```
%%
%
% Il sistema di messaggistica di Facebook permette di ricevere messaggi da
% qualsiasi mittente. Un messaggio e' caratterizzato da un mittente e da un
% testo.
%
% Vogliamo implementare un sistema di filtraggio per rilevare automaticamente
% messaggi indesiderati, basandoci sulle seguenti ipotesi semplificative:
%
%    * se il messaggio e' ricevuto da una persona conosciuta, ossia da una
%    persona nella lista degli amici, allora il messaggio non e' da scartare
%
%    * se il messaggio e' ricevuto da una persona sconosciuta, ossia non
%    presente nella lista degli amici, allora e' necessario esaminare la storia
%    dei messaggi ricevuti in passato, per determinare un "valore atteso" che
%    ci permetta di decidere se il messaggio appena ricevuto e' "nella media".
%
% Quindi servira' una funzione:
%
%    [buono motivo] = filtra_messaggio(messaggio, messaggi, amici);
%
% dove:
%    buono
%        e' di tipo logical che vale vero solo se il messaggio e' buono
%
%    motivo
%        e' di tipo char, e vale 'a' ad indicare che il messaggio e' buono (o
%        cattivo) perche' (non) inviato da un amico, o vale 'm' ad indicare che
%        il messaggio e' buono (o cattivo) perche' (non) "nella media" dei
%        messaggi passati.
%
%    messaggio
%        e' una struttura dati a due campi (mittente e testo), entrambi
%        stringhe il cui significato e' autoesplicativo. Ad esempio:
%
%           * messaggio(1).testo = 'Ciao come stai?'
%           * messaggio(1).mittente.nome = 'Federico'
%           * messaggio(1).mittente.cognome = 'Maggi'
%
%    messaggi
%        e' un vettore contenente i messaggi ricevuti in passato. Utilizzeremo
%        la medesima struttura dati di cui sopra.
%
%   amici
%        e' un vettore contenente gli amici. Utilizzeremo una struttura
%        dati contenente i campi nome e cognome.
%
% Il filtraggio basato sul mittente e' semplice: e' sufficiente verificare che
% il mittente sia presente nel vettore degli amici.
%
% Invece, il filtraggio basato sul contenuto e' piu' complesso, pertanto e'
% necessario implementarlo con cura. Utilizzeremo una funzione:
%
%    * buono = controlla_contenuto(messaggio, messaggi)
%
% Tale funzione fara' uso della seguente ulteriore funzione, che calcola il
% "valore atteso" di un vettore di messaggi.
%
%    * Fm = valore_atteso(messaggi)
%
% si assume che messaggi contenga N elementi (vettore di N messaggi).
%
% Fm e' un vettore riga di 6 colonne, che sono, rispettivamente, il valor medio
% (mean(vettore)) e la deviazione standard (sqrt(var(vettore))) dei seguenti
% tre valori calcolati su tutti i messaggi con la seguente funzione
%
%    * [l v c] = estrai_caratteristiche(testo)
%
% dove:
%
%        l e' la lunghezza del messaggio, esclusi gli spazi
%        v e' il numero di vocali
%        c e' il numero di consonanti
%
```

```
% Quindi, per ogni messaggio si calcolano l, v, e c, e di questi si calcolano
% la media e la deviazione standard. Il tutto va memorizzato, in un vettore
% di 6 elementi: [Ml Vl Mv Vv Mc Vc]
%
% Si chiede di implementare:
%
%    * estrai_caratteristiche(testo)
%    * valore_atteso(messaggi)
%
% Facoltativamente, implementare la funzione:
%
%   controlla_contenuto(messaggio, messaggi)
%
% la quale ritornera' un valore logical vero solo se il messaggio ha le
% caratteristiche [l v c] che soddisfano tutte le tre seguenti condizioni:
%
%    *  medio(l) - sqrt(2) * sqrt(var(l)) <= l <= medio(l) + sqrt(2) * sqrt(var(l))
%    *  medio(v) - sqrt(2) * sqrt(var(v)) <= v <= medio(v) + sqrt(2) * sqrt(var(v))
%    *  medio(c) - sqrt(2) * sqrt(var(c)) <= c <= medio(c) + sqrt(2) * sqrt(var(c))
%
% Le strutture dati 'amici' e 'messaggi' sono già disponibili in un file
% "facebook.mat", vanno caricate all'inizio dello script.
%

% dichiarazione e inizializzazione della lista degli amici
%
% amici = [];
%
% amici(1).nome = 'cras';
% amici(1).cognome = 'rhoncus';
%
% amici(2).nome = 'aliquam';
% amici(2).cognome = 'erat';
%
% amici(3).nome = 'turpis';
% amici(3).cognome = 'velit';
%
% amici(4).nome = 'ornare';
% amici(4).cognome = 'aliquam';
%
% amici(5).nome = 'nibh';
% amici(5).cognome = 'quisque';
%

% dichiarazione e inizializzazione della lista dei messaggi
%
% messaggi = [];
%
% messaggi(1).testo = 'Praesent vitae ligula nec orci pretium vestibulum';
% messaggi(1).mittente = amici(1);
%
% messaggi(2).testo = 'Curabitur quis dui sit amet elit luctus aliquam';
% messaggi(2).mittente = amici(2);
%
% messaggi(3).testo = 'Vivamus convallis urna id felis';
% messaggi(3).mittente = amici(2);
%
% messaggi(4).testo = 'Cras aliquam massa ullamcorper sapien';
% messaggi(4).mittente = amici(2);
%
% messaggi(5).testo = 'Sed accumsan quam ac tellus';
% messaggi(5).mittente = amici(3);
%
% messaggi(6).testo = 'Nulla porta tempus sapien';
% messaggi(6).mittente = amici(2);
%
% messaggi(7).testo = 'Cras aliquam massa ullamcorper sapien';
% messaggi(7).mittente = amici(2);
%
% messaggi(8).testo = 'Vivamus convallis urna id felis';
% messaggi(8).mittente = amici(2);
%
% messaggi(9).testo = 'Cras aliquam massa ullamcorper sapien';
% messaggi(9).mittente = amici(4);
```

```
%
% messaggi(10).testo = 'Aliquam adipiscing libero vitae leo';
% messaggi(10).mittente = amici(2);

% salvataggio variabili sufile MATLAB (.mat), creato in precedenza come segue,
% dopo aver dichiarato le variabili 'amici' e 'messaggi' come sopra
%
% save('facebook.mat', 'amici', 'messaggi');

% lettura dati da file MATLAB: leggo solo le variabili 'amici' e 'messaggi' (in
% questo caso sono le uniche, ma in altri casi potrebbero essercene delle
% altre)
%
% le varibili sono caricate direttamente nel workspace locale e non c'e' bisogno
% di fare altro
load('facebook.mat', 'amici', 'messaggi');

% Controllo del contenuto di un messaggio, dati i messaggi precedenti
%
testo = 'Ciao come stai'
buono = controlla_contenuto(testo, messaggi)

% Controllo del contenuto di un messaggio, dati i messaggi precedenti
%
messaggi(10).testo
buono = controlla_contenuto(messaggi(10).testo, messaggi)
```