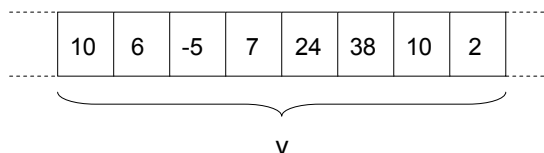


Gli array

- In alcuni casi, l'informazione che bisogna elaborare consiste di un'aggregazione di valori, piuttosto che di un valore solo.
- Questo significa che sarebbe conveniente indicare l'insieme di valori di interesse con una sola variabile piuttosto che con tante variabili quante sono i valori da considerare: una variabile di *tipo strutturato*.
- In Matlab (come nella maggior parte dei linguaggi), questa possibilità è offerta dagli **array**.

Gli array

- Un array è un insieme di variabili, tutte dello stesso tipo, identificato da un nome unico. Gli elementi dell'array sono disposti in memoria in posizioni consecutive.



Definizione di un array

- Come accade per le variabili di tipo non strutturato, gli array in Matlab possono essere definiti dinamicamente.
- Per motivi di efficienza e di leggibilità del codice, è però opportuno operare un'inizializzazione esplicita prima che la variabile array sia usata.
- Per inizializzare un array, è necessario specificare:
 - il nome della variabile array
 - il numero degli elementi presenti (cardinalità dell'array)

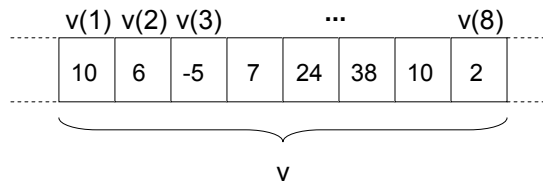
Esempio

- Inizializzazione di una variabile array *v* contenente 20 elementi double:

```
v = zeros(20,1);
```

Accesso agli elementi dell'array

- Per accedere ai singoli elementi di un array, è necessario specificare il nome della variabile array e la posizione dell'elemento di interesse tramite un valore intero (variabile o costante) che si definisce *indice*.



Accesso agli elementi dell'array

- Ogni elemento di un array è, a tutti gli effetti, una variabile del tipo costituente l'array.

`v(2)=3`

`fprintf('il valore è: %g\n', v(7))`

`i=3`

`a=v(i)*4+6`

Lettura e stampa degli elementi di un array

- Per inizializzare da input una variabile array, è necessario realizzare un'operazione di input per ciascuno degli elementi
- Analogamente, per stampare il contenuto di un array, è necessario fare la stampa di ognuno degli elementi.
- Qual è il costrutto da utilizzare ?
- Esempio:
 - leggere da input la dimensione e gli elementi di un array e stampare il risultato della lettura

Esempio: ricerca del massimo

- Leggere da input la dimensione e gli elementi di un array; fornire in uscita valore e posizione dell'elemento di valore massimo.

Soluzione

```
function main
```

```
% si realizza la lettura da input di un array
```

```
% la dimensione dell'array è fornita dall'utente
```

```
% variabili utilizzate
```

```
v=zeros(10,1); % array da leggere; inizialmente si assume una dimensione di 10
```

```
i=0; % indice per scorrere gli elementi dell'array
```

```
n=0; % dimensione dell'array fornito in input
```

```
% input dimensione
```

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

```
% dimensionamento array
```

```
v=zeros(n,1);
```

```
% ciclo di lettura
```

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

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

```
    v(i)=input("");
```

```
end
```

```
% stampa dell'array letto
```

```
fprintf('\nArray letto:\n');
```

