9 MATLAB

9.1 Esercizi

Esercizio 9.1

Scrivere uno script che calcoli la sequenza di Fibonacci di lunghezza 20, e la stampi a schermo. Successivamente si richieda di inserire un numero $2 \leq n \leq 4180$ e valuti se il numero è di Fibonacci. Altrimenti restituisce il numero di Fibonacci più vicino. La successione di Fibonacci è definita così:

$$F(0) = 0$$
$$F(1) = 1$$
$$F(n) = F(n-1) + F(n-2), n > 1$$

Esercizio 9.2

Utilizzando il fatto che il quadrato di n è uguale alla somma dei primi n numeri dispari, calcolare il quadrato di un numero ($n < 100$) inserito dall'utente.

Esercizio 9.3

Chiedere all'utente due parole e stampare a video se una è anagramma dell'altra.

Esercizio 9.4

Creare una matrice `M` di dimensioni 7×5 contenente 0, 1, 2, matrice che rappresenta una situazione in una partita di forza 4 in corso.

Chiedere ai due giocatori, finché uno di questi non inserisce la lettera 'q' (quit), di inserire la colonna (tra 1 e 7) dove intende inserire la propria pedina.

Inserire la pedina nella colonna corretta e visualizzare la matrice `M` così ottenuta.

Bonus: scrivere una porzione dello script che controlli se un giocatore ha vinto, ovvero se ci sono 4 pedine adiacenti dello stesso giocatore in orizzontale, in verticale o **in diagonale**.

Esercizio 9.5

Verificare se una matrice quadrata di dimensione arbitraria è un quadrato magico. Una matrice è un quadrato magico se la somma degli elementi sulle righe, sulle colonne e sulla diagonale principale è la stessa.

Esercizio 9.6

Data una matrice 20×20 che rappresenta le partite di un campionato di calcio (con 0 per vittoria in casa, 1 per pareggio 2 per vittoria in trasferta come risultati possibili). Calcolare la classifica finale ordinata.

Esercizio 9.7

Data una matrice quadrata, leggerla a spirale e metterne il contenuto in un vettore. La lettura a spirale avviene andando a leggere la prima riga, poi l'ultima colonna, quindi l'ultima riga ed infine la prima colonna.

Esercizio 9.8

(TdE 2010 - modificato) Dopo una gara automobilistica si ha come risultato tre tabelle le cui colonne rappresentano gli n partecipanti (numerati da 1 a n) e le righe gli m giri di pista effettuati. Il valore di ogni generica cella (i,j) delle tabelle rappresenta il tempo impiegato (in minuti, secondi e millesimi) dal partecipante j per percorrere il giro i.

Si scrivano le istruzioni per:

- calcolare il tempo totale medio che è stato impiegato dai partecipanti per completare la gara;
- determinare il vincitore della gara (cioè il numero del partecipante il cui tempo di percorrenza totale è minore di quello degli altri partecipanti);

Supponiamo di avere un vettore che ci dica per ogni pilota quanti giri ha effettivamente percorso. Come cambia lo script?

Esercizio 9.9

Scrivere uno script che, ricevendo una matrice M di numeri interi, stampa a video una matrice MR, ottenuta da M nel seguente modo:

- calcola la media aritmetica dei valori di M e ne arrotonda il valore all'intero piú vicino;
- per i valori che in M sono minori della media, in MR si scriva nella posizione corrispondente il valore -1;

- per quelli superiori alla media si pone il valore 1;
- per gli altri (quelli uguali alla media) si pone lo stesso valore in M.

Esercizio 9.10

Si sviluppi uno script che riceve una matrice 8x8, che rappresenta la scacchiera su cui sono disposte le 8 regine, e restituisce se la configurazione delle 8 regine è corretta (nessuna regina mette in scacco un'altra regina), o meno. Si supponga che la matrice contenga il valore 0 in tutte le posizioni libere e il valore 1 nelle posizioni occupate dalle regine.

Soluzioni

Soluzione dell'esercizio 9.1

```
clear
clc
close all

% Inizializza sequenza
fibo = zeros(1,20);

% Calcolo i primi 20 numeri di fibonacci
fibo(1) = 0;
fibo(2) = 1;
for i = 3:20
    fibo(i) = fibo(i-1) + fibo(i-2);
end

fibo

a = input('Inserire un numero (tra 2 e 4180): ');

if sum(a == fibo) > 0
    disp([num2str(a) 'e'' un numero di Fibonacci']);
else
    % Cerco i numeri di fibonacci più piccoli di a e scelgo l'
       ultimo
    inferiori = fibo(fibo < a);
    inferiori = inferiori(end);

    % Cerco i numeri di Fibonacci più grandi di a e scelgo il
       primo
    superiori = fibo(fibo > a);
    superiori = superiori(1);

    % Cerco il più vicino tra il più grande numero di fibonacci
        più piccolo
    % di a e il più piccolo numero di Fibonacci più grandi di a
    if superiori - a < a - inferiori
        vicino = superiori;
    else
        vicino = inferiori;
    end
```

```
37
38      disp(['Il numero di Fibonacci piu'' vicino a ' num2str(a) '
            e'' ' num2str(vicino)]);
39
40  end
```

