



Informatica B

2016-2016

Esercitazione I

Variabili, Matrici, Vettori, Costrutti Condizionali

Alessandro A. Nacci

alessandro.nacci@polimi.it - www.alessandronacci.it



Chi sono

■ **Alessandro Nacci, PhD**

- Nato nel 1987
- Imprenditore (Bottega52 SRL, Sofia SRL)
- Research Affiliate @ POLIMI
- Il vostro esercitatore di Informatica B

● **Mi occupo di:**

- Internet Of Things
 - Smart Buildings, Mobile Devices, Sistemi Embedded

● **Contatti:**

- www.alessandronacci.it, alessandro.nacci@polimi.it

● **Ricevimento:**

- Ci mettiamo d'accordo via e-mail





Esercizio I

Scrivere una funzione che riceve in input una stringa e restituisce true se è palindroma, false altrimenti. Scrivere anche un esempio di chiamata della funzione.



Esercizio I

```
palindroma('ingegni')  
palindroma('falsa')
```

```
function ris=palindroma(stringa)  
    % Ricavo la stringa invertita  
    stringa_r = stringa(end:-1:1);  
  
    % Se le stringhe sono uguali, la stringa palindroma  
    if all(stringa == stringa_r)  
        ris=true;  
    else  
        ris=false;  
    end
```



Esercizio 2

Scrivere una funzione che riceva in ingresso un numero N e restituisca un array contenente tutti i suoi divisori (ad eccezione di 1 ed N stesso). Nel caso N non abbia divisori, la funzione deve restituire l'array vuoto.



Esercizio 2

```
N = input ('Inserire un numero: ');

disp (['I divisori di ' num2str(N) ' sono ' mat2str(divisori(N))]);

function v = divisori(N)
%Possibili divisori
candidati = 2:N-1;

%Calcoliamo i divisori
idx = mod(N,candidati)==0;

% se ci sono divisori elimino i candidati superflui
if any(idx)
    v = candidati(idx);
else
    v=[];
end
```



Esercizio 3

Scrivere un programma che permetta all'utente di inserire un vettore di numeri interi. Dopo aver verificato che l'array inserito sia 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 e trovarne le posizioni
- Verificare se esiste un numero dispari
- Contare i numeri dispari se esistono e dire in che posizione sono



Esercizio 3

```
% Scrivere un programma che permetta all'utente di inserire un vettore di numeri interi.
vett = input('Inserisci un vettore: ');

% Dopo aver verificato che l'array inserito sia numerico, effettuare i seguenti
controlli:
if isnumeric(vett)

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

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




Esercizio 3

```
% Verificare se esiste un numero negativo
if any(vett <0)
    disp('esiste un numero negativo');
else
    disp('non esiste alcun numero negativo');
end

% Applicare la radice quadrata a tutti i valori: ci sono dei valori complessi?
sqrt_vett = sqrt(vett)
if isreal(sqrt_vett)
    disp('non esistono valori complessi');
else
    disp('esistono valori complessi');
end
```



Esercizio 3

```
% Verificare se tutti i numeri sono pari e trovarne le posizioni
vett_mod = mod(vett,2);
if(all(vett_mod==0))
    disp('tutti i numeri sono pari');
else
    disp('non tutti i numeri sono pari');
end
pos_pari = find(1-vett_mod);
disp('I numeri pari sono in posizione: ');
disp(pos_pari);

% Verificare se esiste un numero dispari
if any(vett_mod),
    disp('esiste almeno un numero dispari');

    % Contare i numeri dispari se esistono e dire in che posizioni sono
    count_dispari = sum(vett_mod);
    disp(['i numeri dispari sono ' num2str(count_dispari)]);
    pos_dispari = find(vett_mod);
    disp('i numeri dispari sono in posizione: ');
    disp(pos_dispari);
else
    disp('non esistono numeri dispari');
end
```



Esercizio 4

Si chieda all'utente di inserire una matrice di 5 righe e 5 colonne (alternativamente la si può generare in modo random).

Si stampi a video la somma di ogni riga e colonna.

Chiedere all'utente di inserire un numero di riga e si stampi la somma dei valori di solo quella riga.

Aggiungere un elemento (inserito dall'utente) alla fine della prima riga della matrice A (supponendo che non se ne conosca a priori la dimensione)



Esercizio 4

```
%  
% si chieda all'utente di inserire una matrice di 5 righe e 5 colonne  
% (alternativamente la si puo' generare in modo random)  
%  
  
A = rand(5,5)  
  
% si stampi a video la somma di ogni riga e di ogni colonna  
  
righe = sum(A,2)  
colonne = sum(A,1)  
  
% chiedere all'utente di inserire un numero di riga  
% si stampi la somma dei valori di solo quella riga  
  
n = input('inserire valore di riga')  
riga = sum(A(n,:),2)  
  
  
%%aggiungere un elemento (inserito dall'utente) alla fine della prima riga  
%%della matrice A (supponendo che non se ne conosca a priori la dimensione)  
  
A(1, size(A,2)+1) = input ('inserire un elemento')
```



Esercizio 5

```
% chiedere all'utente di inserire un vettore e un numero
%
% calcolare:
% # il numero di elementi del vettore uguali al numero inserito
% # il numero di elementi del vettore maggiori del numero inserito
% # il numero di elementi del vettore minori del numero inserito
%
% indicare poi il valore di tali elementi,
% la loro posizione del vettore
% il vettore binario per ogni operazione richiesta
```



Esercizio 5

```
%x = input('inserire un numero')
%vettore = input('inserisci un vettore')

x = 10
vettore = [10 20 30 50 60]

%vettore binario
Buguali= vettore==x
Bmaggiori= vettore>x
Bminori= vettore<x

%valore degli elementi
Vuguali= vettore(vettore==x)
Vmaggiori= vettore(vettore>x)
Vminori= vettore(vettore<x)

%numero di elementi
Nuguali=size( vettore (vettore==x),2)
Nmaggiori=size( vettore (vettore>x),2)
Nminori=size( vettore (vettore<x),2)

%alternativa:
% Nmaggiori = sum(vettore > x)

%posizione degli elementi
Puguali= find(vettore==x)
Pmaggiori= find(vettore>x)
Pminori= find(vettore<x)
```

**Tutte il materiale sarà
disponibile sul mio sito
internet!**

www.alessandronacci.it

See You Next Time!

