Corso di Laurea in Matematica
Corso di Informatica
**Prova d'Esame del 02/02/2010**

Dipartimento di MATEMATICA

**Esercizio 1.** Si deve modellare con una classe Java il contenuto di un portamonete, con delle opportune semplificazioni. Le monete da considerare sono di soli 4 tagli: 10, 20, 50 e 100 centesimi (cioè 1 Euro). Gli importi sono espressi con numeri INTERI, che rappresentano centesimi di euro (ad esempio 2 euro e 50 centesimi è rappresentato dal numero 250). Per fare riferimento ai tagli, la classe, denominata "**ImportoInMonete**", contiene un array statico di 4 elementi:

```
private static int[] tagli = {10, 20, 50, 100};
```

In aggiunta, è necessario modellare il contenuto: pertanto ogni oggetto della classe dovrà memorizzare quante monete di ciascun taglio contiene (è possibile scegliere liberamente in che modo farlo). Si implementino almeno i seguenti metodi (Se necessario, si potranno ovviamente definire altri metodi ausiliari).

```
private ImportoInMonete (int c10, int c20, int c50, int c100)   // Costruttore privato (utilizzato dal metodo "leggi()".
public static ImportoInMonete leggi (Scanner input)             // Metodo statico per la lettura da input di un oggetto.
public String toString()                                        // Metodo per la stampa su output.
public int getImporto()                                         // Restituisce un intero che rappresenta l'importo contenuto
                                                                // nel portamonete
private int estraiMonete (int t, int num)                       // Estrae "num" monete da "tagli[t]" centesimi e
                                                                // restituisce l'importo corrispondente. Es.: se "num == 5" e
                                                                // "t == 0", allora il metodo estrarrà 5 monete da
                                                                // "tagli[0]", cioè 10 centesimi, e restituirà "50" (cioè l'importo
                                                                // corrispondente a 5 monete da 10 centesimi).
public void aggiungiImportoInMonete (ImportoInMonete m)         // Aggiunge al portamonete il contenuto di un altro
                                                                // portamonete.
public ImportoInMonete estraiImporto (int m)                    // Dato un intero "m" che esprime un importo in centesimi,
                                                                // estrae dal portamonete il minor numero di monete che
                                                                // approssima "m" (per difetto, se per caso non ci fossero
                                                                // abbastanza monete). Es.: se "m == 75" e l'oggetto di
                                                                // di tipo ImportoInMonete su cui è invocato il metodo
                                                                // contenesse 0 monete da 100, 3 monete da 50, 2 monete da
                                                                // 20 e 1 moneta da 10, allora l'oggetto di tipo
                                                                // ImportoInMonete restituito dovrebbe contenere 0 monete
                                                                // da 100, 1 monete da 50, 1 moneta da 20 e 0 monete da 10.
                                                                // L'oggetto di tipo ImportoInMonete su cui è stato invocato il
                                                                // metodo, invece, vedrebbe decrementato il numero di
                                                                // monete da 50 e da 20 centesimi di una unità.
```

**SOLUZIONE:**

```
public class ImportoInMonete {

        private int[] monete = new int[4];
        private static int[] tagli = {10, 20, 50, 100};

        public ImportoInMonete(){
        }

        private ImportoInMonete(int c10, int c20, int c50, int c100){
                monete[0] = c10;
                monete[1] = c20;
                monete[2] = c50;
                monete[3] = c100;
        }

        public ImportoInMonete(ImportoInMonete i1){
                for (int i = 0; i < monete.length; i++)
                        monete[i] = i1.monete[i];
        }

        public static ImportoInMonete 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

```
        System.out.print("Numero monete da 10 centesimi: ");
        int c10 = input.nextInt();
        if(c10 < 0)
                c10 = 0;
        System.out.print("Numero monete da 20 centesimi: ");
        int c20 = input.nextInt();
        if(c20 < 0)
                c20 = 0;
        System.out.print("Numero monete da 50 centesimi: ");
        int c50 = input.nextInt();
        if(c50 < 0)
                c50 = 0;
        System.out.print("Numero monete da 1 euro: ");
        int c100 = input.nextInt();
        if(c100 < 0)
                c100 = 0;
        System.out.println();
        return new ImportoInMonete(c10, c20, c50, c100);
        }

        public String toString(){
                String output = "";
                for(int i = monete.length-1; i >= 0; i--){
                        String t = "";
                        if(tagli[i] == 100)
                                t = " euro ";
                        else
                                t = " " + tagli[i] + " centesimi ";
                        output += monete[i] + t;
                }
                return output;
        }

        public boolean equals(ImportoInMonete m){
                for(int i = 0; i < monete.length; i++)
                        if(monete[i] != m.monete[i])
                                return false;
                return true;
        }

        public int getImporto(){
                int importo = 0;
                for(int i = 0; i<monete.length; i++)
                        importo += monete[i]*tagli[i];
                return importo;
        }

        public int svuota(){
                int importo = getImporto();
                for(int i = 0; i<monete.length; i++)
                        monete[i] = 0;
                return importo;
        }

        private int estraiMonete(int taglio, int num){
                if(monete[taglio] < num){
                        String t = "";
```



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

