

Laboratorio di Programmazione

(Corso di Laurea in Informatica)

APPELLO del 15 Giugno 2018

Avvertenze

- VERRANNO CORRETTI SOLO E SOLTANTO I COMPITI IL CUI ESERCIZIO FILTRO FUNZIONA PERFETTAMENTE
- I programmi realizzati DEVONO rispettare le specifiche fornite sull'input e sulla forma dell'output, senza aggiungere messaggistica ulteriore o utilizzare termini e/o spaziatura differenti. Parte della valutazione è infatti sulla comprensione e sul rispetto delle specifiche.
- 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. In ogni caso NON sarà riaperta la sessione di upload per chi non ha consegnato in tempo.
- Per la procedura di **consegna** si veda in fondo al documento.

1 ESERCIZIO FILTRO

1.1 Descrizione

Scrivere un programma (la classe si DEVE chiamare “Filtro” ed essere pubblica) che, dato un intero n non negativo su linea di comando, emetta sul flusso di uscita il disegno di una “capanna” la cui altezza (in numero di righe) è pari a n . Nel caso n sia uguale a 0, il programma non deve stampare niente.

1.2 Vincoli

Si assuma che il numero e il formato degli argomenti sulla linea di comando sia quello atteso, cioè un intero. Non è richiesto quindi che venga fatto un controllo.

1.3 Esempi

Caso degenere:

```
$ java Filtro 0
```

Caso altezza 1:

```
$ java Filtro 1
/\
```

Caso altezza 4:

```
$ java Filtro 4
  /\
 //  \
//    \
//      \
```

2 Conta lunghezze

2.1 Descrizione

Scrivere un programma (la classe si deve chiamare “ContaLunghezze” ed essere pubblica) che, letta dal flusso di input standard una sequenza di stringhe, separate tra loro da un numero arbitrario di caratteri di spaziatura, calcoli il numero di occorrenze delle relative lunghezze, riportando in output solo le frequenze diverse da 0.

Si ricorda che la fine del flusso di input standard in sistemi Linux è segnalata dalla pressione dei tasti CTRL+D.

2.2 Vincoli

La sequenza in input potrà essere arbitrariamente lunga e disposta su più righe (sono ammesse anche righe vuote). Si assuma che la lunghezza massima di una stringa sia 100.

2.3 Esempio

```
$ java ContaLunghezze
io sono quasi sempre
    allegro
```

```
anche se talvolta
    perdo la pazienza precipitevolissimevolmente
ma questo succede a molti e non ci posso
```

```
assolutamente
```

```
fare nulla
1 : 2
2 : 5
3 : 1
4 : 2
5 : 6
6 : 2
7 : 2
8 : 2
13 : 1
26 : 1
```

Nota bene: le righe dopo il comando `java ContaLunghezze` da “io...” fino a “... nulla” sono il testo inserito in standard input, il risultato del programma inizia dal successivo “1 : 2”. Volendo provare il programma più rapidamente si può creare un file di testo, ad es. un *input.txt*, e poi “somministrarlo” al programma `ContaLunghezze` usando la redirectione in input, i.e.:

```
$ java ContaLunghezze < input.txt
```

3 Giorno dopo

Scrivere un programma (la classe si deve chiamare “Update” ed essere pubblica) che, letti da linea di comando tre numeri che rappresentano una data (nel formato gg mm aaaa), produca in output la data corrispondente al giorno successivo.

Nel caso la data non sia valida, si produca il messaggio di errore “data non valida”.

Nel caso il numero e il formato degli argomenti sulla linea di comando non sia quello atteso (tre interi), si produca il messaggio di errore “input non valido”.

La gestione degli anni bisestili¹ è facoltativa; nel caso sia presente, costituirà un “bonus” ai fini della valutazione.

3.1 Esempi

```
$ java Update 15 6 2018
16 6 2018
```

```
$ java Update 30 6 2018
1 7 2018
```

```
$ java Update 31 6 2018
data non valida
```

```
$ java Update 12 marzo 2018
input non valido
```

```
$ java Update 12 6
input non valido
```

```
$ java Update 31 12 2018
1 1 2019
```

Nel caso si tenga conto degli anni bisestili

```
$ java Update 28 2 2016
29 2 2016
```

¹Si ricorda che un anno è bisestile se e solo 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).

Quindi ad esempio non è bisestile l’anno 1800, anno secolare non divisibile per 400.

4 Tartaglia

4.1 Descrizione

Nel triangolo di Tartaglia la prima riga è costituita da un solo elemento pari a 1 e la seconda da due elementi, ciascuno pari a 1. In ogni riga successiva, il primo e l'ultimo elemento sono pari a 1 e ogni altro elemento è dato dalla somma dei due elementi adiacenti della riga precedente. Quindi, per esempio, la terza riga è 1 2 1, dove il valore 2 è dato dalla somma dei due valori 1 sulla riga precedente, uno adiacente a sinistra e uno adiacente a destra rispetto a 2; la sesta riga è 1 5 10 10 5 1, dove il valore 10 è dato dalla somma del valore 4, adiacente a destra, e del valore 6, adiacente a sinistra, nella riga precedente.

Il triangolo per le prime sei righe è:

```
      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
 1 5 10 10 5 1
```

Scrivere un programma (una classe pubblica “Tartaglia”) dotata di un metodo ricorsivo

```
public static int[] tartaglia (int n)
```

che calcola la n -esima riga del triangolo di Tartaglia per n numero naturale > 0 .

Il programma legge un numero intero $n > 0$ da linea di comando e produce in output le prime n righe del triangolo di Tartaglia.

4.2 Vincoli

Si assuma che il numero e il formato degli argomenti sulla linea di comando sia quello atteso, cioè un intero > 0 . Non è richiesto quindi che venga fatto un controllo.

4.3 Esempi

```
$ java Tartaglia 7
[1]
[1, 1]
[1, 2, 1]
[1, 3, 3, 1]
[1, 4, 6, 4, 1]
[1, 5, 10, 10, 5, 1]
[1, 6, 15, 20, 15, 6, 1]
```

```
$ java Tartaglia 2
[1]
[1, 1]
```

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 ESERCIZI CON ERRORI DI COMPILAZIONE O LE CONSEGNE CHE NON RISPETTANO LE SPECIFICHE (ad esempio consegnare un archivio zippato è sbagliato)

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