

Fondamenti di Informatica - a.a. 2022-2023
Prova pratica di programmazione del 5 luglio 2023

ESERCIZIO 1: Chi cerca (forse) trova! (10 punti)

Scrivere una classe Cerca.java che:

- chiede all'utente di introdurre due stringhe (una per riga), **superStringa** e **sottoStringa**; ciascuna stringa è costituita da tutti i caratteri presenti sulla riga, compresi eventuali spazi iniziali, finali e/o intermedi;
- verifica se la seconda stringa, **sottoStringa**, è una sottostringa di **superStringa**, ossia se esiste una coppia di numeri interi, x e y, per cui **superStringa.substring(x, y)** restituisce una stringa uguale a **sottoStringa** (al termine della verifica viene visualizzato un messaggio opportuno: nel caso sia una sottostringa, il programma comunica anche l'informazione relativa al numero di carattere da cui inizia).

Attenzione:

Il programma può usare, della classe **String**, i soli metodi: **charAt** e **length**

(in particolare, quindi, **non si può utilizzare** il metodo **substring** sopra menzionato).

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

È valutato **10 punti** se risponde alle specifiche, **0 punti** se presenta errori di compilazione o se fa uso del metodo **substring**.

Alla fine della prova il candidato lascerà nella directory di lavoro il file Cerca.java modificato. Il file prodotto dal candidato dovrà contenere il **nome e cognome**, matricola, data, numero della postazione.

ESERCIZIO 2: Problema di Giuseppe (punti 20)

Scrivere una Classe che simula il Problema (o Permutazione) di Giuseppe (*Josephus problem*). Il problema presenta n persone disposte in circolo in attesa di una esecuzione. Scelta una persona iniziale e un senso di rotazione, si saltano $k-1$ persone, raggiungendo così la k -esima persona, che viene giustiziata ed eliminata dal cerchio; di nuovo si saltano $k-1$ persone (degli $n-1$ rimasti) e si giustizia la k -esima persona. Le esecuzioni proseguono e il cerchio si restringe sempre più, finché non rimane che una sola persona, la quale viene graziata. Dati n e k , si chiede di realizzare una Classe in grado di determinare la posizione del sopravvissuto all'interno del cerchio iniziale.

La Classe

- Legge da standard input il numero intero positivo k ;
- Legge da un file di testo (il cui nome è passato come argomento sulla riga di comando) n stringhe, che rappresentano i nomi di n persone, e le inserisce in un oggetto di tipo **ArrayQueue**
- Simula la conta:
 1. la prima stringa inserita nella coda rappresenta la persona attualmente indicata, quindi a ogni incremento di un apposito contatore la prima stringa della coda viene disaccodata dalla cima della coda e riaccodata in fondo alla coda;
 2. in particolare, il programma effettua **$k-1$** operazioni di **dequeue** e **$k-1$** operazioni di **enqueue**;
 3. alla **k** -esima operazione di **dequeue**, la stringa disaccodata non viene riaccodata nella coda (la persona esce dal cerchio), e viene invece stampata a standard output;
- Ripete la simulazione della lettura della filastrocca finché la coda non è vuota (l'ultima stringa stampata è quindi il nome del sopravvissuto)

Per il collaudo si utilizzi il file di persone.txt.

Per $k=9$ il programma dovrà visualizzare il seguente andamento:

```
Persone estratte dal cerchio:
Pippo
DareDevil
Pluto
Thor
Paperina
Topolino
Flash
Spiderman
Gambadilegno
Superman
```

All'atto della correzione il codice verrà provato nel seguente modo:

```
$ rm *.class
$ javac *.java
$ java Giuseppe persone.txt
```

Nell'esecuzione della prova si possono usare solo le classi dei pacchetti `java.lang` e le classi `Scanner` e `NoSuchElementException` del pacchetto `java.util`.

Alla fine della prova il candidato lascerà nella directory di lavoro i file modificati che dovrà contenere in testa un commento con **nome e cognome** del candidato, matricola, data, numero della postazione.