

Classi di istruzioni

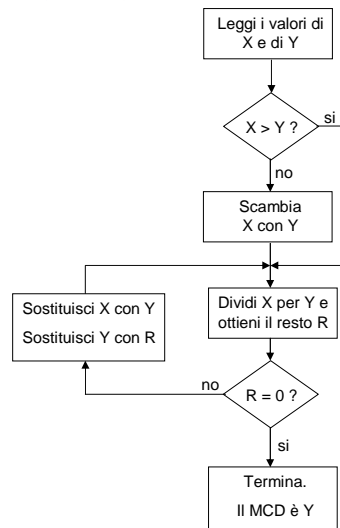
- In maniera simile a quanto fatto per i dati, un linguaggio mette a disposizione dei costrutti per realizzare la parte esecutiva dell'algoritmo.
- Questa consiste di:
 - assegnazioni di valori a variabili (in base a calcolo o da I/O)
 - selezione di azioni alternative in base alla valutazione di una condizione
 - esecuzione ciclica di una o più azioni
- I costrutti del linguaggio si dividono in corrispondenti classi di istruzioni

F. Tortorella

Corso di Elementi di Informatica
2006/2007

Università degli Studi
di Cassino

Classi di istruzioni



F. Tortorella

Corso di Elementi di Informatica
2006/2007

Università degli Studi
di Cassino

Istruzioni di calcolo e assegnazione

- L'effetto è di aggiornare il valore di una variabile di un certo tipo con il valore ottenuto dalla valutazione di un'espressione dello stesso tipo.

- Il formato è:

`variabile = espressione`

- Esempi:

`a=4` `a=a+1` `cond= x > y`
`b=0` `a=a+b` `cond=(a>=0) & (a<=9)`
`b=a`

F. Tortorella

Corso di Elementi di Informatica
2006/2007

Università degli Studi
di Cassino

Istruzioni di Input/Output

- Con le istruzioni di input, il valore di una variabile viene modificato con il valore ottenuto grazie ad un'operazione di lettura dall'unità di ingresso (tastiera).

Con le istruzioni di output, un'espressione viene valutata ed il valore ottenuto viene presentato sull'unità di uscita (monitor).

F. Tortorella

Corso di Elementi di Informatica
2006/2007

Università degli Studi
di Cassino

Istruzioni di input

- Con l'istruzione **input** è possibile richiedere all'utente di inserire un valore ed inizializzare una variabile con il valore inserito.
Es.:
`x = input('Fornire il valore: ');`
- Quando l'istruzione è eseguita, Matlab stampa la stringa 'Fornire il valore: ' e si mette in attesa della risposta dell'utente.
- L'utente scrive un numero con la tastiera e alla fine batte il tasto INVIO.
- A questo punto, il valore letto viene trasferito nella variabile `x`.

F. Tortorella

Corso di Elementi di Informatica
2006/2007

Università degli Studi
di Cassino

Istruzioni di Output

- L'istruzione **disp** permette di stampare a video il contenuto di una variabile.
- L'utente non può controllare la modalità di stampa.
- La forma generale è
`disp(var);`

F. Tortorella

Corso di Elementi di Informatica
2006/2007

Università degli Studi
di Cassino

Istruzioni di Output

- L'istruzione **fprintf** permette di stampare a video il contenuto di una o più variabili, insieme con del testo relativo.
- L'utente può controllare la modalità di stampa.
- La forma generale è
`fprintf(format,data);`

F. Tortorella

Corso di Elementi di Informatica
2006/2007

Università degli Studi
di Cassino

Istruzioni di Output

- *format* è una stringa che descrive l'organizzazione dell'output
- Contiene testo da stampare e caratteri speciali che descrivono il formato dei dati
- Il formato viene definito in base a speciali sequenze di caratteri, definiti *caratteri di conversione*

F. Tortorella

Corso di Elementi di Informatica
2006/2007

Università degli Studi
di Cassino

Sequenze di caratteri di formato

Sequenza	Risultato
%d	Visualizza il valore come un intero
%e	Visualizza il valore in virgola mobile
%f	Visualizza il valore in virgola fissa
%g	Visualizza il valore nel formato più breve tra virgola fissa e mobile
\n	Va all'inizio della linea successiva

