

Esercitazioni di Laboratorio

5. Laboratorio 5

Esercizio 0

Presentazioni dei Tutor Junior:

- 4) [Good practices](#)

Esercizio 1

Argomento: cicli

Modificare il programma dell'esercizio 10 del laboratorio precedente in modo che stampi i numeri incolonnati in 10 colonne. Provare a stampare i primi 200 numeri.

L'esecuzione della classe genera a Standard Output un risultato simile al seguente:

```
$ java NewLineNumberLister
***STAMPA DEI NUMERI DA 1 A N***
N?: 200
1 2 3 4 5 6 7 8 9 10
11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30
31 32 33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48 49 50
51 52 53 54 55 56 57 58 59 60
61 62 63 64 65 66 67 68 69 70
71 72 73 74 75 76 77 78 79 80
81 82 83 84 85 86 87 88 89 90
91 92 93 94 95 96 97 98 99 100
101 102 103 104 105 106 107 108 109 110
111 112 113 114 115 116 117 118 119 120
121 122 123 124 125 126 127 128 129 130
131 132 133 134 135 136 137 138 139 140
141 142 143 144 145 146 147 148 149 150
151 152 153 154 155 156 157 158 159 160
161 162 163 164 165 166 167 168 169 170
171 172 173 174 175 176 177 178 179 180
181 182 183 184 185 186 187 188 189 190
191 192 193 194 195 196 197 198 199 200
```

Esercizio 2

Argomento: cicli e manipolazione di stringhe

Modificare il programma precedente in modo che stampi i numeri incolonnati in 10 colonne allineati a destra (unita' sotto unita', decine sotto decine, centinaia sotto centinaia). Provare a stampare i primi 200 numeri.

L'esecuzione della classe genera a Standard Output un risultato simile al seguente:

```
$ java NewLineNumberLister
***STAMPA DEI NUMERI DA 1 A N***
```

N?: 200

```
1  2  3  4  5  6  7  8  9 10
11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30
31 32 33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48 49 50
51 52 53 54 55 56 57 58 59 60
61 62 63 64 65 66 67 68 69 70
71 72 73 74 75 76 77 78 79 80
81 82 83 84 85 86 87 88 89 90
91 92 93 94 95 96 97 98 99 100
101 102 103 104 105 106 107 108 109 110
111 112 113 114 115 116 117 118 119 120
121 122 123 124 125 126 127 128 129 130
131 132 133 134 135 136 137 138 139 140
141 142 143 144 145 146 147 148 149 150
151 152 153 154 155 156 157 158 159 160
161 162 163 164 165 166 167 168 169 170
171 172 173 174 175 176 177 178 179 180
181 182 183 184 185 186 187 188 189 190
191 192 193 194 195 196 197 198 199 200
```

Esercizio 3

Argomento: cicli e manipolazione di stringhe

Leggere da standard input una riga di testo. Inviare a standard output una corrispondente riga in cui i caratteri sono disposti in ordine inverso.

Esempio:

input: uno due tre

output: ert eud onu

Esercizio 4

Argomento: cicli e manipolazione di stringhe

Leggere da standard input un numero arbitrario di parole, una per ogni riga. Inviare a standard output una riga di testo unica in cui i caratteri di ciascuna parola sono disposti in ordine inverso.

Esempio:

input:

uno

due

tre

output: onu eud ert

Esercizio 5

Argomento: cicli e manipolazione di stringhe

Leggere da standard input una stringa definita sull'alfabeto del DNA (A,C,G,T). Dopo aver verificato che non siano presenti altri caratteri, inviare a standard output una la stringa "inversa-complementare" cosi' definita:

- Ad ogni A della stringa originale corrisponde una T;
- Ad ogni T della stringa originale corrisponde una A;
- Ad ogni C della stringa originale corrisponde una G;

- Ad ogni G della stringa originale corrisponde una C;

la stringa viene invertita (l'ultimo carattere e' il primo).

Esempio:

input: ACCTGGTAC

output: GTACCAGGT

Esercizio 6

Argomento: lettura da standard input e cicli

Progettare il programma SomeNumbers che chieda all'utente di inserire una sequenza di numeri e ne visualizzi:

la somma

la somma dei valori assoluti

il prodotto

il valore medio

Il programma non può far uso di sentinelle né può chiedere all'utente quanto sarà lunga la sequenza: deve continuare a leggere numeri fino a quando non viene chiuso il flusso di ingresso.

Il programma deve visualizzare un messaggio d'errore (e niente altro) se il flusso viene chiuso prima che siano stati introdotti almeno due numeri.

Esercizio 7

Argomento: lettura da standard input e cicli

Progettare il programma StandardDeviation che chieda all'utente di inserire una sequenza di numeri e ne visualizzi la deviazione standard.

La deviazione standard D si calcola con la formula seguente:

$$D = \text{Math.sqrt}((A - B*B/n)/(n-1))$$

dove n è il numero di valori, B è la somma di tutti i valori e A è la somma dei quadrati di tutti i valori. La formula è valida soltanto per $n > 1$ (per $n = 1$ la deviazione standard vale 0, per definizione).

Il programma non può far uso di sentinelle né può chiedere all'utente quanto sarà lunga la sequenza: deve continuare a leggere numeri fino a quando non viene chiuso il flusso di ingresso.

Il programma deve visualizzare un messaggio d'errore (e niente altro) se il flusso viene chiuso prima che siano stati introdotti almeno due numeri.

