

# Laboratorio di Algoritmi e Strutture Dati

Docente: S. Aguzzoli

Progetto “CSI”

valido per gli appelli di gennaio e febbraio 2012

## 1 Il problema

Un truculento assassinio è avvenuto in un vicolo vicino a Trafalgar Square nell’ora di punta. Donald Algos, l’ispettore capo di Scotland Yard, ha avviato le indagini in modo minuzioso, raccogliendo le dichiarazioni di numerosissimi testimoni e ricavando informazioni utili da alcuni supporti tecnici disponibili (registrazioni delle videocamere che controllano il traffico, ricevute dei parchimetri, ecc). I dati sono troppi per poterli analizzare a mano, ma l’ispettore Algos non si scoraggia perchè ha a disposizione CSI, il *Calcolatore Speciale per Investigazioni*, un sistema informatico che può aiutarlo a ricostruire quanto è accaduto e scoprire chi è il colpevole.

Obiettivo del progetto è sviluppare un sistema che simuli CSI e che sappia svolgere in modo efficiente le operazioni descritte in seguito, su una dimensione arbitraria dei dati in ingresso.

Molte persone risultano presenti a Trafalgar Square il giorno del delitto: sono tutte considerate *testimoni*. Ciascun testimone è identificato da un nome (non ci sono due testimoni con lo stesso nome); indichiamo con  $\mathcal{N}$  l’insieme dei nomi dei testimoni.

Se due testimoni risultano presenti in Trafalgar Square nello stesso momento, diciamo che si sono *incontrati*.

Un *orario* della giornata in cui è avvenuto il delitto è specificato nella forma  $hh : mm$ , e deve appartenere all’intervallo  $[0 : 00, 24 : 00]$  estremi inclusi (se un orario  $hh : mm$  è tale che  $hh < 10$ , si usa una sola cifra per scriverla, mentre per i minuti si usano sempre due cifre, esempio:  $9 : 02$ ).

La scena del delitto è considerata *deserta* all’orario  $x$  se dalle testimonianze nessun testimone risulta presente in questo orario.

Le informazioni raccolte dall’ispettore Algos sono di due tipi.

Una *testimonianza oraria* consiste nella dichiarazione di un testimone, il quale afferma di essere stato in Trafalgar Square il giorno del delitto, specificando a che ora è arrivato e a che ora se n’è andato; tali testimonianze sono rappresentate da triple della forma  $(y, arrivo_y, partenza_y)$ , dove  $y$  è il nome del testimone mentre  $arrivo_y$  e  $partenza_y$  sono i due orari di arrivo e partenza e  $arrivo_y < partenza_y$ . Un *sistema di testimonianze orarie* è un insieme di testimonianze orarie ciascuna relativa ad un testimone diverso. Si osservi che si sta assumendo che ogni testimone è stato sulla scena del delitto una e una sola volta.

**Esempio 1** *Supponiamo di conoscere le quattro testimonianze seguenti:*

1. *Al è arrivato alle 12 : 00 e se ne è andato alle 16 : 45.*
2. *John è arrivato alle 11 : 40 e se ne è andato alle 15 : 45.*
3. *Jack è arrivato alle 17 : 22 e se ne è andato alle 17 : 54.*
4. *Paul è arrivato alle 16 : 22 e se ne è andato alle 17 : 22.*

Allora il sistema di testimonianze orarie associato a queste testimonianze è  $\{(Al, 12 : 00, 16 : 45), (John, 11 : 40, 15 : 45), (Jack, 17 : 22, 17 : 54), (Paul, 16 : 22, 17 : 22)\}$ . Si noti che Jack e Paul si sono incontrati, poiché  $arrivo_{Jack} = partenza_{Paul} = 17 : 22$ .

Una *precedenza oraria* è invece un'informazione che riguarda una coppia di testimoni in  $\mathcal{N}$  che non si sono incontrati: la precedenza oraria  $(y, z)$ , dove  $y, z \in \mathcal{N}$ , indica che il testimone di nome  $y$  se n'è andato prima dell'arrivo del testimone di nome  $z$ .

Un insieme  $\mathcal{P}$  di precedenze orarie può *indurre logicamente* eventuali altre precedenze orarie. In generale, indicheremo con  $\bar{\mathcal{P}}$  l'insieme ottenuto aggiungendo a  $\mathcal{P}$  tutte le precedenze orarie indotte logicamente da  $\mathcal{P}$ .

