

**Laboratorio di Programmazione I (CdL Informatica) &  
Laboratorio di Programmazione (CdL Matematica)  
a.a. 2005-2006**

Prova Totale - 14 Dicembre 2005

NOME

COGNOME

MATRICOLA

CORSO DI LAUREA

**Nota Bene:** Le soluzioni degli esercizi **non** devono contenere alcun metodo **main** né alcuna istruzione di stampa né istruzioni di **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, r = 0; boolean p = false;
for (i = 0; i < a.length; i++) {
    j = 0; p = false;
    while (j < a[i].length && !p) {
        if (a[i][j].equals(s)) {
            r++;
            if (r >= k) p = true;
        }
        j++;
    }
    r = 0;
}
```

Determinare il valore finale delle variabili  $i$ ,  $j$ ,  $r$  e  $p$  per i seguenti valori di  $b$ ,  $s$  e  $k$ :

- 1)  $b = \{\{ "z", "ff", "z", "ghh" \}, \{ "yy", "z", "z", "pr", "z" \} \}$ ,  $s = "z"$  e  $k = 2$ ;
- 2)  $b = \{\{ "ab", "abc", "abc" \}, \{ "abc", "ab", "r" \}, \{ "abc", "abc" \} \}$ ,  $s = "abc"$  e  $k = 2$ .

**Esercizio 2.** Scrivere un metodo che, dati un array bidimensionale  $a$  di stringhe ed un intero  $k > 0$ , restituisce un array monodimensionale di stringhe  $b$  tale che  $b[i]$  è la prima stringa di  $a[i]$ , la cui lunghezza è divisibile per  $k$ . Se in  $a[i]$  non esistono stringhe la cui lunghezza è divisibile per  $k$ , allora  $b[i]$  è la stringa "-". Ad esempio, dati  $a = \{\{ "treno", "auto", "tram" \}, \{ "uno", "b", "ter" \}, \{ "pope", "papa" \} \}$  e  $k = 2$ , il metodo restituisce l'array  $b = \{ "auto", "-", "pope" \}$ .

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

**Esercizio 4.** La *presentazione di una relazione* in una conferenza può essere descritta tramite il titolo della relazione, il nome del relatore, l'università di provenienza del relatore, la lingua usata ed il tempo a disposizione per la presentazione. Scrivere una classe **Presentazione** con gli opportuni costruttori ed i metodi che restituiscono i valori delle variabili istanza. Inoltre, definire un metodo per modificare il tempo a disposizione ed un metodo che restituisce una stringa che descrive la presentazione di una relazione.

**Esercizio 5.** La *demo di un sistema informatico* è una presentazione, in cui viene mostrato un sistema informatico, caratterizzata anche dal nome del sistema presentato, dal tipo di sistema (Es. open-source) e dal tempo a disposizione per la demo. Scrivere una classe `Demo` con gli opportuni costruttori ed i metodi che restituiscono i valori delle variabili istanza. Infine, definire un metodo che restituisce il tempo a disposizione per la presentazione di relazione e demo, ed un metodo che restituisce una stringa che descrive una demo.

**Esercizio 6.** La *sessione di una conferenza* può essere caratterizzata dal nome della conferenza, dal numero di sessione, dal nome del coordinatore della sessione e dall'elenco delle presentazioni. Scrivere una classe `SessioneConferenza`, il cui costruttore imposta il numero massimo delle presentazioni. Oltre ai metodi che restituiscono i valori delle variabili istanza, definire i seguenti metodi:

- un metodo che aggiunge una presentazione all'elenco di una sessione;
- un metodo che scambia due presentazioni in una sessione;
- un metodo che restituisce *true* se in una sessione esiste almeno una presentazione di cui è relatore il coordinatore della sessione.

**Esercizio 7.** Siano date le seguenti classi:

```
class A {
    private String s;

    public A (String t) {
        this.s = t;
    }
    public String leggiNome() {
        return this.s;
    }
    public String m (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 int c;

    public B (String t, int n) {
        super(t);
        this.c = n;
    }
    public int leggiC() {
        return this.c;
    }
    public String m (A a) {
        return super.m(a) + this.c;
    }
}
```

Dati gli oggetti seguenti:

```
A p = new A ("lpr");
A q = new B ("lp",3);
B r = new B ("lpq",2);
```

determinare qual è il valore restituito dalle seguenti chiamate di metodo:

- 1) `p.m(r)`;
- 2) `q.m(p)`;
- 3) `r.m(q)`.