
Fondamenti di informatica II (elettronici) — A.A. 2004-2005
Appello del 23 giugno 2005 — Prima prova scritta — Compito A
Libri e appunti chiusi — Tempo a disposizione: 45 minuti

Cognome: _____ **Nome:** _____ **Matricola:** _____

Esercizio A1 (30%) — *Scrivere circa 1-2 pagine*

Descrivere il modo in cui vengono eseguiti i metodi ricorsivi, facendo anche (ma non solo) un esempio.

Esercizio A2 (35%)

Scrivere, nell'ambito di una classe **Concatenazione**, un metodo di classe **int[] concatena(int[] a, int[] b)** con la seguente specifica:

- Pre-condizione: gli array **a** e **b** sono non nulli
- Post-condizione: il metodo crea e restituisce un nuovo array formato ordinatamente da tutti gli elementi di **a** seguito da tutti gli elementi di **b**

Ad es., se **a** e **b** sono rispettivamente gli array {8, 0, 12} e {1, 7}, allora l'invocazione **concatena(a,b)** deve creare e restituire l'array { 8, 0, 12, 1, 7 }.

Esercizio A3 (35%)

Definire una classe **Giocatore** per istanziare oggetti con le seguenti caratteristiche:

- un oggetto **Giocatore** rappresenta il nome e i soldi (in euro) posseduti da un giocatore in un gioco (tipo Monopoli)
- va definito un costruttore che, dato il nome **n** di un giocatore e un numero intero positivo **s**, crea un nuovo oggetto **Giocatore** il cui nome è **n** e che possiede inizialmente **s** euro
- va definito un metodo d'istanza **String nome()** che restituisce il nome di questo giocatore
- va definito un metodo d'istanza **int soldi()** che restituisce i soldi attualmente posseduti da questo giocatore
- va definito un metodo d'istanza **void incassa(int s)** che incrementa i soldi posseduti da questo giocatore di **s** euro
- va definito un metodo d'istanza **void paga(int s)** che decrementa i soldi posseduti da questo giocatore di **s** euro

Ad esempio:

```
Giocatore a,b;
a = new Giocatore("Alice", 100);
b = new Giocatore("Bob", 100);
System.out.println(a.nome());           // visualizza Alice
System.out.println(a.soldi());           // visualizza 100
System.out.println(b.soldi());           // visualizza 100
a.incassa(20);
b.paga(20);
System.out.println(a.soldi());           // visualizza 120
System.out.println(b.soldi());           // visualizza 80
```

Cognome: _____ Nome: _____ Matricola: _____

Criterio di valutazione

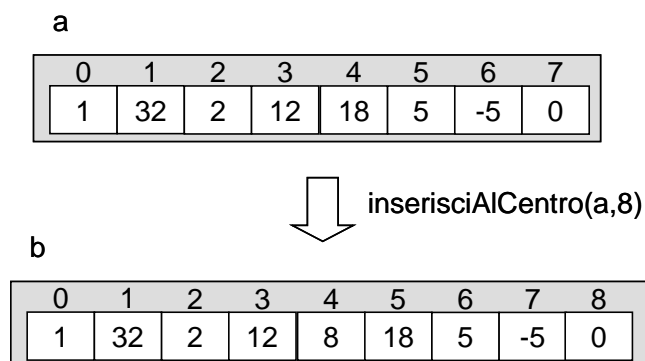
A ciascuno dei due esercizi viene assegnato un voto tra 0 e 5. Siano v_1 e v_2 i voti ottenuti ai due esercizi. Il voto complessivo della prova è $v_1+v_2+\text{massimo}(v_1,v_2)$.

Esercizio A1

Scrivere un metodo `int[] inserisciAlCentro(int[] a, int v)` che:

- ha come parametri un array **a** non nullo di interi e un valore intero **v**;
- crea e restituisce un nuovo array di interi **b** ottenuto da **a** inserendo, nella posizione centrale, un nuovo elemento uguale a **v**.

Ad esempio:



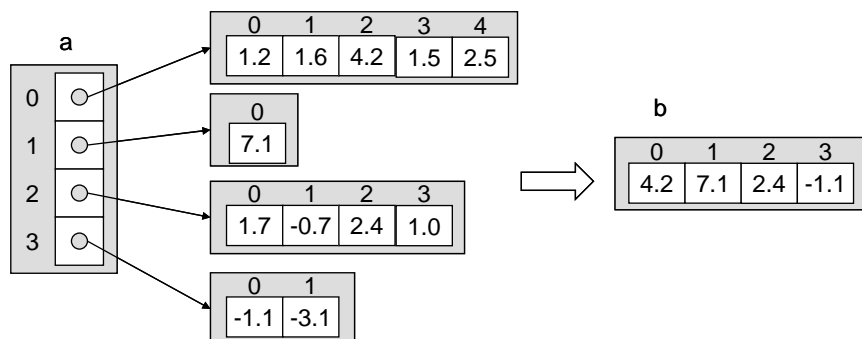
Si discuta esplicitamente il modo in cui si vuole gestire il caso in cui l'array **a** abbia lunghezza dispari anziché pari.

Esercizio A2

Scrivere un metodo `double[] massimi(double[][] a)` che, ricevendo come parametro un array di array di reali **a** non nullo in cui ciascun elemento è un array di reali non nullo e non vuoto, calcola un nuovo array di reali **b** con le seguenti caratteristiche: (i) **b** contiene un elemento in corrispondenza a ciascuna riga di **a** (dunque, la lunghezza di **b** è pari al numero di righe di **a**); (ii) ogni elemento di **b**, che è definito in corrispondenza ad una certa riga di **a**, è uguale al massimo tra gli elementi di quella riga.

Scrivere e usare anche un metodo di supporto che calcola il massimo di un array di reali.

Ad esempio:



Infatti:

- il massimo tra cinque elementi della prima riga di **a** vale 4.2