Prova d'Esame del 15/09/2011

Esercizio 1. Si deve modellare, utilizzando il linguaggio di programmazione Java, un listino di titoli azionari quotati in borsa. In particolare, ciò che interessa rappresentare, in uno scenario molto semplificato, è l'andamento dei titoli e del listino. Per semplicità, supponiamo che un titolo azionario sia caratterizzato dal nome, dalla quotazione del titolo ad inizio seduta di borsa e dalla quotazione corrente. Tutti i titoli sono contenuti in uno stesso listino.
Si implementino le classi TitoloAzionario e ListinoAzionario con le opportune interfacce. In particolare, la classe ListinoAzionario dovrà prevedere almeno i seguenti metodi (se necessario, se ne potranno ovviamente definire degli altri ausiliari):

- **costruttore**: letti da input una sequenza di titoli azionari, provvede a riempire il listino. La sequenza in input termina quando si inserisce un titolo con nome XXX e valori iniziale e corrente entrambi pari a 0 (zero).
- **stampaListino**: stampa in output tutti i titoli inclusi nel listino. Si noti che la stampa di un titolo consiste qui nella stampa del suo nome seguito dal valore attuale e dalla variazione percentuale rispetto al valore di apertura. ESEMPIO:
  - Mediobanca 5,94 -4,04%
- **stampaValoreListino**: stampa in output il valore corrente del listino, calcolato come la somma dei valori correnti di tutti i titoli.
- **stampaIndiceListino**: stampa in output la variazione percentuale del listino, calcolata come segue: *(VLC-VLI)/VLI * 100*, dove
  - **VLC** è il valore totale corrente del listino (quello stampato dal metodo "stampaValoreListino");
  - **VLI** è il valore totale iniziale del listino (calcolato come VLC, ma sui valori iniziali dei titoli invece che su quelli correnti).
- ***I metodi descritti di seguito sono obbligatori SOLO per gli studenti di Informatica. Gli studenti di Informatica 2 possono considerarli opzionali; coloro tra questi che dovessero decidere di svolgerli, e lo facessero correttamente, si vedranno riconosciuto un BONUS sul punteggio della prova scritta.***
  - **ordinaListino**: ordina il listino in base ai NOMI dei titoli secondo l'ordine lessicografico crescente.
  - **stampaStar**: stampa in output i due titoli con la variazione percentuale più alta.

SOLUZIONE

```
import java.util.Scanner;

/*
 * Un titolo azionario e` caratterizzato dal nome, dalla quotazione del titolo ad inizio seduta di borsa e
 * dalla quotazione corrente.
 */
public class TitoloAzionario {

    private String nome;
    private double quotazioneIniziale;
    private double quotazioneCorrente;

    public String getNome() {
        return nome;
    }
    public void setNome(String nome) {
        this.nome = nome;
    }
    public double getQuotazioneIniziale() {
        return quotazioneIniziale;
    }
    public void setQuotazioneIniziale(double quotazioneIniziale) {
        this.quotazioneIniziale = quotazioneIniziale;
    }
    public double getQuotazioneCorrente() {
        return quotazioneCorrente;
    }
    public void setQuotazioneCorrente(double quotazioneCorrente) {
        this.quotazioneCorrente = quotazioneCorrente;
    }

    public String toString() {
        double variazione = getVariazionePercentuale();
        return nome + "\t\t\t" + quotazioneCorrente + "\t" + (variazione > 0 ? "+" : "") + variazione + "%";
    }

    public double getVariazionePercentuale() {
        return (quotazioneCorrente - quotazioneIniziale) / quotazioneIniziale * 100;
    }

    public static TitoloAzionario leggi(Scanner input) {
```


CAMPUS DI ARCAVACATA - Ponte Pietro Bucci Cubo 30B - 87036 Arcavacata di Rende (CS) tel. (+39) 0984 496400 fax (+39) 0984 496410
http://sv.mat.unical.it