F. Tortorella

Corso di Elementi di Informatica
2006/2007

Università degli Studi
di Cassino

Output. Esempio

```
>> x=pi/2;  
>> fprintf('Il risultato e': %f\n',x);  
Il risultato e': 1.570796
```

F. Tortorella

Corso di Elementi di Informatica
2006/2007

Università degli Studi
di Cassino

Esempi

- [Scambio dei valori di due variabili](#)
- Soluzione di un sistema di due equazioni lineari in due incognite
 - [Versione 1](#)
 - [Versione 2](#)

Istruzioni selettive

- Permettono di selezionare insiemi di istruzioni alternativi in base alla valutazione di una o più condizioni

Istruzioni selettive: if

- Sintassi

```
if(condizione)
  istruzione_1
  istruzione_2
  ...
  istruzione_n
end
```

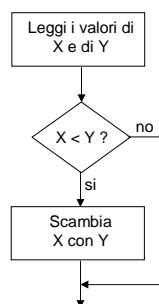
Le istruzioni sono eseguite solo se *condizione* è vera (*condizione* == 1)

F. Tortorella

Corso di Elementi di Informatica
2006/2007

Università degli Studi
di Cassino

Esempio



```
x = input("");
y = input("");
```

```
if(x < y)
  % scambia x e y
  Z = X;
  X = y;
  y = Z;
end
```

F. Tortorella

Corso di Elementi di Informatica
2006/2007

Università degli Studi
di Cassino

Istruzioni selettive: if...else

- Sintassi

if(condizione)

istruzione_1

istruzione_2



istruzioni eseguite se
condizione è vera

else

istruzione_3

istruzione_4



istruzioni eseguite se
condizione è falsa

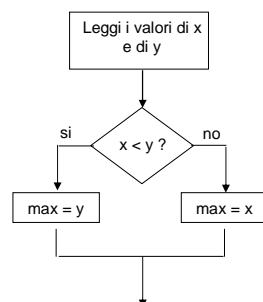
end

F. Tortorella

Corso di Elementi di Informatica
2006/2007

Università degli Studi
di Cassino

Esempio: qual è il max fra due ?



```
x = input("");
```

```
y = input("");
```

```
if(x > y)
```

```
  % il max è x
```

```
  max = x;
```

```
else
```

```
  % il max è y
```

```
  max = y;
```

```
end
```

```
fprintf('Max: %d\n',max);
```

F. Tortorella

Corso di Elementi di Informatica
2006/2007

Università degli Studi
di Cassino

Esempio

- Soluzione di un sistema di due equazioni lineari in due incognite
 - [Versione 3](#): verifica se il determinante è nullo

F. Tortorella

Corso di Elementi di Informatica
2006/2007

Università degli Studi
di Cassino

Esempio: qual è il max fra tre ?

```
x = input(""); y = input(""); z = input("");
```

```
max = x;
```

```
if(y > max)  
    max = y;  
end
```

```
if(z > max)  
    max = z;  
end
```

```
fprintf('Max: %d\n',max);
```

F. Tortorella

Corso di Elementi di Informatica
2006/2007

Università degli Studi
di Cassino

Istruzioni selettive: if...else if ... else

- Sintassi

```
if(condizione_1)
    blocco_1
else if(condizione_2)
    blocco_2
else if(condizione_3)
    blocco_3
else
    blocco_4
end
```

eseguito solo se *condizione_1* è vera

eseguito solo se *condizione_1* è falsa e *condizione_2* è vera

eseguito solo se *condizione_1* è falsa, *condizione_2* è falsa e *condizione_3* è vera

eseguito solo se *condizione_1* è falsa, *condizione_2* è falsa e *condizione_3* è falsa

F. Tortorella

Corso di Elementi di Informatica
2006/2007

Università degli Studi
di Cassino

Esempio

```
voto = input('Voto ricevuto: ');
if(voto < 18)
    fprintf('Ritorna\n');
else if(voto < 24)
    fprintf('Si può dare di più\n');
else if(voto < 27)
    fprintf('Non c"è male\n');
else if(voto < 30)
    fprintf('C"è mancato poco\n');
else if(voto == 30)
    fprintf('Finalmente ci siamo\n');
else
    fprintf(' WOW !\n');
end
```

