(linf, abertolinf);
        limiteSup = new Limite(lsup, abertolsup);
    }

    public boolean contem (float v) {
        return ((v > limiteInf.getValor() && v < limiteSup.getValor()) ||
(!limiteInf.isAberto() && v == limiteInf.getValor()) ||
(!limiteSup.isAberto() && v == limiteSup.getValor()));
    }

    public float media () {
        return ((limiteInf.getValor() + limiteSup.getValor()) / 2);
    }

    public void exhibe () {
        System.out.println("Intervalo: " + this);
    }

    public String toString() {
        String abre = "[", fecha = "]";
        if (limiteInf.isAberto())
            abre = "(";
        if (limiteSup.isAberto())
            fecha = ")";
        return abre + limiteInf.getValor() + "," + limiteSup.getValor() + fecha;
    }
}

```