

Modulo di Laboratorio di Programmazione I (CdL Informatica) &
Corso di Laboratorio di Programmazione (disattivato - CdL Matematica)

a.a. 2011-2012

Prova Totale - 1 Marzo 2012

NOME

COGNOME

MATRICOLA

CORSO DI LAUREA

Scrivere in stampatello **Nome**, **Cognome** e **Matricola** su ogni foglio consegnato.

Nota Bene: Il testo del compito deve essere riconsegnato insieme al proprio elaborato scritto. Le soluzioni degli esercizi **non** devono contenere alcun metodo **main** né alcuna istruzione di stampa né istruzioni **switch**, **break**, **continue**, etc. La soluzione per l'Esercizio 3 **non** deve contenere alcuna istruzione di ciclo (**for**, **while**, etc.). La non osservanza di tali requisiti comporterà l'annullamento della prova scritta dello studente.

Esercizio 1. Si consideri il seguente frammento di codice in Java:

```
String[] a = b;
int i = 0, j = 0; boolean p = false; String s = "";
while (i < a.length && !p) {
    int n = a[i].length();
    for (j = 0; j < n/2; j++) {
        if (a[i].charAt(j) == a[i].charAt(n-j-1))
            p = true;
    }
    if (p) s = s + a[i];
    i++;
}
```

Determinare il valore finale delle variabili **i**, **j**, **p** ed **s** nei seguenti casi:

- 1) **b** = {"scritto", "prova", "orale", "anno"};
- 2) **b** = {"arco", "maglia", "", "cappa", "orzo", "loop"}.

Esercizio 2. Scrivere un metodo *statico* iterativo che, dati un array bidimensionale di stringhe **a** ed un array monodimensionale di stringhe **c**, restituisce un array monodimensionale di booleani **b** tale che **b**[*i*] vale *true* se la stringa **c**[*i*] è prefisso di *almeno* una stringa in **a**[*i*], altrimenti **b**[*i*] vale *false*. (N.B. utilizzare il metodo `substring`.)

Esempio: se **a** = {"cappello", "", "bozza"}, {"carota", "carro"}, {"su", "pericolo", "sopra", "pero"} e **c** = {"appello", "caro", "per"}, il metodo restituisce l'array {**false**, **true**, **true**}. Modificare la definizione del metodo in modo da sollevare un'eccezione se le lunghezze di **a** e **c** sono diverse.

Esercizio 3. Scrivere una versione ricorsiva del metodo definito per risolvere l'Es.2.

Esercizio 4. Un *giocattolo* può essere caratterizzato da nome, materiale, casa produttrice, nazione in cui è stato prodotto ed età minima consigliata. Scrivere una classe **Giocattolo** con un opportuno costruttore ed i metodi che restituiscono i valori delle variabili istanza. Inoltre, definire un metodo che modifica la nazione, un metodo che modifica l'età minima ed

un metodo `equals` che restituisce *true* se due giocattoli sono uguali, *false* altrimenti. Infine, definire un metodo che restituisce una stringa che descrive un giocattolo.

Esercizio 5. Un *giocattolo che funziona a pile* è un giocattolo caratterizzato anche dal tipo e dal numero di pile necessarie, e dalla presenza o meno di pile nel giocattolo al momento dell'acquisto. Scrivere una classe `GiocattoloPile` con un opportuno costruttore ed i metodi che restituiscono i valori delle variabili istanza. Definire inoltre un metodo che modifica la presenza di pile nel giocattolo ed un metodo che restituisce una stringa che descrive un oggetto della classe `GiocattoloPile`.

Esercizio 6. Un *negozio di giocattoli* può essere caratterizzato tramite nome, nome del responsabile, indirizzo e l'elenco dei giocattoli in vendita. Scrivere una classe `NegozioG`, il cui costruttore imposta il numero massimo dei giocattoli trattati da un negozio. Oltre ai metodi che restituiscono i valori delle variabili istanza, definire i seguenti metodi:

- un metodo che, dato un giocattolo *g*, aggiunge *g* nell'elenco dei giocattoli (il metodo deve sollevare un'opportuna eccezione in caso di elenco pieno);
- un metodo che, dati due interi positivi *m, n* (con $m < n$), restituisce l'elenco di tutti i giocattoli la cui età minima consigliata è compresa tra *m* ed *n*;
- un metodo che, dati una stringa *s* (che denota un tipo di pila) ed un intero $k > 0$, restituisce l'elenco di tutti i giocattoli che funzionano con *k* pile di tipo *s*.

Esercizio 7. Siano date le seguenti classi:

```
class A {
    private String s;

    public A (String t) {
        this.s = t;
    }
    public String getS() {
        return this.s;
    }
    public String met (A a) {
        if (this.s.length() >= a.s.length())
            return this.s + a.s;
        else
            return a.s + this.s;
    }
}

class B extends A {
    private char c;

    public B (String t, char x) {
        super(t);
        this.c = x;
    }
    public char getC() {
        return this.c;
    }
    public String met (A a) {
        if (a.getS().charAt(0) == this.c)
            return super.met(a);
        else
            return this.c + super.met(a);
    }
}
```

Dato il seguente frammento di codice:

```
A obj1 = new A ("febbraio");
B obj2 = new B ("marzo", 'f');
String s = obj1.met(obj2), t = obj2.met(new A ("marzo"));
obj1 = new B ("maggio", 'm');
String u = obj1.met(obj2);
```

determinare il valore delle variabili *s*, *t* ed *u*.