**Esempio 2** Se  $(Anna, Beatrice) \in \mathcal{P}$  e  $(Beatrice, Carlo) \in \mathcal{P}$ , allora anche Anna deve essere andata via prima di Carlo, quindi  $(Anna, Carlo) \in \bar{\mathcal{P}}$ .

Un insieme  $\mathcal{P}$  di precedenze orarie potrebbe però essere contraddittorio, infatti potrebbero esserci dei testimoni che mentono o errori nella raccolta dei dati. Più precisamente, diremo che  $\mathcal{P}$  è *contraddittorio* quando esistono due testimoni  $n$  e  $m$  tali che sia  $(m, n)$  che  $(n, m)$  sono precedenze orarie in  $\bar{\mathcal{P}}$ .

**Esempio 3** *L'insieme*

$$\mathcal{P} = \{(Carlo, Davide), (Anna, Beatrice), (Beatrice, Enrico), (Davide, Anna), (Beatrice, Carlo)\}$$

*di precedenze orarie è contraddittorio perché se Anna è andata via prima dell'arrivo di Beatrice che a sua volta è andata via prima di Carlo, che a sua volta è andato via prima di Davide, non è possibile che Davide sia andato via prima dell'arrivo di Anna (ricordate che ciascun testimone è stato sulla scena del delitto una sola volta!). In altre parole,  $\bar{\mathcal{P}}$  contiene sia  $(Davide, Anna)$  che  $(Anna, Davide)$  e dunque  $\mathcal{P}$  è contraddittorio.*

Se  $\mathcal{P}$  non è contraddittorio, diremo che  $\mathcal{P}$  è un *sistema di precedenze orarie*; in questo caso:

- se  $(y, z) \in \bar{\mathcal{P}}$ , allora diremo che  $y$  è andato via prima dell'arrivo di  $z$  secondo  $\mathcal{P}$ ;
- se  $(z, y) \in \bar{\mathcal{P}}$ , allora diremo che  $z$  è andato via prima dell'arrivo di  $y$  secondo  $\mathcal{P}$ ;
- se né  $(y, z)$  né  $(z, y)$  sono in  $\bar{\mathcal{P}}$ , allora si intende che *sicuramente*  $y$  e  $z$  si sono incontrati in qualche momento.

Un sistema  $\mathcal{T}$  di testimonianze orarie è una *ricostruzione* su  $\mathcal{N}$  per un sistema  $\mathcal{P}$  di precedenze orarie se, per ogni coppia di testimoni  $y$  e  $z$  in  $\mathcal{N}$ , si verifica una e una soltanto delle tre seguenti possibilità:

- $y$  e  $z$  si sono incontrati secondo  $\mathcal{T}$ ;
- $y$  è andato via prima dell'arrivo di  $z$  secondo  $\mathcal{P}$ ;
- $z$  è andato via prima dell'arrivo di  $y$  secondo  $\mathcal{P}$ .

Naturalmente le informazioni date dalle precedenze orarie sono meno precise di quelle date dalle testimonianze orarie: infatti in genere sono possibili diverse ricostruzioni per un dato insieme di precedenze orarie.

**Esempio 4** Consideriamo  $\mathcal{N} = \{Al, John, Jack\}$  e sia  $\mathcal{T} = \{(Al, 12 : 00, 16 : 45), (John, 11 : 40, 15 : 45), (Jack, 17 : 22, 17 : 54)\}$ . Allora  $\mathcal{T}$  è una ricostruzione per il seguente sistema di precedenze orarie:  $\mathcal{P} = \{(Al, Jack), (John, Jack)\}$ . Si noti che anche il seguente sistema di testimonianze orarie  $\mathcal{T}' = \{(Al, 12 : 00, 12 : 30), (John, 12 : 00, 13 : 00), (Jack, 13 : 01, 14 : 00)\}$  è una ricostruzione per  $\mathcal{P}$  su  $\mathcal{N}$ . Se

invece Jack fosse arrivato alle 13 : 00 il sistema di testimonianze orarie così determinato  $\mathcal{T}'' = \{(Al, 12 : 00, 12 : 30), (John, 12 : 00, 13 : 00), (Jack, 13 : 00, 14 : 00)\}$  non sarebbe una ricostruzione di  $\mathcal{T}$  su  $\mathcal{N}$ , dato che non sarebbe più vero che John se n'è andato prima dell'arrivo di Jack. Il sistema  $\mathcal{T}''$  sarebbe invece una ricostruzione su  $\mathcal{N}$  per  $\mathcal{P}' = \{(Al, Jack)\}$ .