Prova d'Esame del 15/09/2011

```
        TitoloAzionario t = new TitoloAzionario();
        System.out.println("*** INSERISCI I DATI DEL TITOLO ***");

        System.out.print("Nome: ");
        t.nome = input.next();

        System.out.print("Quotazione iniziale: ");
        t.quotazioneIniziale = input.nextDouble();

        System.out.print("Quotazione corrente: ");
        t.quotazioneCorrente = input.nextDouble();

        return t;
    }
}

import java.util.Scanner;

public class ListinoAzionario {

    private TitoloAzionario[] titoli = new TitoloAzionario[0];
    private int numeroTitoli = 0;

    public ListinoAzionario(Scanner input) {
        TitoloAzionario t = TitoloAzionario.leggi(input);
        while(!t.getNome().equals("XXX") || t.getQuotazioneIniziale() != 0.0 ||
                t.getQuotazioneCorrente() != 0.0) {
            TitoloAzionario[] tmp = new TitoloAzionario[numeroTitoli + 1];
            for(int i = 0; i < numeroTitoli; i++)
                tmp[i] = titoli[i];
            tmp[numeroTitoli] = t;

            titoli = tmp;
            numeroTitoli++;

            t = TitoloAzionario.leggi(input);
        }
    }

    public void stampaListino() {
        System.out.println("*** TITOLI NEL LISTINO ***");
        for(int i = 0; i < numeroTitoli; i++)
            System.out.println(titoli[i]);
    }

    public void stampaValoreListino() {
        double somma = 0.0;
        for(int i = 0; i < numeroTitoli; i++)
            somma += titoli[i].getQuotazioneCorrente();
        System.out.println("Valore listino: " + somma);
    }

    public void stampaIndiceListino() {
        double VLC = 0.0;
        double VLI = 0.0;
        for(int i = 0; i < numeroTitoli; i++) {
            VLC += titoli[i].getQuotazioneCorrente();
            VLI += titoli[i].getQuotazioneIniziale();
        }
        double variazione = (VLC - VLI) / VLI * 100;
        System.out.println("Indice listino: " + (variazione > 0 ? "+" : "") + variazione);
    }

    public void ordinaListino() {
        mergeSort(0, numeroTitoli - 1);
    }

    private void mergeSort(int first, int last) {
        if(first >= last)
            return;
        System.out.println("." + first + " " + last);
        int mid = (first + last) / 2;
```


CAMPUS DI ARCAVACATA - Ponte Pietro Bucci Cubo 30B - 87036 Arcavacata di Rende (CS) tel. (+39) 0984 496400 fax (+39) 0984 496410
http://sv.mat.unical.it

Prova d'Esame del 15/09/2011

```
        mergeSort(first, mid);
        mergeSort(mid+1, last);

        TitoloAzionario[] tmp = new TitoloAzionario[numeroTitoli];
        int i = first;
        int j = mid + 1;
        int k = first;
        while(i <= mid && j <= last) {
            if(titoli[i].getNome().compareTo(titoli[j].getNome()) < 0)
                tmp[k++] = titoli[i++];
            else
                tmp[k++] = titoli[j++];
        }
        while(i <= mid)
            tmp[k++] = titoli[i++];
        while(j <= last)
            tmp[k++] = titoli[j++];
        while(first <= last) {
            titoli[first] = tmp[first];
            first++;
        }
    }

    public void stampaStar() {
        if(numeroTitoli == 0) {
            System.out.println("*** TITOLI STAR ***");
            System.out.println("Listino vuoto!");
            return;
        }
        if(numeroTitoli == 1) {
            System.out.println("Unico titolo: " + titoli[0]);
            return;
        }

        int max = 0;
        int submax = 1;
        if(titoli[1].getVariazionePercentuale() >= titoli[max].getVariazionePercentuale()) {
            max = 1;
            submax = max;
        }

        for(int i = 2; i < numeroTitoli; i++) {
            if(titoli[i].getVariazionePercentuale() >= titoli[max].getVariazionePercentuale()) {
                max = i;
                submax = max;
            }
            else if(titoli[i].getVariazionePercentuale() > titoli[submax].getVariazionePercentuale()) {
                submax = i;
            }
        }

        System.out.println("1°: " + titoli[max]);
        System.out.println("2°: " + titoli[submax]);
    }

    public TitoloAzionario[] getTitoli() {
        return titoli;
    }

    public void setTitoli(TitoloAzionario[] titoli) {
        this.titoli = titoli;
    }

    public static void main(String[] args) {
        //Scanner input = new Scanner(System.in);
        Scanner input = new Scanner(
                "aaa 10 20 " +
                "ccc 10 5 " +
                "bbb 10 15 " +
                "XXX 0 0 ");
        ListinoAzionario l = new ListinoAzionario(input);
        l.stampaListino();
```


