Laboratorio di Programmazione
(Corso di Laurea in Informatica)

ESAME del 22 Giugno 2017

Avvertenze

- VERRANNO CORRETTI SOLO E SOLTANTO I COMPITI IL CUI ESERCIZIO FILTRO FUNZIONA PERFETTAMENTE
- I programmi realizzati DEVONO rispettare le specifiche funzionali formite (input/output).
- Nello svolgimento dell'elaborato è possibile usare qualunque classe delle librerie standard di Java.
- Si raccomanda di salvare, compilare e fare upload delle soluzioni con una certa regolarità.
- Per la procedura di **consegna** si veda in fondo al documento.
- ATTENZIONE! Rispettare le specifiche dell'output! La correzione avviene in automatico mediante script che verificano la forma dell'output, aggiungere messaggistica ulteriore o utilizzare termini e/o spaziatura differenti significa invalidare il proprio lavoro.

Esercizio Filtro: *lunghezze di parole*

Descrizione

Scrivere una applicazione (una classe "LunghezzeParole" dotata del metodo `main`) che, dato un certo numero (non noto a priori) di parole su standard input, visualizzi su standard output il numero minimo, medio (troncato all'intero) e massimo di caratteri presenti nelle parole in input.

Vincoli

Si assuma che l'input contenga almeno una riga di testo non vuota. L'input termina con *fine file* (i.e., in sistemi Linux la pressione simultanea dei tasti Ctrl e d su riga vuota se provate il programma da prompt).
Il numero di parole (stringhe separate da caratteri di spaziatura, cioè newline, spazi, tab, ecc.) presenti in input non è noto a priori e può essere disposto su più righe (su un numero arbitrario di righe).
Il file sorgente consegnato DEVE chiamarsi `LunghezzeParole.java` e contenere la classe **pubblica** `LunghezzeParole` con un `main` invocabile con: `$ java LunghezzeParole`

Esempi

Ad esempio se l'input fosse

```
dfgr  ciao aiuola
aaa   paolo     dinosauro
```

l'output dovrebbe essere il seguente:

```
max: 9
med: 5
min: 3
```

Se l'input invece fosse:

```
pippo pluto

paperino

gastone paperone
```

l'output dovrebbe essere il seguente:

```
max: 8
med: 6
min: 5
```

---

1 Conteggio di numeri

1.1 Descrizione

Scrivere un programma (una classe "ContaNumeri" dotata del metodo `main`) che legge da standard input numeri (positivi e negativi) interi o con la virgola, inframmezzati a parole varie, separati da caratteri di spaziatura (cioè newline, spazi, tab, ecc.) e manda in output una statistica di quanti numeri interi e quanti con la virgola sono stati effettivamente inseriti, e null'altro (si veda l'esempio sotto). Il numero di parole e numeri non è noto a priori.

1.2 Vincoli

Non ci sono vincoli sul numero di parole e numeri su una riga, né sul numero di righe, né sul numero di caratteri di spaziatura (cioè newline, spazi, tab, ecc.) che li separano.
Il file sorgente consegnato DEVE chiamarsi `ContaNumeri.java` e contenere la classe **pubblica** `ContaNumeri`.

1.3 Esempi

Ad esempio, un input tipo:

```
1243 aaa   -3.5 bb c54

a.b 3-2  a-6


1276767.7
aslkdjaslk

-92873981 76237263.989898
```

deve visualizzare il seguente output:

```
interi: 2
floating: 3
```

---

2 Parole in ordine

2.1 Descrizione

Scrivere un programma (una classe "SortParole" dotata del metodo `main`) che legga da un file, il cui nome è passato per parametro sulla linea di comando, le parole in esso contenute e le visualizzi in ordine alfabetico, una per riga (si veda l'esempio sotto). Il numero di parole non è noto a priori e le parole sono separate da caratteri di spaziatura (newline, spazi, tab, ecc.).

2.2 Esempi

Ad esempio, se il file *brownFox* contiene:

```
the quick brown fox jumps
over the

lazy dog
```

Eseguendo

```
$ java SortParole brownFox
```

si dovrà visualizzare:

```
brown
dog
fox
jumps
lazy
over
quick
the
the
```

2.3 Vincoli

Non implementate un vostro algoritmo di ordinamento.
Il file sorgente consegnato DEVE chiamarsi `SortParole.java` e contenere la classe **pubblica** `SortParole`.

---

3 PoorMansHash

3.1 Descrizione

Realizzare un programma (una classe "PoorMansHash" dotata del metodo `main`) che riceve una stringa di caratteri passata per parametro sulla linea di comando e manda in standard output una stringa di esattamente 15 caratteri, calcolata dal seguente algoritmo:

- si inizializzi la stringa di ouput con tutti spazi (ricordiamo che il codice ASCII dello spazio è 32)
- per ogni posizione i della stringa di ingresso
  - si divida per 3 (troncando all'intero) il valore ASCII del carattere in quella posizione,
  - si sommi il valore così ottenuto al k-esimo carattere della stringa di output, dove k è calcolato come "i modulo 15";
  - si assegni come (nuovo) valore del carattere di output il modulo 95 del valore calcolato al punto precedente, sommato a 33;
- si visualizzi infine la stringa formata dai caratteri corrispondenti ai codici ASCII così calcolati, e null'altro (si veda l'esempio sotto).

3.2 Vincoli

La stringa in input può avere un numero arbitrario di caratteri.
Il file sorgente consegnato DEVE chiamarsi `PoorMansHash.java` e contenere la classe **pubblica** `PoorMansHash`.

3.3 Esempi

```
$ java PoorMansHash supercalifragilistichespiralidoso
385GJFDMKHOCIIH

$ java PoorMansHash ciao
bdaf
```

3.4 Osservazioni

In informatica un 'hash' è una funzione che mappa una stringa di byte di lunghezza arbitraria in una stringa di byte di lunghezza predefinita che dipende dal contenuto della stringa originale. In questo esercizio non si ha nessuna pretesa di creare un algoritmo esente da collisioni.

Consegna

Si ricorda che le classi devono essere tutte *public* e che vanno consegnati tutti (e soli) i file *.java* prodotti. NON vanno consegnati i *.class*. Per la consegna, eseguite l'upload dei SINGOLI file sorgente (NON un file archivio!) dalla pagina web: http://upload.di.unimi.it nella sessione che vi è stata indicata.

*** ATTENZIONE!!! ***

NON VERRANNO VALUTATI GLI ELABORATI CON ERRORI DI COMPILAZIONE O LE CONSEGNE CHE NON RISPETTANO LE SPECIFICHE (ad esempio consegnare un archivio zippato è sbagliato)
UN SINGOLO ERRORE DI COMPILAZIONE O DI PROCEDURA INVALIDA **TUTTO** L'ELABORATO.

Per ritirarsi fare l'upload di un file vuoto di nome `ritirato.txt`.