```
                    if(tagli[taglio] == 100)
                        t = "1 euro";
                    else
                        t = tagli[taglio] + " centesimi";
                System.out.print("Disponibilità di monete da " + t + " insufficiente!");
                }
                else
                    monete[taglio] -= num;
                return num*tagli[taglio];
        }

        public int estraiDieciCents(int num){
                return estraiMonete(0, num);
        }

        public int estraiVentiCents(int num){
                return estraiMonete(1, num);
        }

        public int estraiCinquantaCents(int num){
                return estraiMonete(2, num);
        }

        public int estraiEuro(int num){
                return estraiMonete(3, num);
        }

        public int estraiImportoInMonete(ImportoInMonete m){
                int importo = 0;
                for(int i = 0; i<monete.length; i++)
                    if(monete[i] >= m.monete[i]){
                        monete[i] -= m.monete[i];
                        importo += m.monete[i]*tagli[i];
                    }
                    else{
                        monete[i] = 0;
                        importo += monete[i]*tagli[i];
                    }
                return importo;
        }

        public ImportoInMonete estraiImporto(int m){
                int importo = 0, i = monete.length - 1, residuo = m;
                ImportoInMonete imp = new ImportoInMonete();
                while(importo < m && i >= 0){
                    while(residuo >= tagli[i] && monete[i] > 0){
                        imp.monete[i]++;
                        monete[i]--;
                        importo += tagli[i];
                        residuo -= tagli[i];
                    }
                    i--;
                }
                return imp;
        }

        private void aggiungiMonete(int taglio, int num){
```



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

```
            monete[taglio] += num;
        }

        public void aggiungiDieciCents(int num){
            aggiungiMonete(0, num);
        }

        public void aggiungiVentiCents(int num){
            aggiungiMonete(1, num);
        }

        public void aggiungiCinquantaCents(int num){
            aggiungiMonete(2, num);
        }

        public void aggiungiEuro(int num){
            aggiungiMonete(3, num);
        }

        public void aggiungiImportoInMonete(ImportoInMonete m){
            for(int i = 0; i<monete.length; i++)
                monete[i] += m.monete[i];
        }

}
```

**Esercizio 2.** É data la seguente classe Java.

```
public class InterpretazioneCodice {

  public static void g (int[] v, int a, int b) {
        int c = v[a];
        int d = v[b];
        if(d > c) {
          v[a] = d;
          v[b] = c;
        }
  }

}
```

```
public static void f (int[] v, int a, int b) {
        if(a < b) {
          f(v, a+1, b-1);
          g(v, a, b);
        }
}
```

```
public static void main (String[] args) {
        int[] v = {6,8,1,5,2,10,3,4,7,9};
        f(v, 0, v.length-1);
        for(int i = 0; i < v.length; i++)
          System.out.println(v[i]);
}
```

Si svolgano i seguenti punti.

a) Indicare cosa verrà stampato in output dal "main()" (cioè: indicare qual è il contenuto dell'array "v" alla fine dell'esecuzione del programma).
b) Descrivere il comportamento del metodo "g()".
c) Descrivere il comportamento del metodo "f()".

**SOLUZIONE:**

a) alla fine v conterrà 9,8,4,5,10,2,3,1,7,6.
b) il metodo g(v,a,b) scambia v[a] con v[b] se v[b] > v[a].
c) il metodo f(v,a,b) applica ricorsivamente g().

**Esercizio 3.** Si implementi in Java un metodo avente il seguente prototipo:

**public static boolean** verifica (**int** v[], **int** i)



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

Il metodo deve restituire **true** se e solo se l'array "v" è costituito da una sequenza alternata di numeri positivi e negativi. In altre parole, ogni numero positivo deve sempre essere seguito da un numero negativo, ed ogni numero negativo deve sempre essere seguito da un numero positivo. Il metodo deve restituire **false** altrimenti.
**NOTA: verrà riconosciuto un bonus se il metodo implementato è ricorsivo.**

**SOLUZIONE:**

```
public static boolean verifica (int v[], int i) {
        if ( i < v.length-1)
                if (v[i] < 0 && v[i+1] < 0 || v[i] > 0 && v[i+1] > 0)
                        return false;
                else
                        return verifica(v,i+1);
        else
                return true;
}
```

**Esercizio 4.** Si implementi in Java un metodo avente il seguente prototipo:

**public static void** linearizza (**int** m[][], **int** x, **int** y, **int** nElementi, **int** v[])

La matrice (array bidimensionale) di numeri interi "m" si suppone già inizializzata, mentre lo scopo del metodo è quello di riempire l'array (monodimensionale) "v" con alcuni elementi di "m" secondo quanto descritto di seguito: si riempia "v" con gli "nElementi" presenti nella matrice a partire dalla cella in posizione [x,y].

| | | | |
|---|---|---|---|
| 8 | 7 | 4 | 3 |
| 5 | **8** | **1** | **3** |
| **2** | **6** | 9 | 1 |
| 1 | 2 | 4 | 5 |

ESEMPIO: Se la matrice "m" fosse quella riportata qui di fianco, e il metodo "linearizza()" fosse invocato (ad esempio all'interno di un ipotetico "main()") nel seguente modo:

linearizza (m, 1, 1, 5, v);

il vettore "v" dovrebbe essere riempito con i seguenti valori:

| | | | | |
|---|---|---|---|---|
| 8 | 1 | 3 | 2 | 6 |

Si noti che "v" si suppone essere già stato allocato al momento della chiamata della funzione "linearizza()"; pertanto, la funzione dovrà solo "riempirlo" adeguatamente.

**SOLUZIONE:**

```
public static void linearizza (int m[][], int x, int y, int nElementi, int v[]) {
        int n=0;
        for (int i = x; i < m.length; i++) {
                for (int j = y; j < m.length && n < nElementi; j++) {
                        System.out.println(" sto copiando l'elemento in posizione " + i + " " + j + " , che vale  " + m[i][j]);
                        v[n] = m[i][j];
                        n++;
                }
        y=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