CAMPUS DI ARCAVACATA - Ponte Pietro Bucci Cubo 30B - 87036 Arcavacata di Rende (CS) tel. (+39) 0984 496400 fax (+39) 0984 496410
http://sv.mat.unical.it

Prova d'Esame del 15/09/2011

```
        l.stampaValoreListino();
        l.stampaIndiceListino();
        l.ordinaListino();
        l.stampaListino();
        l.stampaStar();
    }

}
```

Esercizio 2. Si consideri il metodo riportato di seguito.

```
void effe (int n, int v[], int x) {
        if ( x <= n ) {
                if ( n % x == 0 ) {
                        v[x]++;
                        effe (n/x, v, x);
                }
                else
                        effe (n, v, x+1);
        }
}
```

Si descriva sinteticamente la funzione svolta dal metodo e, in particolare, si mostri l'esecuzione e cosa viene restituito nel caso in cui viene invocato con *n=20, v = {0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0}* e *x=2*.

SOLUZIONE

Il metodo è RICORSIVO, ed in pratica effettua la scomposizione in fattori primi di un numero. In particolare, ricevuti un numero intero "n", un array di interi "v" (di dimensione "n", ed inizializzato con zeri), ed un intero "x", modifica i valori di v, ponendo in corrispondenza delle posizioni di v che sono fattori primi di n l'esponente con cui quel fattore compare nella scomposizione di n.
Infatti, in ogni chiamata ricorsiva della funzione, si controlla se x (il fattore da considerare) è minore o uguale ad n (il numero da scomporre); in caso affermativo, si verifica se x è un divisore di n, ed in tal caso si incrementa di 1 il valore nella posizione x di v, e si richiama inoltre la funzione passando come numero da scomporre n/x. Se, invece, x non è un divisore di n, si richiama la funzione passando come fattore da considerare x+1. Il processo naturalmente si ferma quando il fattore da considerare è maggiore del numero da scomporre.
I passi che il metodo esegue nel caso indicato sono i seguenti:

n=20; v = {0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0}; x=2 ← v = {0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0}.
n=10; v = {0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0}; x=2 ← v = {0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0}.
n=5; v = {0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0}; x=2 ← v = {0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0}.
n=5; v = {0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0}; x=3 ← v = {0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0}.
n=5; v = {0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0}; x=4 ← v = {0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0}.
n=5; v = {0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0}; x=5 ← v = {0 0 2 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0}.
n=1; v = {0 0 2 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0}; x=5 ← fine.

Esercizio 3. Si deve scrivere in Java un metodo avente il seguente prototipo:

public static void *verificaMagico*(int M[][])

che stabilisca se una matrice quadrata di dimensioni N x N ricevuta in input rappresenta un "quadrato magico", oppure un "quadrato quasi magico", o nessuno dei due, e comunichi l'esito della verifica stampando in output rispettivamente: MAGICO, QUASI-MAGICO, BABBANO.
In questo esercizio diamo una definizione semplificata di quadrato magico. Intendiamo qui per quadrato magico una matrice in cui la somma degli elementi di ciascuna riga, degli elementi di ciascuna colonna e degli elementi di ciascuna delle due diagonali (principale e secondaria) è esattamente pari ad uno stesso valore, detto "costante magica". Si intende per quadrato quasi magico una matrice che rispetta la definizione di quadrato magico, tranne che per un numero di righe pari al massimo ad N/2, e per un numero di colonne pari al massimo ad N/2. Una matrice in cui più di N/2 righe oppure più di N/2 colonne hanno come somma un valore diverso dalla costante magica non rappresenta né un quadrato magico, né uno quasi magico, ed è quindi un quadrato "babbano".

SUGGERIMENTO: si noti che in una matrice, se essa rappresenta un quadrato magico oppure un quadrato quasi magico, sommando i valori della diagonale principale si ottiene SEMPRE lo stesso valore che si ottiene sommando quelli della diagonale secondaria (valore che è proprio la costante magica).

