

Esame del 15/06/2015
Scritto di Java – Turno 2, A
Tempo per la Soluzione: 2 Ore

Realizzare un programma Java che elabori un file di testo che contiene informazioni relative un insieme di persone e i loro amici:

```
Marco->Matteo, Luca  
Franco, Luca<-Matteo  
Marco->Anna  
Luca, Franco<-Anna
```

Ogni linea è composta da una persona (es. Marco), quindi la stringa “->” e quindi una lista di amici separata da virgola. Oppure da una lista di amici, la stringa “<-”, e quindi la persona di riferimento (la freccia cioè può essere in un verso o nell’altro. Una persona può comparire su più linee

Il programma deve funzionare per ogni file di ingresso che rispetti la sintassi descritta.

Il programma deve calcolare le informazioni richieste nei punti seguenti. Per ogni punto il programma crea un file di testo chiamato rispettivamente 1.txt, 2.txt, 3.txt, 4.txt, 5.txt in cui scrivere la risposta.

I punti da calcolare sono (**ogni punto vale 6 punti all’esame**):

1. Per ogni persona stampare il numero di amici (es. Marco -> 3)
2. Trovare la persona con più amici che iniziano con la lettera “M” (es. Marco che ha 1 amico con la M)
3. Trovare le persone che hanno gli stessi amici (es. Matteo e Anna)
4. Stampare le persone da quella che ha più amici a quella che ha meno amici
5. Stampare la tabella degli amici

	Marco	Matteo	Luca	Franco	Anna
Marco	0	1	1	0	1
Matteo	0	0	1	1	0
Luca	0	0	0	0	0
Franco	0	0	0	0	0
Anna	0	0	1	1	0

Esame del 15/06/2015
Scritto di Java – Turno 2, B
Tempo per la Soluzione: 2 Ore

Realizzare un programma Java che elabori un file di testo che contiene informazioni relative un insieme di persone e i loro amici:

```
Marco->Matteo, Luca  
Franco, Luca<-Matteo  
Marco->Anna  
Luca, Franco<-Anna
```

Ogni linea è composta da una persona (es. Marco), quindi la stringa “->” e quindi una lista di amici separata da virgola. Oppure da una lista di amici, la stringa “<-”, e quindi la persona di riferimento (la freccia cioè può essere in un verso o nell’altro. Una persona può comparire su più linee

Il programma deve funzionare per ogni file di ingresso che rispetti la sintassi descritta.

Il programma deve calcolare le informazioni richieste nei punti seguenti. Per ogni punto il programma crea un file di testo chiamato rispettivamente 1.txt, 2.txt, 3.txt, 4.txt, 5.txt in cui scrivere la risposta.

I punti da calcolare sono (ogni punto vale 6 punti all’esame):

1. Per ogni persona stampare tutti i suoi amici (es. Marco -> Matteo, Luca, Anna)
2. Trovare le persone che non hanno amici che iniziano con la lettera “M” (es. Matteo, Anna, Franco, Luca)
3. Trovare le persone che hanno gli stessi amici (es. Matteo e Anna)
4. Per ogni persona, stampare i suoi amici in ordine alfabetico (*Suggerimento: dato un `ArrayList<String> x`, `Collections.sort(x)` ordina `x` in ordine alfabetico*)
5. Stampare la tabella degli amici

	Marco	Matteo	Luca	Franco	Anna
Marco	0	1	1	0	1
Matteo	0	0	1	1	0
Luca	0	0	0	0	0
Franco	0	0	0	0	0
Anna	0	0	1	1	0