

Esercitazioni di Laboratorio

4. Laboratorio 4

Esercizio 0

Se non l'avete ancora fatto, provate a rispondere alle domande del [Quiz di Autovalutazione 1](#).

Esercizio 1

Argomento: uso dell'input standard; manipolazione di stringhe e numeri

Scrivere il programma `PrintTimeInterval.java` che legga dall'input standard due orari nel formato "24 ore", ciascuno di quattro cifre (ad esempio, 0900 oppure 1730), con il secondo orario successivo al primo visualizzi sull'output standard il numero di ore e di minuti (separatamente) che intercorrono fra i due orari secondo il seguente esempio

Inserire il primo orario: 0900

Inserire il secondo orario: 1730

Tempo trascorso: 8 ore e 30 minuti

Esercizio 2

Argomento: uso dell'input standard; manipolazione di stringhe e numeri

Scrivere il programma `PrintTimeInterval2.java`, modificando il programma `PrintTimeInterval` visto in precedenza in modo che funzioni correttamente anche se il secondo orario è inferiore al primo (cioè per un intervallo di tempo che comprenda la mezzanotte), come in questo esempio di funzionamento:

Inserire il primo orario: 1730

Inserire il secondo orario: 0900

Tempo trascorso: 15 ore e 30 minuti

Esercizio 3

Argomento: decisioni, confronto tra numeri reali

Dopo averne osservato il codice, compilare la classe [Numeric.java](#) e [Apprx.java](#). Eseguire quest'ultima e riflettere sui risultati.

Scrivere un programma che

- legga due numeri in virgola mobile
- visualizzi un messaggio che dice se i due numeri sono o meno approssimativamente uguali

Esercizio 4

Argomento: lettura da input, decisioni, confronto tra numeri

Progettare il programma `EvenNumber` che:

- chieda all'utente di fornire un numero pari (ovviamente intero...)
- se il numero fornito è dispari, chieda di nuovo all'utente di fornire un numero pari, dopo aver visualizzato una segnalazione d'errore
- se il numero fornito (al primo o secondo tentativo) è pari, scriva un messaggio di congratulazioni all'utente, riportando anche il numero fornito (es. "Bravo! Il numero 26 è pari")
- se, invece, neanche il secondo numero fornito è pari, scriva un messaggio di disappunto

Attenzione: se il primo numero fornito è pari, il programma NON deve chiedere un secondo numero.

Esercizio 5

Argomento: decisioni, confronto tra stringhe

Progettare il programma `SortThreeStrings` che

- chieda all'utente di inserire tre stringhe (una per riga)
- visualizzi le stringhe in ordine lessicografico crescente (una per riga)

Esercizio 6

Argomento: decisioni, confronto tra numeri

Progettare il programma `IsLeapYear` che segnali all'utente se il numero intero positivo che ha introdotto corrisponde a un anno bisestile oppure no.

Ricordare che un anno è bisestile se è divisibile per 4. Fanno eccezione gli anni divisibili per 100, che non sono bisestili, e gli anni divisibili per 400, che invece sono bisestili (pur essendo divisibili anche per 100): tali eccezioni esistono però solo dopo l'adozione del calendario gregoriano, che avvenne nel 1582.

Il programma NON dovrà accettare il dato in ingresso (visualizzando, quindi, un messaggio d'errore) quando è un numero intero non positivo.

Provare a risolvere l'esercizio prima senza usare gli operatori booleani (`&&`, `||` e `!`) e poi invece utilizzandoli. Suggerisco di disegnare un diagramma di flusso con le varie diramazioni per chiarirsi le idee prima di scrivere il codice!

