

Laboratorio di Programmazione
Corso di laurea triennale in Informatica Musicale
Università degli Studi di Milano, A.A. 2016-2017

Nicola Basilico, Andrea Lanzi

Appello del 15 Settembre 2017

- L'esame ha una durata di 3 ore e 30 minuti.
- È possibile consultare il libro di testo, appunti e l'API Java.
- È possibile utilizzare qualsiasi ambiente di sviluppo installato sulla macchina.
- È proibito l'accesso ad Internet con qualsiasi mezzo.
- Ogni esercizio consegnato non deve generare errori di compilazione.
- Per ottenere la sufficienza è necessario consegnare almeno due esercizi corretti.

1 Array palindromo (case insensitive)

nome del file sorgente: ArrayPalindromo.java

Si scriva un programma che letto in input un array di 11 caratteri comunichi all'utente se è palindromo oppure no. Il programma dovrà essere *case insensitive* (non farà distinzione tra una lettera minuscola e la sua versione maiuscola).

1.1 Esempio

Dato in input un array con questi 11 caratteri:

```
A b C d E Z e d c B a
```

Il programma stampa:

```
l'array è palindromo
```

Dato in input un array con questi 11 caratteri:

```
A b C d E Z e d B c a
```

Il programma stampa:

```
l'array non è palindromo
```

2 Generazione di classifiche

nome del file sorgente: generaClassifiche.java

Un registro punteggi contiene un elenco non ordinato di 5 squadre, ciascuna con il proprio punteggio associato. Ad esempio:

```
Squadra2 15  
Squadra4 21  
Squadra1 76  
Squadra5 33  
Squadra3 23
```

Si scriva un programma che acquisisca due registri punteggi e stampi:

- la classifica delle squadre del primo registro;
- la classifica delle squadre del secondo registro;
- la classifica complessiva che considera tutte le squadre di entrambi i registri.

Nel caso in cui due o più squadre abbiano lo stesso punteggio si applichi un ordine arbitrario tra di esse.

2.1 Esempio

Dati il registro riportato sopra e questo secondo:

```
Squadra9 141
Squadra8 23
Squadra7 67
Squadra6 99
Squadra10 1
```

Il programma stampa:

```
classifica registro 1:
```

```
1) Squadra1
2) Squadra5
3) Squadra3
4) Squadra4
5) Squadra2
```

```
classifica registro 2:
```

```
1) Squadra9
2) Squadra6
3) Squadra7
4) Squadra8
5) Squadra10
```

```
classifica complessiva:
```

```
1) Squadra9
2) Squadra6
3) Squadra1
4) Squadra7
5) Squadra5
6) Squadra3
7) Squadra8
8) Squadra4
9) Squadra2
10) Squadra1
```

3 Elimina parole doppie

Si scriva un programma che data una frase in input riscriva la stessa frase eliminando le parole doppie consecutive.

3.1 Suggerimento

Si utilizzi la classe *StringTokenizer* e i seguenti metodi da essa forniti: *hasMoreTokens* e *nextToken* che funzionano in questo modo:

```
StringTokenizer stk = new StringTokenizer(s, delim)
// stk ora contiene le sottostringhe (token) della stringa s separate dala stringa delim

while(stk.hasMoreTokens()){ // stk.hasMoreTokens() è true se ci sono token rimanenti

    // preleva il prossimo token
    String token = stk.nextToken();

    // elabora il token
    // ...
}
```

3.2 Esempio di esecuzione

Il programma inizia stampando:

Inserire una frase:

inserendo la frase:

ciao ciao ciao mamma guarda guarda come mi mi mi mi mi mi diverto

il programma stampa:

ciao mamma guarda come mi diverto

inserendo la frase:

sopra la panca la capra campa

il programma stampa:

sopra la panca la capra campa

4 Calcolo tempi della maratona

nome del file sorgente: maratona.java

Si scriva un programma che acquisisca i tempi di percorrenza di una mezza maratona (21 km) dei primi 10 runner. Per ogni runner si hanno due tempi: il tempo di percorrenza del primo settore lungo 12,5 Km e il tempo di percorrenza del secondo settore lungo 8,5 Km.

4.1 Esempio:

```
runner1 60m 45m
runner2 70m 30m
runner3 45m 25m
runner4 70m 15m
runner5 30m 10m
runner6 27m 36m
runner7 100m 25m
runner8 27m 80m
runner9 58m 79m
runner10 23m 13m
```

Si calcoli per ogni runner la velocità media totale in km/h dell'intera gara e il tempo medio in minuti impiegato per correre 1Km. Dopodiché stampare a video la classifica dei primi 3 runner più veloci con i dati velocità media e ritmo (e.g, 5m/Km)

4.2 Output finale riferito dell'esempio precedente

Dati runner dell'esempio:

```
runner1 VelMedia: 12Km/h      ritmo: 5 m/Km
runner2 VelMedia: 12,6 Km/h   ritmo: 4.46 s (m/Km)
runner3 Velmedia: 18,10 Km/h  ritmo: 3.05 s (m/Km)
runner4 Velmedia: 14,89 Km/h  ritmo: 4,02 s (m/Km)
runner5 Velmedia: 31,50 Km/h  ritmo: 1,54 s (m/Km)
runner6 Velmedia: 20 Km/h     ritmo: 3,00 s (m/Km)
runner7 Velmedia: 10 Km/h     ritmo: 5,57 s (m/Km)
runner8 Velmedia: 11,6 Km/h   ritmo: 5,05 s (m/Km)
runner9 Velmedia: 9,13 Km/h   ritmo: 6,31 s (m/Km)
runner10 Velmedia: 35Km/h     ritmo: 1,42 s (m/Km)
```

4.3 Classifica Finale

```
runner10 Velmedia: 35 Km/h     ritmo: 1,42 s (m/Km)
runner5 Velmedia: 31,50 Km/h   ritmo: 1,54 s (m/Km)
runner6 Velmedia: 20 Km/h     ritmo: 3,00 s (m/Km)
```