F. Tortorella

Corso di Elementi di Informatica
2006/2007

Università degli Studi
di Cassino

Istruzioni selettive: switch

- Sintassi

```
switch(espr_sw)
  case espr_1
    blocco_1
  case espr_2
    blocco_2
  case {espr_3,espr_4}
    blocco_3
  otherwise
    blocco_4
end
```

eseguito se $espr_sw == espr_1$

eseguito se $espr_sw <> espr_1$ e $espr_sw == espr_2$

eseguito se $espr_sw <> espr_1$, $espr_sw <> espr_2$ e $espr_sw == espr_3$ o $espr_sw == espr_4$

eseguito se $espr_sw$ è diverso da tutte le $expr_i$ nei **case**

F. Tortorella

Corso di Elementi di Informatica
2006/2007

Università degli Studi
di Cassino

Esempio

```
switch(mese)
  case 2
    ngiorni=28;
  case {4,6,9,11}
    ngiorni=30;
  otherwise
    ngiorni=31;
end
```

F. Tortorella

Corso di Elementi di Informatica
2006/2007

Università degli Studi
di Cassino

Esempio

```
switch(car)
  case '.'
    fprintf('punto\n');
  case ','
    fprintf('virgola\n');
  case {'a','e','i','o','u'}
    fprintf('vocale\n');
  otherwise
    fprintf('consonante\n');
end
```

F. Tortorella

Corso di Elementi di Informatica
2006/2007

Università degli Studi
di Cassino

Problema

- Scrivere un programma che legga da input i coefficienti a , b , c di un'equazione di secondo grado e ne calcoli le radici.
- Considerare i casi in cui uno o più dei coefficienti sia nullo.
- Soluzione
 - [Step 1](#)
 - [Step 2](#)
 - [Step 3](#)
 - [Step 4](#)
 - [Step 5](#)

F. Tortorella

Corso di Elementi di Informatica
2006/2007

Università degli Studi
di Cassino

Istruzioni cicliche

- Servono a ripetere l'esecuzione di un blocco di istruzioni
- A seconda di come viene definito il numero di ripetizioni dell'esecuzione, si distinguono in
 - Istruzioni cicliche a condizione
 - Istruzioni cicliche a conteggio

F. Tortorella

Corso di Elementi di Informatica
2006/2007

Università degli Studi
di Cassino

Istruzioni cicliche: while

- E' un costrutto ciclico *a condizione*
- Non si definisce esplicitamente il numero di ripetizioni dell'esecuzione, ma si valuta all'inizio del ciclo un'espressione logica che, fin quando risulta vera, causa un'ulteriore esecuzione del blocco di istruzioni.

F. Tortorella

Corso di Elementi di Informatica
2006/2007

Università degli Studi
di Cassino

Istruzioni cicliche: while

- Sintassi

```
while(condizione)
    istruzione_1
    istruzione_2
    ...
    istruzione_n
end
```
- Si valuta la condizione
- Se risulta vera, si eseguono le istruzioni; dopo l'esecuzione dell'ultima istruzione sotto il while, si torna a verificare la condizione
- Se la condizione risulta falsa, si passa a eseguire le istruzioni che si trovano dopo la chiusura del while
- Qual è il minor numero di cicli che si può effettuare ?

F. Tortorella

Corso di Elementi di Informatica
2006/2007

Università degli Studi
di Cassino

Esempio

- Stampare in output i primi 10 numeri naturali.

F. Tortorella

Corso di Elementi di Informatica
2006/2007

Università degli Studi
di Cassino

Esempio

- Stampare in output i primi 10 numeri naturali.

```
x=1;
while(x<=10)
    printf('x: %d\n',x);
    x=x+1;
end
```

Esempio

- Leggere da input un insieme di numeri interi e calcolarne la somma. Non si conosce in anticipo la quantità di valori da leggere; la lettura di un valore == 0 indica che l'insieme da leggere è terminato.