Esercizio 7

Argomento: decisioni e operatori booleani

Scrivere un programma **`SimpleTriangleTester`**, che riceve da standard input tre numeri interi positivi che rappresentano le lunghezze dei lati di un triangolo e che invia a standard output una stringa contenente le seguenti informazioni:

- relativamente ai lati: equilatero, isoscele, scaleno
- relativamente agli angoli: acutangolo, rettangolo, ottusangolo

Esempi

- se vengono inseriti i numeri "3 4 5", il programma visualizzerà la stringa "triangolo scaleno rettangolo".
- se vengono inseriti i numeri "5 7 7", il programma visualizzerà la stringa "triangolo isoscele acutangolo".
- se vengono inseriti i numeri "5 3 3", il programma visualizzerà la stringa "triangolo isoscele ottusangolo".
- se vengono inseriti i numeri "3 3 3", il programma visualizzerà la stringa "triangolo equilatero" (non serve l'informazione relativa agli angoli perché i triangoli equilateri sono sempre acutangoli avendo tre angoli uguali pari a $\pi/3$).
- se vengono inseriti i numeri "3 4 8", il programma visualizzerà la stringa "non è un triangolo" (non sempre tre lati rappresentano un triangolo).

Alcuni suggerimenti sono disponibili [qui](#).

Esercizio 8

Argomento: decisioni e operatori booleani

Scrivere un programma che riceve da standard input la data di compleanno dell'utente (in formato giorno mese), e visualizza a standard output l'oroscopo corrispondente. Il programma deve gestire correttamente anche il caso in cui l'input non corrisponda al formato prescritto.

Esempio: Se viene inserito l'input "26 7" (ovvero, 26 luglio), il programma potrà visualizzare la seguente stringa

LEONE	
Amore:	4/5
Amicizia:	3/5
Lavoro:	3/5

Se invece l'input inserito non è interpretabile come una data in formato giorno mese (ad esempio, l'input "43 21", oppure l'input "32 1", oppure l'input "30 2"), il programma dovrà visualizzare la stringa

L'input inserito non e` una data.

Suggerimenti: le associazioni tra date di nascita e segni zodiacali possono essere reperite in rete, ad esempio [qui](#). Le stringhe contenenti gli oroscopi di ciascun segno possono venire inventate a vostro piacimento (come si fa solitamente con gli oroscopi...), l'unico requisito è che ciascuna di esse deve contenere il nome del segno corrispondente.

Esercizio 9

Argomento: cicli

Scrivere un programma che invii a standard output la successione dei numeri interi da 1 a n (n compreso), dove il numero n e' letto da Standard Input.
Provare a stampare i primi 200 numeri.

Casi particolari:

Se $n \leq 0$, inviare a Standard Output un messaggio del tipo: "Ingresso non ammesso".

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

```
$ java NumberLister
***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 10

argomento: cicli, metodi sulle stringhe

"In [crittografia](#), il **cifrario di Cesare** è uno dei più antichi [algoritmi crittografici](#) di cui si abbia traccia storica. È un [cifrario a sostituzione monoalfabetica](#), in cui ogni lettera del testo in chiaro è sostituita, nel testo cifrato, dalla lettera che si trova un certo numero di posizioni dopo nell'[alfabeto](#). Questi tipi di cifrari sono detti anche **cifrari a sostituzione** o **cifrari a scorrimento** a causa del loro modo di operare: la sostituzione avviene lettera per lettera, scorrendo il testo dall'inizio alla fine." [source: wikipedia.it].



Esempio di cifratura con spostamento di 3 a destra di 3 posizioni: La A diventa una D, la B una E, e così via...

Testo in chiaro	: X Y Z A B C D E F G H I J K L M N O P Q R S T U V W
Testo cifrato	: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Esempio di conversione
FONDAMENTI DI INFORMATICA
IRQGDPHQWL GL LQIRUPDWLFD

Progettare due programmi eseguibili EncodeCeasar.java e DecodeCeasar.java che codifichino e decodifichino un messaggio.

Il programma EncodeCeasar dovra' leggere in ingresso una stringa, trasformarla tutta in lettere maiuscole (per semplicita') e leggere un numero intero compreso tra 1 e 25. Il programma dovra' stampare in output la stringa codificata secondo il cifrario di Cesare (dovranno essere convertite solo le lettere e si assume che il cifrario sia ciclico, ovvero se devo codificare la lettera Z con scorrimento 4 il risultato sara' la lettera D.

Il programma DecodeCeasar dovra' leggere in ingresso una stringa codificata con il Cifrario di Cesare e un numero che indichi lo scorrimento e stampare in uscita il messaggio originale.

Alcuni suggerimenti sono disponibili [qui](#).