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

```
    fprintf('V(%d): %g\n',i,v(i));
```

```
end
```

```

function main
% lettura di un array e ricerca del massimo
% il programma fornisce in uscita valore e posizione del massimo

% variabili utilizzate
v=zeros(10,1); % array da leggere
    % inizialmente si assume una dimensione di 10 elementi
i=0; % indice per scorrere gli elementi dell'array
n=0; % dimensione dell'array fornito in input
max=0; % variabile contenente il valore massimo
posmax=0; %variabile contenente la posizione del valore massimo

% input dimensione
n=input('Numero elementi: ');

% dimensionamento array
v=zeros(n,1);

% ciclo di lettura
for i=1:n
    fprintf('Valore %d: ',i);
    v(i)=input("");
end

% ricerca del massimo e del minimo
max=v(1);
posmax=1;

for i=2:n
    if(v(i)>max)
        max=v(i);
        posmax=i;
    end
end

% stampa dei risultati
fprintf('\nArray letto:\n');
for i=1:n
    fprintf('V(%d): %g\n',i,v(i));
end
fprintf('\nIl massimo è %g e si trova in posizione %d\n',max,posmax);

```

Esempio: ricerca del massimo e del minimo

- Leggere da input la dimensione e gli elementi di un array; fornire in uscita valori e posizioni dell'elemento di valore massimo e dell'elemento di valore minimo.

[Soluzione](#)

Esempio: calcolo della media

- Leggere da input la dimensione e gli elementi di un array; fornire in uscita la media degli elementi presenti nell'array.

[Soluzione](#)

```

function main
% lettura di un array e ricerca del massimo e del minimo
% il programma fornisce in uscita valori e posizioni del massimo e del
% minimo

% variabili utilizzate
v=zeros(10,1); % array da leggere
    % inizialmente si assume una dimensione di 10 elementi
i=0; % indice per scorrere gli elementi dell'array
n=0; % dimensione dell'array fornito in input
max=0; % variabile contenente il valore massimo
posmax=0; %variabile contenente la posizione del valore massimo
min=0; % variabile contenente il valore massimo
posmin=0; %variabile contenente la posizione del valore massimo

% input dimensione
n=input('Numero elementi: ');

% dimensionamento array
v=zeros(n,1);

% ciclo di lettura
for i=1:n
    fprintf('Valore %d: ',i);
    v(i)=input("");
end

% ricerca del massimo e del minimo
max=v(1);
posmax=1;

min=v(1);
posmin=1;

for i=2:n
    if(v(i)>max)
        max=v(i);
        posmax=i;
    elseif(v(i)<min)
        min=v(i);
        posmin=i;
    end
end

% stampa dei risultati
fprintf('\nArray letto:\n');
for i=1:n
    fprintf('V(%d): %g\n',i,v(i));
end
fprintf('\nIl massimo è %g e si trova in posizione %d\n',max,posmax);
fprintf('\nIl minimo è %g e si trova in posizione %d\n',min,posmin);

```



```

function main
% lettura di un array e calcolo della media
% il programma fornisce in uscita il valore della media

% variabili utilizzate
v=zeros(10,1); % array da leggere; dimensione iniziale di 10 elementi
i=0; % indice per scorrere gli elementi dell'array
n=0; % dimensione dell'array fornito in input
somma=0; % variabile contenente la somma degli elementi
media=0; % variabile contenente la media degli elementi

% input dimensione
n=input('Numero elementi: ');

% dimensionamento array
v=zeros(n,1);

% ciclo di lettura
for i=1:n
    fprintf('Valore %d: ',i);
    v(i)=input("");
end

% calcolo della somma
somma=0;
for i=1:n
    somma=somma+v(i);
end

% calcolo della media
media=somma/n;

% stampa dei risultati
fprintf('\nArray letto:\n');
for i=1:n
    fprintf('V(%d): %g\n',i,v(i));
end
fprintf('\nIl valor medio è %g\n',media);

```

Array bidimensionali

- Finora abbiamo considerato *array monodimensionali*, i quali richiedono un solo indice per l'individuazione di un elemento.
- Matlab mette a disposizione anche *array bidimensionali*, in cui l'organizzazione degli elementi è di tipo matriciale.
- In questo caso, sono necessari più indici per identificare un elemento nell'array.
- Questo tipo strutturato permette di affrontare tutte quelle situazioni in cui è necessario lavorare con matrici, tabelle, ecc.