Un sistema  $\mathcal{P}$  di precedenze orarie si dice *realistico* su un insieme di nomi  $\mathcal{N}$  se esiste almeno una ricostruzione  $\mathcal{T}$  di  $\mathcal{P}$  su  $\mathcal{N}$ .

**Esempio 5** Il sistema  $\mathcal{P} = \{(Anna, Carlo), (Beatrice, Elisabetta)\}$  non è realistico su  $\mathcal{N} = \{Anna, Beatrice, Carlo, Elisabetta\}$  infatti, dato che né  $(Beatrice, Carlo)$  né  $(Carlo, Beatrice)$  sono indotti logicamente da  $\mathcal{P}$ , una qualsiasi ricostruzione  $\mathcal{T}$  di  $\mathcal{P}$  su  $\mathcal{N}$  prevederebbe che Carlo abbia incontrato Beatrice in qualche momento. Più in generale, in  $\mathcal{T}$  si avrebbe che sia Carlo che Anna hanno incontrato entrambi sia Elisabetta che Beatrice. Ma questo sarebbe impossibile dato che ciascuno dei 4 è stato presente sulla scena del delitto una volta sola.

Un testimone  $y$  è considerato *testimone chiave* se, dalle informazioni raccolte, ogni altro testimone ha incontrato sulla scena del delitto al massimo lo stesso numero di persone incontrate da  $y$ .

Un insieme di testimoni  $\{y_1, y_2, \dots, y_n\}$  è considerata una potenziale *banda* se, dalle informazioni raccolte, durante ogni istante nell'arco di tempo compreso tra l'arrivo del primo (ad arrivare) degli  $y_i$  e la partenza dell'ultimo (a partire) degli  $y_i$ , almeno uno degli  $y_i$  è stato presente, mentre non è stato mai presente nessun altro testimone al di fuori degli  $y_i$ .

**Esempio 6** Sia  $\mathcal{T} = \{(a, 2 : 00, 5 : 00), (b, 1 : 00, 3 : 00), (c, 2 : 30, 2 : 45), (d, 3 : 30, 5 : 40), (e, 7 : 00, 9 : 00), (f, 7 : 30, 9 : 30), (g, 7 : 15, 9 : 15)\}$  un sistema di testimonianze orarie. Allora  $a$  è l'unico testimone chiave in quanto ha incontrato 3 altri testimoni:  $b, c, d$ . Ogni altro testimone ha incontrato al massimo due altri testimoni. Vi sono due potenziali bande:  $\{a, b, c, d\}$  e  $\{e, f, g\}$ . Consideriamo ora il sistema  $\mathcal{T}' = \mathcal{T} \cup \{(h, 1 : 00, 10 : 00), (i, 1 : 30, 9 : 45)\}$ . In questa nuova situazione vi sono due testimoni chiave:  $h$  che ha incontrato  $a, b, c, d, e, f, g, i$  e  $i$  che ha incontrato  $a, b, c, d, e, f, g, h$ . Vi è inoltre una sola possibile banda:  $\{a, b, c, d, e, f, g, h, i\}$ .

**Esempio 7** Sia  $\mathcal{P} = \{(e, a), (e, b), (e, c), (b, d), (b, c)\}$  un sistema di precedenze orarie sull'insieme di testimoni  $\mathcal{N} = \{a, b, c, d, e\}$ . Allora  $a$  è l'unico testimone chiave in quanto ha incontrato 3 altri testimoni:  $b, c, d$ . Ogni altro testimone ha incontrato al massimo due altri testimoni. Vi sono due potenziali bande:  $\{a, b, c, d\}$  ed  $\{e\}$ . Si consideri di nuovo  $\mathcal{P}$  ma sull'insieme di testimoni  $\mathcal{N}' = \{a, b, c, d, e, f\}$ . Allora  $f$  incontra tutti gli altri testimoni e dunque è l'unico testimone chiave, dato che nessun altro testimone incontra tutti. Vi è una sola potenziale banda che coincide con  $\mathcal{N}'$ .

Il testimone  $y$  è *sospettato* se, dalle informazioni raccolte, risulta essere rimasto solo sulla scena del delitto, in ogni possibile ricostruzione di  $\mathcal{P}$  su  $\mathcal{N}$ .