Esempio

- Leggere da input un insieme di numeri interi e calcolarne la somma. Non si conosce in anticipo la quantità di valori da leggere; la lettura di un valore `== 0` indica che l'insieme da leggere è terminato.

```
x=input('Valore: ');
somma=0;
while(x~=0)
    somma=somma+x;
    x=input('Valore: ');
end
fprintf('Somma: %d\n',
        somma);
```

F. Tortorella

Corso di Elementi di Informatica
2006/2007

Università degli Studi
di Cassino

Problema

Leggere da input un insieme di numeri reali e calcolarne la media. Non si conosce in anticipo la quantità di valori da leggere, che comunque è limitata ad un massimo di 50; la lettura di un valore `< 0` indica che l'insieme da leggere è terminato.

[soluzione](#)

F. Tortorella

Corso di Elementi di Informatica
2006/2007

Università degli Studi
di Cassino

Problema ricerca del minimo e del massimo

- **Problema 1**

Leggere da input un insieme di numeri reali ≥ 0 e determinare il valore minimo. Non si conosce in anticipo la quantità di valori da leggere; la lettura di un valore < 0 indica che l'insieme da leggere è terminato. [soluzione](#)

- **Problema 2**

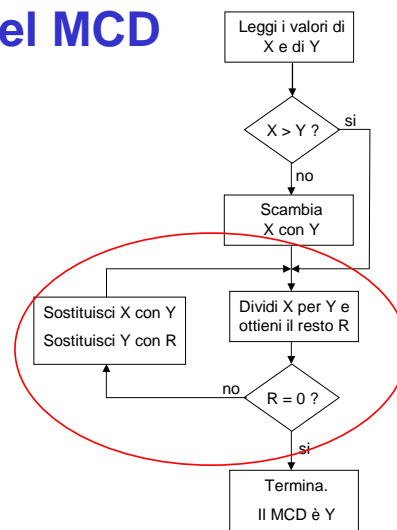
Nelle stesse ipotesi del problema 1, determinare il valore massimo dell'insieme dei valori letti.

F. Tortorella

Corso di Elementi di Informatica
2006/2007

Università degli Studi
di Cassino

Problema calcolo del MCD



ACHTUNG !!!

**Non è un ciclo
WHILE**

Come fare ?

[soluzione](#)

F. Tortorella

Corso di Elementi di Informatica
2006/2007

Università degli Studi
di Cassino

Istruzioni cicliche: for

- E' un costrutto ciclico *a conteggio*
- Si definisce esplicitamente il numero di ripetizioni dell'esecuzione
- Il conteggio viene gestito grazie ad una variabile (*variabile di conteggio*) che assume un valore iniziale e viene incrementata di un valore fisso ad ogni ripetizione del ciclo finché non raggiunge o supera un valore finale.

F. Tortorella

Corso di Elementi di Informatica
2006/2007

Università degli Studi
di Cassino

Istruzioni cicliche: for

- Sintassi

```
for var = val_in:step:val_fin  
    istruzione_1  
    istruzione_2  
    ...  
    istruzione_n  
end
```
- Si inizializza la variabile di ciclo
- Si verifica se il suo valore è maggiore di *val_fin*
- Se la variabile di ciclo è minore del valore finale si eseguono le istruzioni sotto il ciclo for; al termine dell'esecuzione, la variabile di ciclo viene incrementata di *step* e si torna a fare il confronto con *val_fin*
- Se la variabile di ciclo è maggiore di *val_fin*, il ciclo termina e si eseguono le istruzioni che seguono il for

F. Tortorella

Corso di Elementi di Informatica
2006/2007

Università degli Studi
di Cassino

Esempio

- Stampare in output i primi 10 numeri naturali.

F. Tortorella

Corso di Elementi di Informatica
2006/2007

Università degli Studi
di Cassino

Esempio

- Stampare in output i primi 10 numeri naturali.

```
for x=1:10
    fprintf('x: %d\n',x);
end
```

```
x=1;
while(x<=10)
    fprintf('x: %d\n',x);
    x=x+1;
end
```

F. Tortorella

Corso di Elementi di Informatica
2006/2007

Università degli Studi
di Cassino

Esempio

- Stampare in output i primi 100 numeri dispari.