Esercizio 8

Argomento: lettura da standard input, decisioni e cicli

Scrivere il programma EuclideMCD che calcoli il massimo comun divisore (MCD) di due numeri interi positivi m e n acquisiti da standard input e lo visualizzi sul flusso di uscita standard.

Si usi il noto Algoritmo di Euclide per il calcolo del MCD di due numeri interi positivi m ed n (con $m > n$):

Finchè il resto della divisione di m per n è diverso da zero

- il nuovo valore di m è uguale al precedente valore di n
- il nuovo valore di n è uguale al resto della divisione del precedente valore di m per il precedente valore di n

Il MCD è l'attuale valore di n

Esercizio 9

Argomento: Manipolazione di stringhe e caratteri, cicli annidati

Scrivere un programma che

- chiede all'utente di introdurre due stringhe (una per riga), **s1** e **s2**; ciascuna stringa è costituita da tutti i caratteri presenti sulla riga, compresi eventuali spazi iniziali, finali e/o intermedi
- verifica se la seconda stringa, **s2**, è una sottostringa di **s1**, ossia se esiste una coppia di numeri interi, x e y, per cui **s1.substring(x, y)** restituisce una stringa uguale a **s2** (al termine della verifica viene visualizzato un messaggio opportuno)

Soluzione 1: Il programma può usare solo i metodi `length()`, `substring(...)` ed `equals(...)` della classe `String`.

Soluzione 2: Il programma può usare solo i metodi `charAt()` e `length()` della classe `String`.

Verificare che il programma gestisca correttamente la situazione in cui **s2** è la stringa vuota (che, in base alla definizione precedente, è sottostringa di qualsiasi stringa).

Esercizio 10

La zia Bettina è appassionata di viaggi e di elaborazione delle immagini. Durante il suo ultimo viaggio ti ha mandato tre cartoline da 3 posti diversi ma per farti uno scherzo ha distorto le immagini. Utilizzate la classe [SimpleImage.java](#) fornita e lo scheletro di una classe di decodifica [DecodelImage.java](#) per ottenere le immagini originali. Per far ciò completa la classe `DecodelImage.java`, rinominandola opportunamente in `DecodelImageIron.java`, `DecodelImageCopper.java`, `DecodelImageWest.java` a seconda dell'immagine da decodificare.

Decodifica quindi le immagini con le istruzioni che trovi più sotto per scoprire le tappe del viaggio di zia Bettina scrivendo il codice necessario a "disfare" le distorsioni. Il codice già scritto in `DecodelImage.java` permette di leggere il nome del file da decodificare e il nome del file da scrivere in output eseguendo il programma così:

```
java DecodelImage nome_file_input nome_file_output
```

Ad esempio:

```
java DecodelImageIron iron-puzzle.png iron-real.png
```

Se hai agito correttamente, aprendo il file `iron-real.png` apparirà l'immagine originale.

Le slide mostrate a lezione su come vengono rappresentate le immagini nei computer sono presenti tra il [Materiale di approfondimento](#) nella pagina principale del sito moodle del corso. Le immagini e le classi sono scaricabili dalla solita cartella "file utili e soluzioni degli esercizi - Lab 5" e cliccando sul nome del file.

L'immagine [iron-puzzle.png](#) contiene l'immagine di qualcosa di famoso, ma l'immagine è stata distorta. L'oggetto famoso è nei valori di rosso, tuttavia i valori di rosso sono stati tutti divisi per 10, quindi sono tutti più piccoli di un fattore 10. I valori di blue e verde sono valori casuali senza significato ("rumore") aggiunti per oscurare l'immagine reale. Il tuo compito è "disfare" tutte queste distorsioni per rivelare l'immagine reale e scoprire la prima tappa del viaggio di zia Bettina. Innanzitutto metti tutti i valori di blu e verde a 0, per eliminarli. Guarda il risultato... se guardi bene puoi già vedere la vera immagine, anche se è molto scura. Quindi moltiplica ogni valore di rosso per 10 ripristinandolo al valore originale. Di che oggetto famoso si tratta?

L'immagine [copper-puzzle.png](#) contiene l'immagine di qualcosa di famoso, ma l'immagine è stata distorta. L'oggetto famoso è nei valori di blu e verde, tuttavia tali valori sono stati tutti divisi per 19, quindi sono tutti più piccoli di un fattore 19. I valori di rosso sono valori casuali senza significato ("rumore") aggiunti per oscurare l'immagine reale. Per scoprire la seconda tappa del viaggio devi "disfare" le distorsioni in modo analogo a quanto fatto prima. Di che oggetto famoso si tratta?

L'immagine [west-puzzle.png](#) contiene l'immagine di qualcosa di famoso, ma l'immagine è stata distorta. L'oggetto famoso è nei valori di blu, mentre i valori di rosso e verdi sono rumore casuale. Per scoprire l'ultima tappa del viaggio di zia Bettina, oltre a rimuovere il rumore dovrai operare come segue: se il valore blu è minore di 16, moltiplicalo per 16 per farlo tornare al suo valore originale. Se invece il suo valore è 16 o più, allora è rumore e deve essere messo a 0. Di che oggetto famoso si tratta?

Per rendere le immagini più realistiche puoi trasformarle in toni di grigio. Semplicemente attribuisce ai colori di ciascun pixel la media dei colori di quel particolare pixel. Puoi chiamare la classe che fa questo passaggio `MakeGreyImage.java`

Le cartoline della zia non sono bellissime coi toni di grigio perche' espresse nello spettro di uno o due colori. Provare a rendere in bianco e nero l'immagine [flowers.jpg](#) da mandare alla zia Bettina come ringraziamento per questo esercizio!

(ispirato da nifty assignment)