Programmazione e Algoritmi 1

A.A. 2023/24 — Compito del 20 giugno 2024 – Versione Java

prof. Gianluca Amato

Gli esercizi di programmazione saranno valutati sulla base della correttezza, efficienza e comprensibilità della soluzione proposta. In generale, **se volete usare una funzione o un metodo che non è stato presentato a lezione, chiedete prima al docente se è consentito**.

Esercizio 1 (5 punti)

Spiegare la differenza tra un tipo di dato *mutabile* e uno *immutabile*, fornendo un esempio per ciascuno di essi.

Esercizio 2 (8 punti)

Si consideri il seguente codice Java, e se ne scriva la traccia di esecuzione negli appositi moduli.

```
1  class Esercizio2 {
2      public static void mistero(int[] l1, int[] l2) {
3          for (int i = 0; i < l1.length; i++) {
4              l1[i] = l1[i] + l2[l2.length - 1 - i];
5          }
6      }
7
8      public static void mistero2(int[] l) {
9          for (var x : l) {
10             System.out.println(x);
11         }
12     }
13
14     public static void main(String[] args) {
15         var x = new int[] {1, 2, 3, 4};
16         int[] y = {50, 60, 70, 80};
17         mistero(x, y);
18         mistero2(x);
19     }
20
21 }
```

Esercizio 3 (5 punti)

Scrivere un metodo statico `raddoppia` che prende come parametri un array di interi `l` e una matrice (array di interi bidimensionale) `m`. La funzione deve modificare la matrice `m` in maniera tale che uno ed un solo elemento per ogni riga della matrice venga raddoppato. In particolare, nella riga i-esima l'elemento da raddoppiare è quello nella colonna `l[i]`. L'array `l` invece non deve subire modifiche.

Ad esempio, se `l = {1, 2, 0}` e `m = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}}`, dopo la chiamata `raddoppia(l, m)` la matrice `m` diventa `{{1, 4, 3}, {4, 5, 12}, {14, 8, 9}}`.

Si può assumere che l'array `l` contenga dati corretti, ad esempio che abbia lunghezza uguale al numero di righe della matrice `m`, e che i valori presenti all'interno di `l` siano indici validi per le colonne della matrice `m`.

Si determini, infine, **la complessità computazionale** del metodo `raddoppia`.

Esercizio 4 (5 punti)

Scrivere due test nel framework `JUnit` per verificare il corretto funzionamento di `raddoppia`. In particolare, si controllino i seguenti casi:

- l'input di esempio dell'Esercizio 3;
- `m` è una matrice generata casualmente con 3 righe e 3 colonne e `l={0,0,0}`.

Esercizio 5 (5 punti)

Scrivere un metodo statico `raddoppia2` simile al metodo `raddoppia` dell'Esercizio 3, ma per il quale la correttezza dell'array `l` non è assicurata. Potrebbe quindi accadere che, con questi input, il metodo `raddoppia` generi un errore. Il meotdo `raddoppia2`, in questi casi, non deve generare errori, ma semplicemente non apportare nessuna modifica ad `m`.

Ad esempio, l'input `l = {1, 6, 0}` e `m = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}}` non è valido, perché la riga numero 1 di `m` ha solo 3 elementi, quindi non ha senso raddoppiare l'elemento in colonna 6. In questo caso, il metodo `raddoppia2` **non deve modificare nulla, neanche le altre righe di** `m`.

Suggerimento. Ci sono due modi per risolvere l'esercizio. Uno è controllare preventivamente che la lista `l` sia corretta, e in caso contrario non fare nulla. L'altro è salvare una copia di `m` all'inizio della funzione, e se si verifica un errore ripristinare la matrice `m` alla copia salvata.

Esercizio 6 (5 punti)

Scrivere un programma che genera un numero casuale e chiede all'utente di indovinarlo. Se l'utente indovina, il programma stampa un messaggio di congratulazioni ed esce, altrimenti comunica all'utente se il numero misterioso è più grande o più piccolo del numero inserito dall'utente e chiede un altro tentativo. Il programma continua finché l'utente non indovina il numero misterioso.

Per una valutazione ottimale, i valori minimo e massimo del numero casuale da generare devono essere letti dalla **riga di comando**.

Quello che segue è un esempio di interazione con il programma. In rosso trovate quello scritto dall'utente, in nero l'output del programma.

```
Ho pensato un numero, prova a indovinare.
Inserisci numero: 34
No, il numero che ho pensato è minore di 34.
Inserisci numero: 17
No, il numero che ho pensato è maggiore di 17.
Inserisci numero: 25
Complimenti, hai indovinato|
```