F. Tortorella

Corso di Elementi di Informatica
2006/2007

Università degli Studi
di Cassino

Esempio

- Stampare in output i primi 100 numeri dispari.

```
for x=1:2:100
    fprintf('x: %d\n',x);
end
```

F. Tortorella

Corso di Elementi di Informatica
2006/2007

Università degli Studi
di Cassino

Problema

- Leggere da input un insieme di numeri interi e calcolarne la somma. Il numero di valori da leggere è fornito in ingresso prima della sequenza di valori

[soluzione](#)

F. Tortorella

Corso di Elementi di Informatica
2006/2007

Università degli Studi
di Cassino

Problema

- Stampare la “tabellina” di n , dove n è dato in input
- Stampare le “tabelline” dei valori compresi tra 1 e 10.

F. Tortorella

Corso di Elementi di Informatica
2006/2007

Università degli Studi
di Cassino

Problema

- Realizzare un programma che, letti due valori x ed n da input, calcoli x^n .
- n può essere negativo o nullo

[soluzione](#)

```
function main
```

```
% Lo scopo di questo programma è di mostrare le istruzioni  
% di calcolo e assegnazione e le istruzioni di I/O
```

```
% variabili usate nel programma
```

```
x=0; y=0; z=0;
```

```
% fase di lettura
```

```
x=input('Primo valore: ');
```

```
y=input('Secondo valore: ');
```

```
% scambio dei valori
```

```
z=x;
```

```
x=y;
```

```
y=z;
```

```
% fase di output
```

```
fprintf('\nSituazione dopo lo scambio: \n');
```

```
fprintf('Primo valore: %d\n',x);
```

```
fprintf('Secondo valore: %d\n',y);
```

```
function main
% Risoluzione di un sistema di due equazioni lineari
% in due incognite
% versione 1.0

% variabili impiegate nel programma

% variabili contenenti coefficienti e termini noti
a=0; b=0; c=0; d=0; e=0; f=0;

% variabili contenenti le incognite
x=0; y=0;

% fase di input
a=input('Dammi il valore di a: ');
b=input('Dammi il valore di b: ');
c=input('Dammi il valore di c: ');
d=input('Dammi il valore di d: ');
e=input('Dammi il valore di e: ');
f=input('Dammi il valore di f: ');

% calcolo dei valori delle due incognite
x=(c*e-b*f)/(a*e-b*d);
y=(a*f-c*d)/(a*e-b*d);

% fase di output
fprintf('Il valore di x è: %f\n',x);
fprintf('Il valore di y è: %f\n',y);
```



```

function main
% Risoluzione di un sistema di due equazioni lineari
% in due incognite
% versione 1.1

% variabili impiegate nel programma

% variabili contenenti coefficienti e termini noti
a=0; b=0; c=0; d=0; e=0; f=0;

% variabili contenenti le incognite
x=0; y=0;

% variabile contenente il determinante
det=0;

% fase di input
a=input('Dammi il valore di a: ');
b=input('Dammi il valore di b: ');
c=input('Dammi il valore di c: ');
d=input('Dammi il valore di d: ');
e=input('Dammi il valore di e: ');
f=input('Dammi il valore di f: ');

% calcolo del determinante e dei valori delle due incognite
det=(a*e-b*d);
x=(c*e-b*f)/det;
y=(a*f-c*d)/det;

% fase di output
fprintf('Il valore di x è: %f\n',x);
fprintf('Il valore di y è: %f\n',y);

```

```

function main
% Risoluzione di un sistema di due equazioni lineari
% in due incognite
% versione 2.0

% variabili impiegate nel programma

% variabili contenenti coefficienti e termini noti
a=0; b=0; c=0; d=0; e=0; f=0;

% variabili contenenti le incognite
x=0; y=0;

% variabile contenente il determinante
det=0;

% fase di input
a=input('Dammi il valore di a: ');
b=input('Dammi il valore di b: ');
c=input('Dammi il valore di c: ');
d=input('Dammi il valore di d: ');
e=input('Dammi il valore di e: ');
f=input('Dammi il valore di f: ');

% calcolo del determinante
det=(a*e-b*d);

% verifica se il determinante è nullo
if(det~=0)
    % calcolo dei valori delle due incognite
    x=(c*e-b*f)/det;
    y=(a*f-c*d)/det;

    % fase di output
    fprintf('Il valore di x è: %f\n',x);
    fprintf('Il valore di y è: %f\n',y);
else
    fprintf('Sistema non risolubile\n');