I testimoni  $y$  e  $z$  sono potenziali *complici* se, secondo le informazioni raccolte, sono stati presenti sulla scena del delitto contemporaneamente ma senza altri testimoni, in ogni possibile ricostruzione di  $\mathcal{P}$  su  $\mathcal{N}$ .

**Esempio 8** Sia  $\mathcal{P} = \{(b, c)\}$  un sistema di precedenze orarie sull'insieme di testimoni  $\mathcal{N} = \{a, b, c\}$ . Allora, dato che  $b$  se n'è andato prima dell'arrivo di  $c$ , e dato che  $a$  ha incontrato sia  $b$  che  $c$ , si conclude che  $a$  deve essere stato presente dopo la partenza di  $b$  e prima dell'arrivo di  $c$ . Dunque  $a$  è presente necessariamente da solo, a una certa ora, in ogni possibile ricostruzione di  $\mathcal{P}$  su  $\mathcal{N}$ , ed è perciò sospettato. Ragionando analogamente si osserva che sia  $(a, b)$  che  $(a, c)$  costituiscono coppie di complici, in quanto entrambi i componenti di ognuna delle due coppie risultano presenti sulla scena contemporaneamente, e senza altri testimoni, in ogni possibile ricostruzione.

Si consideri ora  $\mathcal{P}' = \{(a, b), (c, b)\}$  Allora  $a$  e  $c$  si debbono incontrare in ogni ricostruzione, dato che non è il caso che  $a$  sia partito prima dell'arrivo di  $c$ , e neppure che  $c$  sia partito prima dell'arrivo di

a. Si verifica facilmente che sia  $\mathcal{T}' = \{(a : 3 : 00, 4 : 00), (b : 5 : 00, 6 : 00), (c : 3 : 30, 4 : 00)\}$  che  $\mathcal{T}'' = \{(a : 3 : 00, 4 : 00), (b : 5 : 00, 6 : 00), (c : 3 : 00, 4 : 00)\}$  sono ricostruzioni di  $\mathcal{P}$  su  $\mathcal{N}$ , ma mentre secondo  $\mathcal{T}'$ ,  $a$  è rimasto da solo sulla scena del delitto, secondo  $\mathcal{T}''$   $a$  e  $c$  sono sempre stati presenti insieme sulla scena. Dunque non è vero che  $a$  risulta essere rimasto solo sulla scena del delitto, in ogni possibile ricostruzione, per cui  $a$  non è sospettato.

Si noti che per  $\mathcal{P}'$ ,  $a$  e  $c$  costituiscono una coppia di complici, in quanto presenti sulla scena contemporaneamente, e senza altri testimoni, in ogni possibile ricostruzione.

## 2 Specifiche di implementazione

Il programma deve leggere dallo standard input (`stdin`) una sequenza di righe (separate da `\n`), ciascuna delle quali corrisponde a una riga della prima colonna delle tabelle 1, 2 e 3 (a seconda dell'appello di consegna) e dove  $s$  è una stringa che rappresenta un nome di file,  $y, z$  sono stringhe sull'alfabeto  $\{a, b, \dots, z\}$  delle lettere minuscole, mentre  $x, a, p$  sono orari espressi nella forma  $hh : mm$  per  $hh$  e  $mm$  interi ( $mm$  sempre di due cifre, con eventuale aggiunta dello 0 iniziale,  $hh$  eventualmente di una cifra sola, se l'ora è  $< 9$ ),

I vari elementi sulla riga sono separati da uno o più spazi. Quando una riga è letta, viene eseguita l'operazione associata; le operazioni di stampa sono effettuate sullo standard output (`stdout`), e ogni operazione deve iniziare su una nuova riga.

Le operazioni da implementare variano a seconda dell'appello per il quale avviene la consegna.

Chi intende consegnare il progetto per l'appello di gennaio dovrà implementare le operazioni descritte nelle sezioni 2.1 e 2.3; chi consegnerà il progetto per l'appello di febbraio dovrà implementare le operazioni specificate nelle sezioni 2.2 e 2.3.

Si noti che le operazioni richieste sono liberamente implementabili; in particolare, non vanno necessariamente intese come prototipi di funzioni.

### 2.1 Operazioni da implementare per l'appello di gennaio

RIGA DI INPUT	OPERAZIONE
L $s$	<code>leggi_testimonianze(s)</code>
T $y a p$	<code>testimonianza(y, a, p)</code>
C $y$	<code>cancella(y)</code>
o $x$	<code>orario(x)</code>
i	<code>interrogatori()</code>
t	<code>tempo_deserta()</code>
a $s$	<code>analizza_precedenze(s)</code>

Tabella 1: Comandi da implementare per l'appello di gennaio

#### - `leggi_testimonianze(s)`

Se non esiste nessun file di nome  $s$  stampa il messaggio "FILE  $s$  NON TROVATO".

