Laboratorio di Programmazione
(Corso di Laurea in Informatica)

ESAME del 13 Febbraio 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).
- 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.

---

1 Esercizio Filtro: *reverse+reverse*

1.1 Descrizione

Realizzare un programma `RevRev.java` che legga un certo numero (non noto a priori) di linee di testo e le riproponga in output in ordine inverso (la prima riga per ultima e così via) e ogni riga ribaltata (il primo carattere per ultimo e così via). Ad esempio, dato l'input:

```
pippo        pluto

gastone
paperino
qui paperone
```

Si deve ottenere l'output:

```
enorepap iuq
onirepap
enotsag

otulp        oppip
```

E NULL'ALTRO! Non inserire altri messaggi all'utente, la correzione avviene per confronto diretto con l'output atteso!

1.2 Vincoli e condizioni

Si assuma che l'input contenga almeno una riga di testo non vuota. Il numero di linee di testo in ingresso "non noto a priori" implica che l'input può constare anche di decine, centinaia, migliaia, etc. di righe.
Il file sorgente consegnato DEVE chiamarsi `RevRev.java` e contenere la classe **pubblica** `RevRev` con un `main` invocabile con:
`$ java RevRev`

---

2 Sistema di numerazione n-ario

2.1 Descrizione

Si consideri il sistema di numerazione posizionale S3 in base 3 avente come cifre i tre simboli

```
z u d
```

i cui valori sono rispettivamente:

- val(z) = 0
- val(u) = 1
- val(d) = 2

Realizzare un programma `S3ToInt.java` che, dato un numerale *num* di S3 come argomento sulla linea di comando, stampi il valore di *num*. Il programma deve controllare che l'argomento sia un numerale in S3 e stampare "input non valido" nel caso non lo sia.

2.2 Vincoli

L'input al programma è un unico argomento da linea di comando.

2.3 Esempi

```
$ java S3ToInt uzz
9
$ java S3ToInt dddd
80
$ java S3ToInt duz
21
$ java S3ToInt ciao
input non valido
$ java S3ToInt udzzzduuz
11001
```

2.4 Osservazioni

Si ricorda che un sistema di numerazione si dice posizionale se i simboli (cifre) usati per scrivere i numeri assumono valori diversi a seconda della posizione che occupano. Il nostro sistema di numerazione è un sistema posizionale in base 10.

---

3 Giorno dell'anno

3.1 Descrizione

Scrivere un programma `CalcolaGiorno.java` che riceve da linea di comando una data nel formato

gg mm aaaa

dove i tre argomenti sono valori interi separati da spazi, corrispondenti rispettivamente al giorno, mese, anno. Il programma deve produrre in output il numero corrispondente alla data indicata, assumendo che il 1 gennaio corrisponda a 1, il 31 gennaio a 31, il 1 febbraio a 32, etc. Precisamente il programma dovrà produrre il seguente output:

- esclusivamente il valore numerico corrispondente alla data, se la data è espressa in forma corretta;
- esclusivamente la stringa "errore" in ciascuno dei seguenti casi:
  - se uno dei tre campi della data non è un valore intero (si ricordi che i metodi usati per convertire una stringa in un intero sollevano una eccezione nel caso la stringa non sia nel formato atteso)
  - se la data specificata non esiste, ad esempio: 29 2 1900, 32 12 2000, 29 2 2017, oppure 11 13 1964.

Si ricorda infine che un anno è bisestile se soddisfa una delle seguenti condizioni:

- è un anno non secolare divisibile per 4 (ad es. 1964);
- è un anno secolare divisibile per 400 (ad es. 2000).

3.2 Vincoli

Si assuma che l'argomento corrispondente all'anno, se è un numero intero, corrisponda a un anno valido (non va cioè effettuato alcun controllo sul valore corrispondente all'anno).

3.3 Esempi

```
$java CalcolaGiorno  29  2  1964
60
$java CalcolaGiorno  29  2  2017
errore
$java CalcolaGiorno  1   3  2017
60
$java CalcolaGiorno  1   marzo  2017
errore
```

3.4 Suggerimenti

Conviene definire un metodo statico che, dato un anno, restituisca `true` se l'anno è bisestile e false altrimenti e un metodo statico che dati un mese ed un anno, restituisca il numero di giorni del mese specificato (l'anno serve per determinare i giorni di Febbraio).

---

4 Campo minato

4.1 Descrizione

Campo minato è un ben noto gioco di strategia in cui il giocatore deve individuare la posizione di alcune mine in un campo di battaglia rappresentato come una matrice di possibili posizioni. Un ingrediente fondamentale del gioco è determinare, per ogni posizione del campo, quante mine si trovino in posizioni adiacenti ad essa; per ciascuna posizione si considerano adiacenti le (al piú) otto posizioni che si ottengono spostandosi di una riga e/o di una colonna nella matrice. Ad esempio, nella matrice

```
........
..BBB...
..BAB...
..BBB...
........
```

le otto posizioni adiacenti a quella marcata con la lettera A sono quelle marcate con la lettera B; evidentemente, le posizioni sui bordi hanno meno di otto posizioni adiacenti.

Realizzare un programma `CampoMinato.java` che, dato su linea di comando un valore intero n (la dimensione di un campo minato quadrato), legge da standard input n righe di n caratteri ciascuna che rappresentano un campo minato, dove le posizioni libere sono rappresentate da un punto e le mine da un asterisco. Il programma non deve fare nessun controllo sulla correttezza dell'input. Il programma deve stampare una matrice che abbia un asterisco in corrispondenza di ogni mina e, in corrispondenza delle posizioni senza mine, un numero (compreso tra 0 e 8) pari a quante mine sono contenute nelle posizioni adiacenti del campo.
Ad esempio, dato in ingresso il campo minato

```
*...
....
.*..
....
```

va stampata la matrice

```
*100
2210
1*10
1110
```

4.2 Osservazioni

Per capire meglio l'output, consideriamo la matrice

```
XX..
YX..
XX..
....
```

Il 2 presente nella soluzione nella posizione corrispondente alla lettera Y indica il fatto che nelle posizioni adiacenti, marcate con la lettera X, ci sono complessivamente 2 mine.

4.3 Vincoli

Il campo minato ha dimensione massima pari a 100 x 100. Il programma non deve fare nessun controllo sulla correttezza dell'input.

4.4 Esempio

Eseguendo

```
$java CampoMinato 8
```

su input

```
.*.*.*.*
........
***.....
.......*
........
..**....
..**....
.......*
```

il programma emette

```
1*2*2*2*
34422121
***10011
2321001*
01221011
02**2000
02**2011
0122101*
```

4.5 Suggerimenti

Per contare il numero di mine adiacenti a una posizione (r, c) conviene definire un metodo (statico) che, data una matrice M di caratteri e due interi i e j, restituisca:

- 1 se i corrisponde a una riga di M, j corrisponde a una colonna di M e nella posizione (i, j) di M c'è un asterisco;
- 0 altrimenti.

Conviene inoltre definire un altro metodo statico che, data una matrice M di caratteri e due interi r e c, restituisca il numero di mine presenti nelle celle adiacenti alla cella di posizione (r, c).

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`.