```

```
function main
```

```
% Vengono calcolate le radici di un'equazione di 2° grado  $ax^2+bx+c=0$ .
```

```
% Sviluppo top-down del programma: passo 1
```

```
% variabili usate nel programma
```

```
a=0; b=0; c=0; % coefficienti dell'equazione letti in input
```

```
d=0; % discriminante dell'equazione
```

```
x1=0; x2=0; % radici dell'equazione
```

```
% fase di lettura
```

```
fprintf('Fornire i coefficienti dell'equazione  $ax^2+bx+c=0$ \n');
```

```
a=input('a: ');
```

```
b=input('b: ');
```

```
c=input('c: ');
```

```
% si verifica il valore del coefficiente a
```

```
if(a~=0)
```

```
    % si applica il metodo di soluzione generale
```

```
elseif(b~=0)
```

```
    % è un'equazione di 1° grado
```

```
elseif(c==0)
```

```
    % è un'equazione indeterminata
```

```
else
```

```
    % è un'equazione impossibile
```

```
end
```

```
function main
```

```
% Vengono calcolate le radici di un'equazione di 2° grado  $ax^2+bx+c=0$ .
```

```
% Sviluppo top-down del programma: passo 2
```

```
% variabili usate nel programma
```

```
a=0; b=0; c=0; % coefficienti dell'equazione letti in input
```

```
d=0; % discriminante dell'equazione
```

```
x1=0; x2=0; % radici dell'equazione
```

```
% fase di lettura
```

```
fprintf('Fornire i coefficienti dell'equazione  $ax^2+bx+c=0$ \n');
```

```
a=input('a: ');
```

```
b=input('b: ');
```

```
c=input('c: ');
```

```
% si verifica il valore del coefficiente a
```

```
if(a~=0)
```

```
    % si applica il metodo di soluzione generale
```

```
elseif(b~=0)
```

```
    % è un'equazione di 1° grado
```

```
elseif(c==0)
```

```
    % è un'equazione indeterminata
```

```
    fprintf('Equazione indeterminata\n');
```

```
else
```

```
    % è un'equazione impossibile
```

```
    fprintf('Equazione impossibile\n');
```

```
end
```

```
function main
```

```
% Vengono calcolate le radici di un'equazione di 2° grado  $ax^2+bx+c=0$ .
```

```
% Sviluppo top-down del programma: passo 3
```

```
% variabili usate nel programma
```

```
a=0; b=0; c=0; % coefficienti dell'equazione letti in input
```

```
d=0; % discriminante dell'equazione
```

```
x1=0; x2=0; % radici dell'equazione
```

```
% fase di lettura
```

```
fprintf('Fornire i coefficienti dell'equazione  $ax^2+bx+c=0$ \n');
```

```
a=input('a: ');
```

```
b=input('b: ');
```

```
c=input('c: ');
```

```
% si verifica il valore del coefficiente a
```

```
if(a~=0)
```

```
    % si applica il metodo di soluzione generale
```

```
elseif(b~=0)
```

```
    % è un'equazione di 1° grado
```

```
    x1=-c/b;
```

```
    fprintf('Equazione di 1° grado, unica radice: %f\n',x1);
```

```
elseif(c==0)
```

```
    % è un'equazione indeterminata
```

```
    fprintf('Equazione indeterminata\n');
```

```
else
```

```
    % è un'equazione impossibile
```

```
    fprintf('Equazione impossibile\n');
```

```
end
```

```
function main
```

```
% Vengono calcolate le radici di un'equazione di 2° grado  $ax^2+bx+c=0$ .
```

```
% Sviluppo top-down del programma: passo 4
```

```
% variabili usate nel programma
```

```
a=0; b=0; c=0; % coefficienti dell'equazione letti in input
```

```
d=0; % discriminante dell'equazione
```

```