CAMPUS DI ARCAVACATA - Ponte Pietro Bucci Cubo 30B - 87036 Arcavacata di Rende (CS) tel. (+39) 0984 496400 fax (+39) 0984 496410
http://sv.mat.unical.it

Prova d'Esame del 15/09/2011

16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

quadrato magico

16	1	2	13
5	10	11	8
9	6	7	12
4	15	14	1

quadrato quasi magico

16	1	2	13
5	10	11	8
9	6	7	12
9	15	14	1

quadrato babbano

ESEMPIO: nella figura qui sopra è facile individuare matrici dei tre tipi descritti. In particolare, la prima matrice rappresenta un quadrato magico. Qualunque riga, colonna o diagonale si scelga, la somma degli elementi è sempre pari alla costante magica 34. La seconda matrice, invece, è un quadrato quasi magico: infatti, soltanto una riga ed una colonna vedono la somma dei propri elementi diversa da quella delle due diagonali. La terza matrice, infine, vede le due diagonali a somma differente, caratteristica che, assieme alla riga ed alla colonna "sbagliate" fa di essa un quadrato babbano.

SOLUZIONE

```
import java.util.*;

public class QuadratiMagici {

	public static Scanner input = new Scanner(System.in);

	public static void leggiMatrice(int m[][]) {
		System.out.println("Inserire gli elementi della matrice - " + m.length
				+ " righe, ");

		for (int i = 0; i < m.length; i++) {
			System.out.println(m[i].length + " elementi per la riga " + i);
			for (int j = 0; j < m[i].length; j++)
				m[i][j] = input.nextInt();
			System.out.println();
		}
	}

	public static void scriviMatrice(int m[][]) {
		for (int i = 0; i < m.length; i++) {
			for (int j = 0; j < m[i].length; j++)
				System.out.print(m[i][j] + " ");
			System.out.println();
		}
	}

	public static int sommaRiga(int m[][], int i) {
		if (i >= m.length)
			return -1;
		int somma = 0;
		for (int j = 0; j < m.length; j++)
			somma += m[i][j];
		return somma;
	}

	public static int sommaColonna(int m[][], int i) {
		if (i >= m.length)
			return -1;
		int somma = 0;
		for (int j = 0; j < m.length; j++)
			somma += m[j][i];
		return somma;
	}

	public static void verificaMagico(int m[][]) {
		int diagPrinc = 0, diagSec = 0;
```


CAMPUS DI ARCAVACATA - Ponte Pietro Bucci Cubo 30B - 87036 Arcavacata di Rende (CS) tel. (+39) 0984 496400 fax (+39) 0984 496410
http://sv.mat.unical.it

Prova d'Esame del 15/09/2011

```
            // verifica che le somme degli elementi sulle due diagonali coincidano:
            // se non e' cosi', abbiamo un quadrato babbano
            for (int i = 0; i < m.length; i++) {
                  diagPrinc += m[i][i];
                  diagSec += m[i][m.length - 1 - i];
            }
            if (diagPrinc != diagSec) {
                  System.out.println(" QUADRATO BABBANO!!! ");
                  return;
            }

            // se siamo arrivati qui, c'e' speranza che il quadrato sia magico o
            // quasi magico; in ogni caso, l'eventuale costante magica e' il valore
            // delle diagonali
            int costanteMagica = diagPrinc;

            int righeNonMagiche = 0, colonneNonMagiche = 0;
            // conta righe e colonne "non magiche"
            for (int i = 0; i < m.length; i++) {
                  if (sommaRiga(m, i) != costanteMagica)
                        righeNonMagiche++;
                  if (sommaColonna(m, i) != costanteMagica)
                        colonneNonMagiche++;
            }

            // verifica: se il numero di righe/colonne non magiche e' zero, abbiamo
            // un quadrato magico! Se non lo e', ma in entrambi i casi e' inferiore a N/2,
            // abbiamo un quadrato quasi magico; altrimenti, resta un quadrato babbano…
            if (righeNonMagiche == 0 && colonneNonMagiche == 0)
                  System.out.println(" QUADRATO MAGICO!!! ");
            else if (righeNonMagiche <= m.length / 2
                        && colonneNonMagiche <= m.length / 2)
                  System.out.println(" QUADRATO QUASI-MAGICO!!! ");
            else
                  System.out.println(" QUADRATO BABBANO!!! ");
      }


      public static void main(String[] args) {
            System.out.println(" *** Quadrati Magici, Quasi Magici e Babbani *** \n ");
            System.out.println(" Inserire la dimensione della matrice da
                                    verificare... ");
            int n = input.nextInt();
            int matrice[][] = new int[n][n];
            leggiMatrice(matrice);
            verificaMagico(matrice);
      }
}
```


CAMPUS DI ARCAVACATA - Ponte Pietro Bucci Cubo 30B - 87036 Arcavacata di Rende (CS) tel. (+39) 0984 496400 fax (+39) 0984 496410
http://sv.mat.unical.it