

Corso di Laurea in Matematica (a.a. 2009-2010)

Informatica (6 cfu)

Prova Scritta - 24 Giugno 2010

NOME

COGNOME

MATRICOLA

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 sui metodi in Java **non** devono contenere alcun metodo **main** né alcuna istruzione di stampa né istruzioni **switch**, **break**, **continue**, etc. La soluzione per l'Esercizio 5 **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 2.** Si consideri il seguente frammento di programma in linguaggio macchina, memorizzato a partire dalla locazione di memoria di indirizzo 100:

**Esercizio 1.**

- i) Dato il numero binario 1001110111, dire qual è la sua rappresentazione nei sistemi decimale, ottale ed esadecimale.
- ii) Convertire il numero decimale 327 nella sua rappresentazione binaria.
- iii) Dato il numero  $327_8$  nel sistema ottale, dire qual è la sua rappresentazione nei sistemi decimale, binario ed esadecimale.
- iv) Convertire il numero decimale  $-114$  nella sua rappresentazione binaria in complemento a due.
- v) Convertire il numero frazionario 0.95 nella sua rappresentazione binaria utilizzando 8 cifre.

```
100 READ 22
101 LOADA 22
102 LOADB 20
103 DIF
104 JUMPZ 119
105 STOREA 23
106 READ 22
107 LOADB 22
108 DIF
109 JUMPZ 111
110 JUMP 101
111 READ 22
112 LOADA 22
113 LOADB 23
114 DIF
115 JUMPZ 117
116 JUMP 101
117 WRITE 21
118 JUMP 120
119 WRITE 20
120 HALT
```

Si supponga che le celle di indirizzo 20 e 21 contengano i valori 0 ed 1 rispettivamente. Data in ingresso una sequenza di interi  $n_1 n_2 \dots n_k$ , dire che cosa fa il programma e che cosa viene stampato in uscita.

**Esercizio 3.** Si consideri il seguente frammento di codice in Java:

```
String[] a = b;
int i = 0, j = 0, k = 0;
boolean p = false;
for (i=0; i<a.length; i++) {
    j = 0; p = false;
    while (j<a[i].length() && !p) {
        if (a[i].charAt(j) == c)
            p = true;
        j++;
    }
    if (p)
        k++;
}
```

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

- 1)  $b = \{ "abddd", "fd", "dgh", "gag" \}$  e  $c = 'd'$ ;
- 2)  $b = \{ "kfg", "m", "bbrrr" \}$  e  $c = 'r'$ .

**Esercizio 4.** Scrivere un metodo statico iterativo che, dato un array monodimensionale  $a$  di caratteri, restituisce un array bidimensionale  $b$  di caratteri tale che la prima riga di  $b$  contiene un'occorrenza del primo elemento di  $a$ , la seconda riga di  $b$  contiene due occorrenze del secondo elemento di  $a$ , ..., l'ultima riga di  $b$  contiene  $a.length$  occorrenze dell'ultimo elemento di  $a$ . Esempio: dato  $a = \{ 'g', 'w', 'p', 'z' \}$ , il metodo restituisce l'array  $b = \{ \{ 'g' \}, \{ 'w', 'w' \}, \{ 'p', 'p', 'p' \}, \{ 'z', 'z', 'z', 'z' \} \}$ .

**Esercizio 5.** Scrivere una versione ricorsiva del metodo definito per risolvere l'Es.4.

**Esercizio 6.** Una *via cittadina* può essere caratterizzata tramite il nome, la città in cui si trova, la lunghezza (espressa in km) e l'essere pedonale o meno. Scrivere una classe `ViaCittadina` con un opportuno costruttore ed i metodi che restituiscono i valori delle variabili istanza. Inoltre, definire un metodo per modificare il nome della via, un metodo per modificare lo status di via pedonale o meno, un metodo che restituisce una stringa che descrive una via cittadina, ed un metodo che restituisce *true* se due vie cittadine si trovano nella stessa città, altrimenti restituisce *false*.

**Esercizio 7.** Una *via storica* è una via cittadina caratterizzata anche dall'elenco dei nomi dei monumenti storici che si trovano in tale via. Scrivere una classe `ViaStorica`, il cui costruttore imposta il numero massimo di monumenti storici di una via. Oltre ai metodi che restituiscono i valori delle variabili istanza, definire:

- un metodo che aggiunge il nome di un monumento nell'elenco di una via storica;
- un metodo che, dato il nome di un monumento, restituisce *true* se tale monumento si trova in una via storica, altrimenti restituisce *false*;
- un metodo che restituisce una stringa che descrive una via storica;
- un metodo che, date due vie storiche, restituisce quella che ha un numero maggiore di monumenti storici.