Soluzione dell'esercizio 9.2

```
1   clc
2   clear
3
4   N = input('Inserire il numero da elevare al quadrato (n < 100):
        ');
5
6   numeri = 2 * [0 : N - 1] + 1;
7
8   % Soluzione alla C 1
9   c = 1;
10  somma_while = 0;
11  while c <= N
12      somma_while = somma_while + numeri(c);
13      c = c + 1;
14  end
15
16  % Soluzione ibrida
17  somma_ibrida = 0;
18  for c = numeri
19      somma_ibrida = somma_ibrida + c;
20  end
21
22  % Soluzione alla Matlab
23  somma_matlab = sum(numeri);
24
25  fprintf('%d^2 = %d\n', N, somma_while);
26  fprintf('%d^2 = %d\n', N, somma_ibrida);
27  fprintf('%d^2 = %d\n', N, somma_matlab);
```

Soluzione dell'esercizio 9.3

```
1   clear
2   clc
3
4   parola1 = input('Inserire la prima parola: ','s');
```

```
parola2 = input('Inserire la seconda parola: ','s');

isto1 = zeros(1,255);
isto2 = zeros(1,255);

for ii = parola1
    isto1(ii) = isto1(ii) + 1;
end

for ii = parola2
    isto2(ii) = isto2(ii) + 1;
end

fprintf('Le due parole ');
if any(isto1 ~= isto2)
    fprintf('non ');
end
fprintf('sono una l''anagramma dell''altra\n');
```

Soluzione dell'esercizio 9.4

```
clear
clc
close all;

M = [ 0 0 0 0 0 0 0 ;...
      0 0 0 0 0 0 0 ;...
      0 0 0 0 0 0 0 ;...
      0 0 0 0 0 0 0 ;...
      0 0 0 0 0 0 0 ];

turno_giocatore = 1;
a = 6;
while (a ~= 'q')
    disp(['E'' il turno del giocatore ' num2str(turno_giocatore
       )]);
    a = input('Inserire una giocata (numero di colonna 1-7) o
       uscire (q): ');

    if a~='q'
        if (M(1,a) ~= 0)
            disp('Giocata illegale');
        else
```

```
            indici = M(:,a) == 0;
            pos_libera = sum(indici);
            M(pos_libera,a) = turno_giocatore;
            imagesc(M);
            if turno_giocatore == 1
                turno_giocatore = 2;
            else
                turno_giocatore = 1;
            end
        end
    end
end
```

Soluzione dell'esercizio 9.5

```
clear
clc
close all

M = magic(4);
%M = randi(4,3);
[r, c] = size(M);

% Controllo matrice quadrata
assert(r == c);

% Calcolo somme su righe
somme = zeros(1,2 * r + 1);
for ii = 1:r
    somme(ii) = sum(M(ii,:));
end

% Calcolo somme su colonne
for ii = (r+1):2*r
    somme(ii) = sum(M(:,ii-r));
end

% Calcolo somma su diagonale
somme(2*r+1) = sum(diag(M));

somme

if sum(somme == somme(1)) == 2*r+1
```

```
29      disp('La matrice e'' un quadrato magico');
30  else
31      disp('La matrice non e'' un quadrato magico');
32  end
```

Soluzione dell'esercizio 9.6

```
1   clear
2   clc
3
4   squadre = { 'Atalanta' 'Bologna' 'Carpi' 'Chievo' 'Empoli' '
        Fiorentina' 'Frosinone' ...
5       'Genoa' 'Inter' 'Juventus' 'Lazio'  'Milan' 'Napoli' '
            Palermo' 'Roma'...
6       'Sampdoria' 'Sassuolo' 'Torino' 'Udinese' 'Verona'};
7   squadre_alt = squadre;
8
9   risultati = randi(3,20)-1;
10  for ii = 1:20
11      risultati(ii,ii) = -1;
12  end
13
14  % Versione alla C
15  punti = zeros(20,1);
16  for ii = 1:20
17      punti(ii) = sum(risultati(ii,:) == 0) * 3 + sum(risultati(
            ii,:) == 1) + ...
18          sum(risultati(:,ii) == 2) * 3 + sum(risultati(:,ii) ==
                1);
19  end
20
21  %Versione alla Matlab
22  punti_alt = sum(risultati == 0,2) * 3 + sum(risultati == 1,2) +
        ...
23          sum(risultati' == 2,2) * 3 + sum(risultati' == 1,2);
24
25  assert(sum(punti == punti_alt) == 20)
26
27  %Ordiniamo le squadre
28  while (~isempty(punti))
29      maxi = max(punti);
30      trovato = 0;
31      contatore = 1;
```

```
32      while trovato == 0
33          if punti(contatore) == maxi
34              disp(['La squadra ' squadre{contatore} ' ha
                    totalizzato ' num2str(punti(contatore)) ' punti.
                    ']);
35              punti(contatore) = [];
36              squadre(contatore) = [];
37              trovato = 1;
38          else
39              contatore = contatore + 1;
40          end
41      end
42  end
43  disp('-----------------------------------');
44
45  %Oppure chiediamo a MATLAB
46  [punti_alt, indici] = sort(punti_alt,'descend');
47  squadre_alt = squadre_alt(indici);
48  for ii = 1:20
49      disp(['La squadra ' squadre_alt{ii} ' ha totalizzato '
           num2str(punti_alt(ii)) ' punti.']);
50  end
```