x1=0; x2=0; % radici dell'equazione
```

```
% fase di lettura
```

```
fprintf('Fornire i coefficienti dell'equazione  $ax^2+bx+c=0$ \n');
```

```
a=input('a: ');
```

```
b=input('b: ');
```

```
c=input('c: ');
```

```
% si verifica il valore del coefficiente a
```

```
if(a~=0)
```

```
    % si applica il metodo di soluzione generale
```

```
    % si calcola il discriminante
```

```
    d=b*b-4*a*c;
```

```
    % si valuta il tipo delle radici
```

```
    if(d > 0)
```

```
        % due radici reali distinte
```

```
    elseif(d==0)
```

```
        % due radici reali coincidenti
```

```
    else
```

```
        % due radici complesse coniugate
```

```
    end
```

```
elseif(b~=0)
```

```
    % è un'equazione di 1° grado
```

```
    x1=-c/b;
```

```
    fprintf('Equazione di 1° grado, unica radice: %f\n',x1);
```

```
elseif(c==0)
```

```
    % è un'equazione indeterminata
```

```
    fprintf('Equazione indeterminata\n');
```

```
else
```

```
    % è un'equazione impossibile
```

```
    fprintf('Equazione impossibile\n');
```

```
end
```

```
function main
```

```
% Vengono calcolate le radici di un'equazione di 2° grado  $ax^2+bx+c=0$ .
```

```
% Sviluppo top-down del programma: passo 5. Versione finale
```

```
% variabili usate nel programma
```

```
a=0; b=0; c=0; % coefficienti dell'equazione letti in input
```

```
d=0; % discriminante dell'equazione
```

```
x1=0; x2=0; % radici dell'equazione
```

```
% fase di lettura
```

```
printf('Fornire i coefficienti dell'equazione  $ax^2+bx+c=0$ ');
```

```
a=input('a: ');
```

```
b=input('b: ');
```

```
c=input('c: ');
```

```
% si verifica il valore del coefficiente a
```

```
if(a~=0)
```

```
    % si applica il metodo di soluzione generale
```

```
    % si calcola il discriminante
```

```
    d=b*b-4*a*c;
```

```
    % si valuta il tipo delle radici
```

```
    if(d > 0)
```

```
        % due radici reali distinte
```

```
        x1=(-b-sqrt(d))/(2*a);
```

```
        x2=(-b+sqrt(d))/(2*a);
```

```
        printf('L'equazione ha due radici reali distinte\n');
```

```
        printf('x1: %f\nx2: %f\n',x1,x2);
```

```
    elseif(d==0)
```

```
        % due radici reali coincidenti
```

```
        x1=(-b)/(2*a);
```

```
        printf('L'equazione ha due radici reali coincidenti\n');
```

```
        printf('x1: %f\n',x1);
```

```
    else
```

```
        % due radici complesse coniugate
```

```
        % si calcola la parte reale ed il coefficiente dell'immaginario
```

```
        x1=(-b)/(2*a); % parte reale
```

```
        x2=sqrt(-d)/(2*a); % coefficiente dell'immaginario
```

```
        printf('L'equazione ha due radici complesse coniugate\n');
```

```
        printf('x1: %f + j %f\n',x1,x2);
```

```
        printf('x2: %f - j %f\n',x1,x2);
```

```
    end
```

```
elseif(b~=0)
```

```
    % è un'equazione di 1° grado
```

```
    x1=-c/b;
```

```
    printf('Equazione di 1° grado, unica radice: %f\n',x1);
```

```
elseif(c==0)
```

```
    % è un'equazione indeterminata
```

```
    printf('Equazione indeterminata\n');
```

```
else
```

```
    % è un'equazione impossibile
```

```
    printf('Equazione impossibile\n');
```

```
end
```