Altrimenti legge le testimonianze contenute nel file di nome  $s$  e le carica in memoria, cancellando eventuali informazioni memorizzate in precedenza.

#### - `testimonianza(y, a, p)`

Se nel sistema è già presente una testimonianza relativa a  $y$ , la rimuove. Quindi inserisce la nuova testimonianza  $(y, a, p)$ .

- **cancella( $y$ )**  
Cancella la testimonianza relativa al testimone  $y$ .
- **orario( $x$ )**  
Stampa l'elenco delle persone presenti sulla scena del delitto all'orario  $x$ .
- **interrogatori()**  
Stabilisce il minimo numero  $n$  di stanze necessarie a separare i testimoni (in modo che nessun testimone sia nella stessa stanza con altri testimoni incontrati sulla scena del delitto) e suddivide i testimoni nelle  $n$  stanze di conseguenza. Stampa la suddivisione determinata secondo il formato specificato nell'apposita sezione. Se è possibile più di una suddivisione, stampa una qualunque di queste.
- **tempo\_deserta()**  
Calcola per quanto tempo Trafalgar Square risulta deserta nel giorno del delitto. L'output deve essere nella forma  $hh : mm$ .
- [FACOLTATIVO] **analizza\_precedenze( $s$ )**  
Verifica innanzitutto se le precedenze contenute nel file di nome  $s$  sono contraddittorie e in questo caso stampa il messaggio "PRECEDENZE CONTRADDITTORIE". In caso contrario, verifica se il sistema di precedenze è inoltre realistico e, a seconda dell'esito, stampa i messaggi "SISTEMA DI PRECEDENZE REALISTICO" o "SISTEMA DI PRECEDENZE NON REALISTICO". Se non esiste nessun file di nome  $s$  stampa il messaggio "FILE  $s$  NON TROVATO".  
Questa operazione è da considerarsi facoltativa.

I comandi relativi alle precedenti operazioni sono riassunte in Tabella 1.

## 2.2 Operazioni da implementare per l'appello di febbraio

RIGA DI INPUT	OPERAZIONE
L $s$	<b>leggi_precedenze(<math>s</math>)</b>
P $y z$	<b>precedenza(<math>y, z</math>)</b>
d	<b>deserta()</b>
s $y$	<b>sospettato(<math>y</math>)</b>
c $y z$	<b>complici(<math>y, z</math>)</b>
a	<b>analizza_precedenze()</b>

Tabella 2: Comandi da implementare per l'appello di febbraio

- **leggi\_precedenze( $s$ )**  
Se non esiste nessun file di nome  $s$  stampa il messaggio "FILE  $s$  NON TROVATO".  
Altrimenti legge l'elenco di testimoni e di precedenze contenute nel file di nome  $s$ . Se l'elenco non è contraddittorio carica le informazioni in memoria, cancellando eventuali informazioni memorizzate in precedenza. Altrimenti stampa il messaggio "PRECEDENZE CONTRADDITTORIE".
- **precedenza( $y, z$ )**  
Se non ci sono testimoni di nome  $y$ , inserisce il testimone  $y$  nel sistema; se non ci sono testimoni di nome  $z$ , inserisce il testimone  $z$  nel sistema. Se la precedenza  $(y, z)$  è contraddittoria rispetto alle precedenze già inserite, ignora la precedenza e stampa il messaggio "PRECEDENZA CONTRADDITTORIA". Altrimenti la precedenza  $(y, z)$  viene inserita nel sistema.

- **deserta()**

Stabilisce se ci sono stati dei momenti (compresi tra l'arrivo del primo testimone presente sulla scena del delitto e la partenza dell'ultimo testimone presente sulla scena del delitto) in cui Trafalgar Square risulta deserta e, a seconda dei casi, stampa i messaggi "DESERTA" o "NON DESERTA".

- **sospettato(*y*)**

Se non esistono testimonianze relative a *y*, stampa il messaggio "*y* NON ESISTENTE".

Altrimenti stabilisce se *y* è un sospettato e, a seconda dei casi, stampa i messaggi "*y* SOSPETTATO" o "*y* NON SOSPETTATO".

