

```

%
% Politecnico di Milano
% Corso Informatica B - Prof. Loiacono
% Esercizi di: Alessandro A. Nacci
% Per contatti: alessandro.nacci@mail.polimi.it
% -----
% Esercitazione 7
% ESERCIZIO 1
% -----

% Dato un vettore in ingresso ed un numero a (entrambi float)
% controllare quante occorrenze del numero sono presenti
% nel vettore.

% In seguito elencare i numeri diversi da a presenti nel vettore

vett = input('Inserisci un vettore ')
a = input('inserisci un numero con cui effettuare il confronto: ')

%%% Ricerca dei numeri uguali
%%% SONO NUMERI IN VIRGOLA MOBILE! Usare l'operatore '==' può
%%% portare ad errori dovuti alle approssimazioni!
fprintf("\n\nIl numero di occorrenze del numero %f sono ", a)
disp(sum(abs(vett - a) <= eps))

fprintf("I numeri diversi da %f sono:\n",a)
disp(vett(vett ~= a))

%%% Ricerca dei numeri maggiori
fprintf("\nI numeri maggiori di %f sono ", a)
disp(sum(vett > a))

fprintf("e in particolare sono:\n")
disp(vett(vett > a))

```

```

%
% Politecnico di Milano
% Corso Informatica B - Prof. Loiacono
% Esercizi di: Alessandro A. Nacci
% Per contatti: alessandro.nacci@mail.polimi.it
% -----
% Esercitazione 7
% ESERCIZIO 2
% -----

% Se l'array inserito è numerico, effettuare i seguenti controlli
% Verificare se tutti i numeri sono positivi
% Verificare se esiste un numero negativo
% Applicare la radice quadrata a tutti i valori: ci sono dei valori complessi?

% Verificare se tutti i numeri sono pari
% Trovare gli indici dei numeri pari
% Verificare se esiste un numero dispari
%   Contare i numeri dispari se esistono
%   Dire in che posizioni sono

vett = input('Inserisci un vettore ')

if isnumeric(vett)

    %numeri positivi
    if all(vett > 0),
        disp('tutti i numeri sono positivi')
    else
        disp('non tutti i numeri sono positivi')
    end

    %esistenza numero negativo
    if any(vett < 0),
        disp('esiste un numero negativo')
    else
        disp('non esiste alcun numero negativo')
    end

    %radice quadrata
    sqrt_vett = sqrt(vett)
    if isreal(sqrt_vett),
        disp('non esistono valori complessi')
    else
        disp('esistono valori complessi')
    end

    %numeri pari
    carry_vett = mod(vett,2)
    if(all(carry_vett==0))
        disp('tutti i numeri sono pari')
    else
        disp('non tutti i numeri sono pari')
    end

    %indici dei numeri pari
    pos_pari = find(1-carry_vett)

    %numeri dispari
    if any(carry_vett),
        disp('esiste almeno un numero dispari')
        pos_dispari = find(carry_vett),