Soluzione dell'esercizio 9.7

```
1   clear
2   clc
3
4   M = randi(10,7);
5   M_old = M;
6
7   vec = [];
8
9   while(~isempty(M))
10
11      vec = [vec M(1,:)];
12      M(1,:) = [];
13      if (~isempty(M))
14          vec = [vec M(:,end)'];
15          M(:,end) = [];
16      end
17      if (~isempty(M))
18          vec = [vec M(end,end:-1:1)];
```

```
19          M(end,:) = [];
20      end
21      if (~isempty(M))
22          vec = [vec M(end:-1:1,1)'];
23          M(:,1) = [];
24      end
25  end
26
27  assert(sum(vec) == sum(M_old(:)));
28
29  M_old
30  vec
```

Soluzione dell'esercizio 9.8

```
1   clear
2   clc
3
4   n_piloti = 10;
5   n_giri = 30;
6
7   minuti = randi(2,n_piloti,n_giri);
8   secondi = 60 * rand(n_piloti,n_giri);
9   millesimi = 1000 * rand(n_piloti,n_giri);
10
11  tempo_medio = mean(minuti * 60 + secondi + millesimi / 1000, 2)
        ;
12  tempo_vinc = min(tempo_medio);
13
14  vinc = find(tempo_medio == tempo_vinc);
15
16  disp(['Il vincitore e '' ' num2str(vinc)]);
```

Soluzione dell'esercizio 9.9

```
1   clear
2   clc
3
4   M = randi(20,5);
5
6
7   MR = zeros(size(M));
8
```

```
media = round(mean(M(:)));

MR(M < media) = -1;
MR(M > media) = 1;
MR(M == media) = media;
```

Soluzione dell'esercizio 9.10

```
clear
clc
close all

scacchiera = randi(2,8) - 1;

%Controllo righe
righe_ok = all(sum(scacchiera,2) <= 1);

%Controllo colonne
colonne_ok = all(sum(scacchiera) <= 1);

diag_ok = 1;
anti_diag_ok = 1;

%Controllo diagonali principali (alla C)
if righe_ok && colonne_ok && diag_ok
    for ii = 1:7
        somma = 0;
        count_col = 1;
        count_row = ii;
        while (count_row <= 8)
            somma = somma + scacchiera(count_row,count_col);
            count_col = count_col + 1;
            count_row = count_row + 1;
        end
        if somma > 1
            diag_ok = 0;
        end
    end
end

if righe_ok && colonne_ok && diag_ok
    for ii = 2:7
        somma = 0;
```

```
36          count_col = ii;
37          count_row = 1;
38          while (count_col <= 8)
39              somma = somma + scacchiera(count_row,count_col);
40              count_col = count_col + 1;
41              count_row = count_row + 1;
42          end
43          if somma > 1
44              diag_ok = 0;
45          end
46      end
47  end
48
49  %Controllo diagonali principali (alla Matlab)
50  if righe_ok && colonne_ok
51      sum_diag = zeros(13,1);
52      for ii = -6:6
53          sum_diag(ii+7) = sum(diag(scacchiera,ii));
54      end
55      diag_ok = all(sum_diag <= 1);
56  end
57
58  %Controllo antidiagonali
59  if righe_ok && colonne_ok && diag_ok
60      antiscacchiera = flip(scacchiera);
61      sum_anti_diag = zeros(13,1);
62      for ii = -6:6
63          sum_anti_diag(ii+7) = sum(diag(antiscacchiera,ii));
64      end
65      anti_diag_ok = all(sum_anti_diag <= 1);
66  end
67
68  if righe_ok && colonne_ok && diag_ok && anti_diag_ok
69      disp('Le regine sono ben disposte');
70  else
71      disp('Almeno una coppia di regine e'' mal disposta');
72  end
```