function main

% Viene calcolata la media di un insieme di valori reali letti da input.
% Non si conosce in anticipo la quantità di valori da leggere, che comunque
% è limitata ad un massimo di 50; la lettura di un valore < 0 indica che
% l'insieme da leggere è terminato.

% variabili usate nel programma

x=0; % valore letto in input

cont=0; % tiene traccia del numero di valori letti

somma=0; % contiene la somma corrente

media=0; % contiene la media calcolata

% fase di lettura

x=input('Valore: ');

while((cont<50) & (x>=0))

 somma=somma+x;

 cont=cont+1;

 x=input('Valore: ');

end

% si calcola la media

media=somma/cont;

% fase di output

fprintf('\nValori letti: %d\n',cont);

fprintf('Media: %f\n',media);


```
function main
```

```
% Viene calcolato il minimo di un insieme di valori reali  $\geq 0$  letti da input.  
% Non si conosce in anticipo la quantità di valori da leggere; la lettura di  
% un valore  $< 0$  indica che l'insieme da leggere è terminato.
```

```
% variabili usate nel programma
```

```
x=0; % valore letto in input
```

```
cont=0; % tiene traccia del numero di valori letti
```

```
min=0; % contiene la somma corrente
```

```
posmin=0; % contiene la posizione del minimo trovato
```

```
% fase di lettura
```

```
x=input('Valore: ');
```

```
% inizializzazione della ricerca
```

```
min=x;
```

```
posmin=1;
```

```
while(x $\geq$ 0)
```

```
    cont=cont+1;
```

```
    if(x<min)
```

```
        min=x;
```

```
        posmin=cont;
```

```
    end
```

```
    x=input('Valore: ');
```

```
end
```

```
% fase di output
```

```
fprintf('\nValori letti: %d\n',cont);
```

```
fprintf('Valore minimo: %f\n',min);
```

```
fprintf('Posizione del minimo: %d\n',posmin);
```

```

function main
% Viene calcolato il MCD tra de numeri interi letti da input.

% variabili usate nel programma
xin=0; yin=0;% valori letti in input
x=0; y=0; r=0; % variabili usate nell'algoritmo

% fase di lettura
xin=input('Primo valore: ');
yin=input('Secondo valore: ');

% si assegnano i valori letti alle variabili di algoritmo
x=xin;
y=yin;

% verifica se x >= y ed, eventualmente, scambia le due variabili
if(x<y)
    appo=x;
    x=y;
    y=x;
end

r=mod(x,y);

while(r~=0)
    x=y;
    y=r;
    r=mod(x,y);
end

% fase di output

fprintf('Il MCD tra %d e %d è %d\n',xin,yin,y);

```

```
function main
```

```
% Si esegue la somma di un insieme di valori letti da input  
% il numero di elementi da leggere è fornito in ingresso prima dell'inizio  
% della sequenza
```

```
% variabili
```

```
n=0; % numero di elementi da leggere
```

```
x=0; % elemento corrente
```

```
s=0; % somma corrente
```

```
i=0; % indice di ciclo
```

```
% Input
```

```
% si ottiene il numero degli elementi da leggere in input
```

```
n=input('Numero valori: ');
```

```
% inizializzazione della var. somma
```

```
s=0;
```

```
% ciclo di lettura e somma
```

```
for i=1:n
```

```
    fprintf('Valore %d: ',i);
```

```
    x=input("");
```

```
    s=s+x;
```

```
end
```

```
% messaggio di output
```

```
fprintf('Letti %d valori.\nLa loro somma è %g\n',n,s);
```

```

function main
% Si esegue il calcolo della potenza x^n dove x ed n sono due valori
% letti in input. n può essere <=0.

% variabili
n=0; % esponente
x=0; % base

p=0; % valore corrente della potenza
i=0; % indice di ciclo

% Input
% si ottengono i valori di x ed n
x=input('x: ');
n=input('n: ');

% si inizializza la variabile che contiene il valore della potenza
p=1;

% ciclo per calcolare la potenza come produttoria
% il valore finale è valutato in valore assoluto perchè n può essere <0
for i=1:abs(n)
    p=p*x;
end

% si corregge il valore della potenza se n < 0
if(n<0)
    p=1/p;
end

% stampa del risultato
fprintf('Il risultato della potenza %g^%d è %g\n',x,n,p);

```