- **complici(*y*, *z*)**

Se non ci sono testimoni di nome *y*, stampa il messaggio "*y* NON ESISTENTE". Se non ci sono testimoni di nome *z*, stampa il messaggio "*z* NON ESISTENTE".

Altrimenti stabilisce se *y* e *z* sono potenziali complici e, a seconda dei casi, stampa i messaggi "*y* COMPLICE DI *z*" o "*y* NON COMPLICE DI *z*".

- [FACOLTATIVO] **analizza\_precedenze()**

Verifica se il sistema di precedenze è realistico e, a seconda dell'esito, stampa i messaggi "SISTEMA DI PRECEDENZE REALISTICO" o "SISTEMA DI PRECEDENZE NON REALISTICO".

Questa operazione è da considerarsi facoltativa.

I comandi relativi alle precedenti operazioni sono riassunte in Tabella 2.

### 2.3 Operazioni per entrambi gli appelli

RIGA DI INPUT	OPERAZIONE
e <i>y</i>	<b>elenco(<i>y</i>)</b>
m	<b>massimo()</b>
l	<b>lunga()</b>
k <i>y</i>	<b>chiave(<i>y</i>)</b>
b <i>y</i>	<b>banda(<i>y</i>)</b>
q	termina l'esecuzione

Tabella 3: Comandi da implementare obbligatoriamente per entrambi gli appelli

- **elenco(*y*)**

Se non esistono testimonianze relative a *y*, stampa il messaggio "*y* NON ESISTENTE".

Altrimenti stampa l'elenco delle persone incontrate dal testimone di nome *y*.

- **massimo()**

Stampa il numero massimo di presenze contemporanee sulla scena del delitto.

- **lunga()**

Stampa la più lunga sequenza di testimoni che si sono susseguiti sulla scena del delitto senza mai incontrarsi. Nel caso in cui ci siano più sequenze di lunghezza massima, stampa una qualunque di queste.

- **testimone\_chiave**( $y$ )

Se non esistono testimonianze relative a  $y$ , stampa il messaggio “ $y$  NON ESISTENTE”.

Altrimenti stabilisce se  $y$  è un testimone chiave e, a seconda dei casi, stampa i messaggi “ $y$ : TESTIMONE CHIAVE” oppure “ $y$ : TESTIMONE NON CHIAVE”.

- **banda**( $y$ )

Se non esistono testimonianze relative a  $y$ , stampa il messaggio “ $y$  NON ESISTENTE”.

Altrimenti stampa l’elenco dei membri della potenziale banda di  $y$ .

I comandi relativi alle precedenti operazioni sono riassunte in Tabella 3.

## Specifiche di formato

1. I nomi dei testimoni sono stringhe di lunghezza arbitraria sull’alfabeto  $\{a, b, \dots, z\}$  delle lettere minuscole.
2. I file  $s$  contenenti le testimonianze, da leggere con `leggi_testimonianze(s)`, sono strutturati come segue:

$$\begin{array}{ccc} nome_1 & arrivo_1 & partenza_1 \\ nome_2 & arrivo_2 & partenza_2 \\ \vdots & \vdots & \vdots \\ nome_k & arrivo_k & partenza_k \end{array}$$

dove  $nome_i$  è il nome dell’ $i$ -esimo testimone,  $arrivo_i$  è il suo orario di arrivo, mentre  $partenza_i$  è il suo orario di partenza. Gli elementi su una riga sono separati da uno spazio.

3. I file  $s$  contenenti le precedenze, da leggere con `leggi_precedenze(s)`, sono strutturati come segue:

$$\begin{array}{cccc} n & nome_1 & nome_2 & \cdots & nome_n \\ & nome_{1,1} & nome_{1,2} & & \\ & nome_{2,1} & nome_{2,2} & & \\ & \cdots & & & \\ & nome_{k,1} & nome_{k,2} & & \end{array}$$

dove  $n$  specifica il numero totale di testimoni,  $nome_h$  è il nome dell’ $h$ -esimo testimone, inoltre, per ogni coppia di indici  $i, j$  esiste un indice  $\ell$  tale che  $nome_{i,j} = nome_\ell$  (vale a dire, le asserzioni di precedenza riguardano i testimoni noti) e la  $i$ -esima riga rappresenta l’asserzione che il testimone  $nome_{i,1}$  se n’è andato prima dell’arrivo del testimone  $nome_{i,2}$ . I nomi su una riga sono separati da uno spazio.  $n$  è separato da  $nome_1$  da uno spazio. Vi è una riga vuota dopo la prima.