Definizione di un array bidimensionale

- Per definire un array bidimensionale, è necessario specificare:
 - il nome della variabile array
 - il numero degli elementi presenti nelle due dimensioni (cardinalità di riga e cardinalità di colonna dell'array)

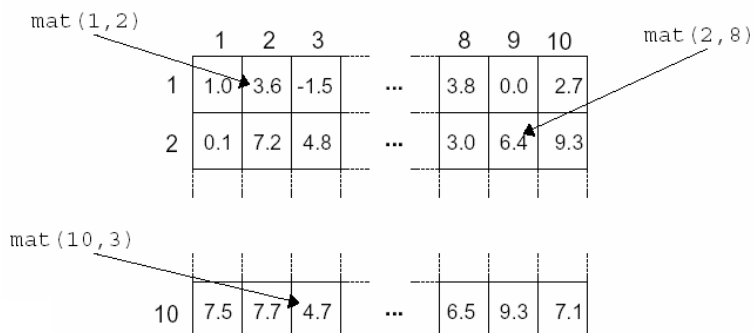
Esempio

- Inizializzazione di una variabile array mat contenente 10x10 elementi double:

```
mat = zeros(10,10);
```

- Che differenza c'è rispetto ad un array monodimensionale di 100 elementi?

Organizzazione di un array bidimensionale



Accesso agli elementi dell'array

- Per accedere ai singoli elementi di un array bidimensionale, è necessario specificare il nome della variabile array e gli indici di riga e di colonna che individuano l'elemento desiderato.
- Esempi:
mat(2,1)=3
fprintf('il valore è: %g\n', mat(2,7))
i=3
j=5
x=mat(i,j)*4+6

Lettura e stampa degli elementi di un array bidimensionale

- Per inizializzare da input una variabile array, è necessario realizzare un'operazione di input per ciascuno degli elementi
- Analogamente, per stampare il contenuto di un array, è necessario fare la stampa di ognuno degli elementi.
- Qual è il costrutto da utilizzare ?
- Esempio:
 - leggere da input le dimensioni e gli elementi di un array bidimensionale e stampare il risultato della lettura

```
function main
```

```
% si realizza la lettura da input di un array bidimensionale
```

```
% le dimensioni dell'array sono fornite dall'utente
```

```
% variabili utilizzate
```

```
mat=zeros(10,10); % array da leggere; dimensione iniziale di 10x10 elementi
```

```
i=0; % indice per scorrere le righe dell'array
```

```
j=0; % indice per scorrere le colonne dell'array
```

```
rig=0; % numero delle righe dell'array fornito in input
```

```
col=0; % numero delle colonne dell'array fornito in input
```

```
% input dimensioni
```

```
rig=input('Numero righe: ');
```

```
col=input('Numero colonne: ');
```

```
% dimensionamento array
```

```
mat=zeros(rig,col);
```

```
% ciclo di lettura
```

```
for i=1:rig
```

```
    % lettura i-ma riga
```

```
    for j=1:col
```

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

```
        mat(i,j)=input("");
```

```
    end
```

```
end
```

```
% stampa dell'array letto
```

```
fprintf('\nArray letto:\n');
```

```
for i=1:rig
```

```
    % stampa i-ma riga
```

```
    for j=1:col
```

```
        fprintf(' %g',mat(i,j));
```

```
    end
```

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

```
end
```

Esempio: calcolo della media per righe

- Leggere da input la dimensione e gli elementi di un array bidimensionale; fornire in uscita le medie degli elementi contenuti nelle righe dell'array.

Soluzione

Esempio: ricerca del massimo e del minimo per colonne

- Leggere da input la dimensione e gli elementi di un array bidimensionale; per ogni colonna, fornire in uscita valori e posizioni dell'elemento di valore massimo e dell'elemento di valore minimo.