```

```
        disp('i numeri dispari sono in posizione')
        disp(pos_dispari)
    else
        disp('non esistono numeri dispari')
    end

else

    disp('Devi inserire un array numerico')
end
```

```

%
% Politecnico di Milano
% Corso Informatica B - Prof. Loiacono
% Esercizi di: Alessandro A. Nacci
% Per contatti: alessandro.nacci@mail.polimi.it
% Tratto da: http://www.unife.it/ing/meccanica/insegnamenti/
%             informatica-industriale/lucidi-delle-lezioni
% -----
% Esercitazione 7
% ESERCIZIO 3 -
% -----

% Data la seguente tabella
%
% op = operaio
% |=====|
% |          op5          |          op1          |          op2          |          op3          |          op4          |
% |-----|
% | Paga oraria          |          5          |          5.50 6.50 6          |          6.25          |
% | Ore settimanali     |          40          |          43          |          37          |          50          |
% |          45          |
% | Pezzi prodotti      |          1000 1100 1000 1200 1100          |
% |-----|
%
% Quanto guadagna ogni operaio?
% Qual è il salario totale di tutti gli operai?
% Quanti pezzi vengono prodotti?
% Qual è il costo medio di un pezzo?
% Quante ore occorrono in media per un pezzo?
% Qual è l'operaio più efficiente?

% Inizializziamo i vettori
pagaoraria=[5 5.5 6.5 5 6.25]
oresett=[40 43 37 50 45]
pezzi=[1000 1100 1000 1200 1100]

% quanto guadagna ogni operaio?
% per rispondere a questa domanda, è sufficiente effettuare una
% moltiplicazione elemento per elemento di vettori paga oraria e oresett
paghe=pagaoraria.*oresett

% Qual'è il salario totale di tutti gli operai?
% Si chiede in pratica di sommare gli elementi della riga 'paghe', che per
% noi è un vettore. è quindi possibile usare direttamente la funzione 'sum',
% che effettua la somma di tutti gli elementi nel vettore in argomento
salariototale=sum(paghe)

% Quanti pezzi vengono prodotti?
% In analogia a quanto detto, si riutilizza la funzione 'sum'
pezzitotale=sum(pezzi)

% Qual'è il costo medio di un prezzo?
% Il costo medio si ricava considerando il numero di pezzi prodotti in totale
% ed il costo totale di produzione (costo operai). Al pezzo, quindi si ha:
% costo pezzo= salariototale/pezzitotale, valori già calcolati con le
% domande precedenti
costopezzo=salariototale/pezzitotale

% Quante ore occorrono in media per pezzo?

```

```
% Considerazioni analoghe al precedente, con la differenza che occorre
% conoscere le ore totali, fin'ora non calcolate, valore ottenibile
% facilmente con la funzione 'sum'
oreperpezzo=sum(oresett)/pezzitotale

% Qual'è l'operaio più efficiente?
% Per rispondere a questa domanda si crea un vettore efficienza, dove ogni
% elemento valuta l'efficienza del singolo operaio. L'efficienza è definita
% come il rapporto tra pezzi prodotti dall'operaio e dalle ore totali
% impiegate per produrre quei pezzi; occorre pertanto un'operazione di
% divisione elemento per elemento tra i vettori 'pezzi' e 'oresett'
efficienza=pezzi./oresett

% Ora che sono state calcolate tutte le efficienze dei singoli operaio,
% occorre individuare l'operaio migliore, ovvero colui che ha efficienza più
% elevata. L'operazione si riduce a ricavare il valore massimo contenuto in
% un vettore (efficienza) e l'indice corrispondente, che individua
% l'operaio. A tal scopo si utilizza la funzione 'max'. help max per
% ulteriori informazioni

% L'operazione max funziona in questo modo:
% [valore_massimo, posizione_valore_massimo] = max(vettore_in_input)
[efficienzamax,chi]=max(efficienza)
```

```

%
% Politecnico di Milano
% Corso Informatica B - Prof. Loiacono
% Esercizi di: Alessandro A. Nacci
% Per contatti: alessandro.nacci@mail.polimi.it
% Tratto da: http://www.unife.it/ing/meccanica/insegnamenti/
%             informatica-industriale/lucidi-delle-lezioni
% -----
% Esercitazione 7
% ESERCIZIO 4
% -----

% DISEGNARE LA TAVOLA PITAGORICA di dimensione nxn
%
%      1   2   3   4   5
%      2   4   6   8  10
%      3   6   9  12  15
%      4   8  12  16  20
%      5  10  15  20  25
%
%
% Per risolvere quest'esercizio, è sufficiente osservare la struttura della
% tavola pitagorica: la prima riga è un vettore di numeri naturali, da 1 a
% 10, così come la prima colonna; mentre l'elemento (i,j)-esimo è ottenuto
% moltiplicando l'elemento j-esimo della prima riga e i-esimo della prima
% colonna.

% Ecco che è possibile vedere la colonna come la trasposta della prima riga,
% e gli altri valori sono ottenibili effettuando una moltiplicazione tra
% vettori

n = input('Inserire la dimensione della tavola pitagorica')

%Determino la riga della tavola pitagorica
a=1:n

%Calcolo (e visualizzazione) della tavola periodica
a'*a