4. L’output di `interrogatori()` è strutturato come segue:

$$\begin{array}{l} (k, \\ nome_{1,1} \ nome_{1,2} \ \cdots \ nome_{1,h_1} \cdot \\ nome_{2,1} \ nome_{2,2} \ \cdots \ nome_{2,h_2} \cdot \\ \vdots \\ nome_{k,1} \ nome_{k,2} \ \cdots \ nome_{k,h_k} \cdot \\ ) \end{array}$$

dove  $k$  è il numero di stanze necessario, mentre la  $i$ -esima riga specifica l’elenco dei testimoni da interrogare nella  $i$ -esima stanza. I nomi su una riga sono separati da uno spazio. Si noti il punto alla fine di ogni riga. L’ordine in cui sono visualizzati i testimoni della riga  $i$ -esima è arbitrario.

5. La sequenza di testimoni output di `lunga()` deve essere visualizzata ordinatamente come segue:

$$nome_1 \ nome_2 \ \dots \ nome_k$$

dove  $(nome_1, nome_2), (nome_2, nome_3), \dots, (nome_{k-1}, nome_k)$  sono le precedenze valide.

6. Gli elenchi di testimoni da fornire in output per i comandi `orario(x)`, `elenco(y)`, `banda(y)` possono essere visualizzati in un ordine qualunque, con i nomi tutti sulla stessa riga, separati da uno spazio.

### Note e suggerimenti

1. Non si richiede – anzi si sconsiglia – l’uso di grafica, se non per test personali: in modo particolare, non si usi `conio.h` e neppure `clrscr()`.
2. Per semplicità si suppone che l’input sia sempre conforme alle specifiche delle Tabelle 1, 2 e 3, per cui non è necessario controllare la correttezza dell’input. Per leggere l’input si usino le funzioni standard ANSI C `getchar()` e/o `scanf()`.

## 3 Esempi

Si supponga di disporre dei file `f1.txt`, `f2.txt`, `f3.txt`, `f4.txt`, `f5.txt` il cui contenuto è conforme a quanto di seguito specificato:

- `f1.txt`

```
anna 2:02 5:05
bart 0:58 3:02
carlo 9:45 11:15
davide 13:30 16:00
elena 0:58 2:44
francesca 8:30 10:00
giorgio 11:00 12:28
henry 2:49 4:20
iole 6:45 8:00
jack 6:00 7:00
```

- `f2.txt`

```
arturo 20:00 22:00
billy 20:30 22:10
carola 20:20 21:50
dario 20:45 22:05
enrico 21:00 21:30
```

- `f3.txt`

```
10 antonio bruno cristina dante erica filippo giacomo roberta sergio tommaso

antonio erica
bruno dante
bruno erica
cristina giacomo
```



```
dante giacomo
erica giacomo
erica filippo
filippo roberta
filippo sergio
giacomo sergio
giacomo tommaso
filippo tommaso
giacomo roberta
```

- f4.txt

```
4 adamo eva caino abele
```

```
adamo eva
caino abele
eva caino
abele adamo
```

- f5.txt

```
4 aldo biagio ciro diego
```

```
aldo ciro
biagio diego
```

Di seguito sono mostrati tre esempi di esecuzione che fanno uso dei file precedenti. Le righe che iniziano con il simbolo > sono da intendersi come righe di input.

## Esempio per la consegna di gennaio

```
>L f1.txt
```

```
>o 7:10
```

```
iole
```

```
>o 3:00
```

```
anna bart henry
```

```
>e antonio
```

```
antonio NON ESISTENTE
```

```
>e bart
```

```
anna elena henry
```

```
>m
```

```
3
```

>l

elena henry iole francesca giorgio davide

>t

11:25

>k anna

anna: TESTIMONE CHIAVE

>k bart

bart: TESTIMONE CHIAVE

>k elena

elena: TESTIMONE NON CHIAVE

>b henry

henry anna bart elena

>b carlo

carlo francesca giorgio

>b davide

davide

>T kurt 3:25 7:25

>k bart

bart: TESTIMONE NON CHIAVE

>k anna

anna: TESTIMONE CHIAVE

>b henry

henry anna bart elena iole jack kurt

>o 7:10

iole kurt

>l

elena henry iole francesca giorgio davide

>t

10:30

>i

(3  
anna jack francesca giorgio davide.  
bart kurt carlo.  
elena henry iole.  
)