Soluzione

```
function main
```

```
% si realizza la lettura da input di un array bidimensionale e si calcolano  
% le medie degli elementi contenuti sulle righe.
```

```
% variabili utilizzate
```

```
mat=zeros(10,10); % array da leggere; dimensione iniziale di 10x10 elementi  
medie=zeros(10,1); % array contenente le medie; dimensione iniziale di 10 elementi  
somma=0; % variabile contenente la somma corrente  
i=0; % indice per scorrere le righe dell'array  
j=0; % indice per scorrere le colonne dell'array  
rig=0; % numero delle righe dell'array fornito in input  
col=0; % numero delle colonne dell'array fornito in input
```

```
% input dimensioni
```

```
rig=input('Numero righe: ');  
col=input('Numero colonne: ');
```

```
% dimensionamento array
```

```
mat=zeros(rig,col);  
medie=zeros(rig,1);
```

```
% ciclo di lettura
```

```
for i=1:rig % lettura i-ma riga  
    for j=1:col  
        fprintf('Valore (%d,%d): ',i,j);  
        mat(i,j)=input("");  
    end  
end
```

```
% calcolo delle medie
```

```
for i=1:rig  
    % calcolo della media dell'i-ma riga  
    somma=0;  
    for j=1:col  
        somma=somma+mat(i,j);  
    end  
    media(i)=somma/col;  
end
```

```
% stampa dell'array letto
```

```
fprintf('\nArray letto:\n');  
for i=1:rig  
    % stampa i-ma riga  
    for j=1:col  
        fprintf(' %g',mat(i,j));  
    end  
    fprintf('\n');  
end
```

```
% stampa dell'array delle medie
```

```
fprintf('\nValori medi:\n');  
for i=1:rig  
    % stampa media dell'i-ma riga  
    fprintf('Media della riga %d: %g\n',i, media(i));  
end
```

```
function main
```

```
% lettura di un array bidimensionale e ricerca del massimo e del minimo per  
% colonne. Il programma fornisce in uscita valori e posizioni del massimo e del  
% minimo
```

```
% variabili utilizzate
```

```
mat=zeros(10,10); % array da leggere; dimensione iniziale di 10x10 el.  
max=zeros(10,1); % array contenente i max; dimensione iniziale di 10 el.  
posmax=zeros(10,1); % array contenente le posizioni dei max; dim. in. di 10 el.  
min=zeros(10,1); % array contenente i min; dimensione iniziale di 10 el.  
posmin=zeros(10,1); % array contenente le posizioni dei min; dim. in. di 10 el.  
i=0; % indice per scorrere le righe dell'array  
j=0; % indice per scorrere le colonne dell'array  
rig=0; % numero delle righe dell'array fornito in input  
col=0; % numero delle colonne dell'array fornito in input
```

```
% input dimensioni
```

```
rig=input('Numero righe: ');  
col=input('Numero colonne: ');
```

```
% dimensionamento array
```

```
mat=zeros(rig,col);  
max=zeros(col,1);  
posmax=zeros(col,1);  
min=zeros(col,1);  
posmin=zeros(col,1);
```

```
% ciclo di lettura
```

```
for i=1:rig  
    % lettura i-ma riga  
    for j=1:col  
        fprintf('Valore (%d,%d): ',i,j);  
        mat(i,j)=input("");  
    end  
end
```



```
% ricerca dei min e dei max
for j=1:col
    % ricerca del min e del max della j-ma colonna
    max(j)=mat(1,j);
    posmax(j)=1;
    min(j)=mat(1,j);
    posmin(j)=1;
```

```
for i=1:rig
    if(mat(i,j)>max(j))
        max(j)=mat(i,j);
        posmax(j)=i;
    elseif(mat(i,j)<min(j))
        min(j)=mat(i,j);
        posmin(j)=i;
    end % end if
end % end ciclo sulle righe
end % end ciclo sulle colonne
```

```
% stampa dell'array letto
fprintf('\nArray letto:\n');
for i=1:rig
    % stampa i-ma riga
    for j=1:col
        fprintf(' %g',mat(i,j));
    end
    fprintf('\n');
end
```

```
% stampa degli array dei max e dei min
fprintf('\nValori max e min trovati:\n');
for j=1:col
    % stampa risultati relativi alla j-ma colonna
    fprintf('\nColonna %d - max: %g in posizione %d\n',j,max(j),posmax(j));
    fprintf('Colonna %d - min: %g in posizione %d\n',j,min(j),posmin(j));
end
```