```

```
%  
% Politecnico di Milano  
% Corso Informatica B - Prof. Loiacono  
% Esercizi di: Alessandro A. Nacci  
% Per contatti: alessandro.nacci@mail.polimi.it  
% -----  
% Esercitazione 7  
% ESERCIZIO 5  
% -----  
  
% Verifica anno bisestile  
  
anno = 2011  
  
% è possibile risolvere agevolmente il problema con la struttura  
% if/elseif/else, dividendo in 4 casi possibili  
  
if mod(anno,400)==0 %se è divisibile per 400, è bisestile  
    disp('L'anno e'' bisestile');  
    out=1;  
elseif mod(anno,100)==0 %se è inizio secolo, non è bisestile  
    disp('L'anno non e'' bisestile');  
    out=0;  
elseif mod(anno,4)==0 %se è divisibile per 4, è bisestile  
    disp('L'anno e'' bisestile');  
    out=1;  
else %in tutti gli altri casi, non è bisestile  
    disp('L'anno non e'' bisestile');  
    out=0;  
end
```

```

%
% Politecnico di Milano
% Corso Informatica B - Prof. Loiacono
% Esercizi di: Alessandro A. Nacci
% Per contatti: alessandro.nacci@mail.polimi.it
% Tratto da: http://www.unife.it/ing/meccanica/insegnamenti/
%             informatica-industriale/lucidi-delle-lezioni
% -----
% Esercitazione 7
% ESERCIZIO 6
% -----

% Progettare uno script matlab che chieda all'utente quale tra le
% seguenti operazioni si vuole effettuare su un generico vettore
%   MEDIA MASSIMO MINIMO

vettore = []

%Creazione menù: per prima cosa occorre che l'utente sia a conoscenza
%dell'esistenza di un menù e delle opzioni di scelta; per far ciò è
%sufficiente visualizzare delle stringe sul command window che indichino le
%opzioni disponibili
disp('selezionare un azione');
disp('1: calcolo della media');
disp('2: calcolo del massimo');
disp('3: calcolo del minimo');
disp('4: esci');
scelta=input('scelta: ');

%Le righe di codice precedenti hanno la SOLA funzione di visualizzare il
%menù. Per far si che sia realmente un menu, occorre valutare con un test
%il contenuto della scelta effettuata (variabile scelta)

%A tal scopo, l'uso della struttura switch/case è consigliabile, in quando
%si hanno un numero di opzioni finite e note a priori

switch scelta
    case 1
        out=mean(vettore);
    case 2
        out=max(vettore);
    case 3
        out=min(vettore);
    case 4
        %in questo caso, uscire corrisponde a non eseguire nessuna
        %operazione
        disp('esco')
end

```

```

%
% Politecnico di Milano
% Corso Informatica B - Prof. Loiacono
% Esercizi di: Alessandro A. Nacci
% Per contatti: alessandro.nacci@mail.polimi.it
% Tratto da: http://www.unife.it/ing/meccanica/insegnamenti/
%             informatica-industriale/lucidi-delle-lezioni
% -----
% Esercitazione 7
% ESERCIZIO 6
% -----

```

```

% Scrivere uno script matlab che prende in ingresso
% un voto in centesimi e lo converte in una parola
% secondo la seguente tabella

```

```

%      voto > 90      OTTIMO
%  80 < voto < 89    BUONO
%  70 < voto < 79    DISCRETO
%  60 < voto < 69    SUFFICIENTE
%      voto < 60     INSUFFICIENTE

```

```
voto = 85
```

```

%Per risolvere questo esercizio, è possibile usare la struttura
%if/elseif/else.

```

```

if voto>100
    disp('Voto non valido!!!');
    parola='';
elseif voto>=90
    parola='buono';
elseif voto >=80
    parola='buono';
elseif voto >=70
    parola='discreto';
elseif voto >=60
    parola='sufficiente';
elseif voto >=0
    parola='insufficiente';
else
    %Consideriamo anche i casi non rappresentati dalle condizioni imposte
    disp('Voto non valido!!!');
    parola='';
end

```

```

%NB: non è l'unica soluzione!! Si poteva risolvere con uno switch case, o
%con degli if innidati (sconsigliato).

```