>C anna

>k henry

henry: TESTIMONE NON CHIAVE

>b henry

henry bart elena iole jack kurt

>e kurt

henry iole jack

>m

3

>l

elena henry iole francesca giorgio davide

>t

10:30

>i

(3  
bart kurt francesca giorgio davide.  
elena henry jack carlo.  
iole.  
)

>T lillo 13:30 15:00

>T marco 15:00 16:00

>o 15:00

davide lillo marco

>L f2.txt  
>b arturo

arturo billy carola dario enrico

>i

(5  
arturo.  
billy.  
carola.  
dario.  
enrico.  
)

>q

### **Esempio per la consegna di febbraio**

>L f3.txt  
>P sergio bruno

PRECEDENZA CONTRADDITTORIA

>e cristina

antonio bruno dante erica filippo

>e bruno

antonio cristina

>e sergio

roberta tommaso

>l

antonio erica filippo tommaso

>m

3

>k dante

dante TESTIMONE NON CHIAVE

>k cristina

cristina TESTIMONE CHIAVE

>b sergio

sergio roberta tommaso

>b giacomo

giacomo antonio bruno cristina dante erica filippo

>d

DESERTA

>s bruno

bruno NON SOSPETTATO

>s tommaso

tommaso NON SOSPETTATO

>s filippo

filippo SOSPETTATO

>c dante erica

dante NON COMPLICE DI erica

>c antonio bruno

antonio NON COMPLICE DI bruno

>c sergio tommaso

sergio NON COMPLICE DI tommaso

>c tommaso roberta

tommaso NON COMPLICE DI roberta

>c sergio roberta

sergio NON COMPLICE DI roberta

>P sergio roberta

>l

antonio erica filippo sergio roberta

>e sergio

tommaso

```
>s tommaso
tommaso SOSPETTATO
>c sergio tommaso
sergio COMPLICE DI tommaso
>c tommaso roberta
tommaso COMPLICE DI roberta
>c sergio roberta
sergio NON COMPLICE DI roberta
>q
```

### **Esempio per la parte facoltativa**

Input per la consegna di gennaio:

```
>a f3.txt
SISTEMA DI PRECEDENZE REALISTICO
>a f4.txt
PRECEDENZE CONTRADDITTORIE
>a f5.txt
SISTEMA DI PRECEDENZE NON REALISTICO
>q
```

Input per la consegna di febbraio:

```
>L f3.txt
>a
SISTEMA DI PRECEDENZE REALISTICO
>L f4.txt
PRECEDENZE CONTRADDITTORIE
>L f5.txt
>a
SISTEMA DI PRECEDENZE NON REALISTICO
>q
```

## 4 Modalità di consegna

Il presente progetto è valido per gli appelli del 24 gennaio e del 13 febbraio 2012. Il progetto va inviato per posta elettronica all'indirizzo `aguzzoli@dsi.unimi.it` entro la data dell'appello scelto. Occorre presentare:

1. il codice sorgente (rigorosamente ANSI C, compilabile con `gcc`);
2. una sintetica relazione (formato pdf o rtf) che illustra le strutture dati utilizzate e le scelte implementative, analizzando il costo delle diverse operazioni richieste dalla specifica;

Tutti i file (file sorgenti + relazione) devono essere contenuti in un unico file `.zip` il cui nome dovrà essere della forma `cognome_matricola.zip`. La relazione e il codice devono riportare nome, cognome e matricola. Una copia cartacea della relazione e del codice deve inoltre essere consegnata al docente entro le scadenze fissate (lasciandola eventualmente nella sua casella postale presso il dipartimento in via Comelico).

La discussione dei progetti si svolgerà in data e luogo da specificarsi. Il calendario dei colloqui sarà disponibile sulla pagina del corso <http://homes.dsi.unimi.it/~aguzzoli/algo.htm> qualche giorno dopo il termine di consegna del progetto.

La realizzazione del progetto è una prova d'esame da svolgersi **individualmente**. I progetti giudicati frutto di **copiatura** saranno **estromessi** d'ufficio dalla valutazione.

Si richiede allo studente di effettuare un **adeguato collaudo** del proprio progetto su numerosi esempi diversi per verificarne la correttezza.

La versione aggiornata del progetto è pubblicata in `.pdf` sul sito:

<http://homes.dsi.unimi.it/~aguzzoli/algo.htm>.

Si consiglia di consultare periodicamente questo sito per eventuali correzioni e/o precisazioni relative al testo del progetto.