

LEZIONE 2

- Interazione con utente;
- Controllo del flusso
- Matrici
- Algebra matriciale

www.stoianov.it matlab@stoianov.it

Interazione coll'utente

- `x = input('invito');`
chiede con un certo 'invito' nel *command* l'inserimento di un valore, che viene assegnato alla variabile `x`.
- `disp(x);`
visualizza nel *command* il valore di una variabile
- `fprintf('formato', var1, var2, ...);`
visualizzazione formattata nel *command*

```
fprintf('R=%.2f S=%.2f \n', R, S);
fprintf('%d forme di tipo %s con radius %.2f \n', N, T, R);
```

% <type>
d integer
f float
c char
s string

Esercizio: Interfaccia

Compito: Calcolare la superficie totale di 2 dischi, di radius richiesto dal utente.

Utilizzare: `R=input('Radius of disk1 ?');`
`fprintf('Surface=: %.1f' \n',S);`

```
R = input('Radius of disk1 ?');  
S = discsurf (R);  
R = input('Radius of disk2 ?');  
S = S + discsurf (R);  
fprintf('Total surface: %.1f \n',S);
```

CONSOLA
>>due_dischi

File
due_dischi.m

matlab@stoianov.it

22

Controllo del flusso

I problemi non triviali richiedono deviazione dalla lineare struttura (semplice sequenza) di elaborazioni.

Ci sono due principali tipi di elaborazioni non-lineari:

- **ESECUZIONE CONDIZIONALE**
 - **IF** (*condizione*) **THEN** ... **ELSE** ...
 - **SWITCH** (*selettore*) [**CASE** ...]*
- **ESECUZIONE ITERATIVA**
 - **FOR** (*iteratore*) **DO** ...;
 - **WHILE** (*condizione*) **DO** ...;
 - **DO** ... **WHILE** (*condizione*)
- **EMERGENZE** (*non error-handling*)
 - **GOTO** ... - **BREAK** - **RETURN**

matlab@stoianov.it

23

Controllo del flusso in Matlab

CONDIZIONI

```
if <condizione>
...
elseif <condizione>
...
else
...
end
```

```
switch <variabile>
case <valore1>
...
case <valore2>
...
otherwise
...
end
```

ITERAZIONI

```
for i = <beg>:<end>
...
end
```

```
while <condizione>
...
end
```

EMERGENZE

```
break uscita anticipata da un ciclo
continue va alla successiva iterazione in un ciclo
return uscita anticipata da una funzione.
```

matlab@stoianov.it

24

Esempio: Flusso, IF

Compito: 1) Implementare una funzione **isdiff2** che verifica se la differenza tra due numeri è pari a 2
2) Verificare il suo funzionamento nella console.

La condizione: un espressione con risultato logico (booleano) del tipo:

`(n == 2)` % ha valore "VERO" se n è uguale a 2, altrimenti ha valore "FALSO"

`(n > 2)` % ha valore "VERO" se n è maggiore a 2, altrimenti ha il valore "FALSO"

Utilizzo: `if (n>2), disp('BIG'); else disp('SMALL'); end`

```
function r = isdiff2(i,j)
if i==(j-2),
    r=1;
elseif i==(j+2),
    r=1;
else
    r=0;
end
```

File
isdiff2.m

```
CONSOLA
>> isdiff2(3, 5)
ans
    1
>> isdiff2(3, 4)
ans
    0
```

matlab@stoianov.it

25

Esempio: Flusso, FOR

Compito: verificare quali dei numeri da 1 a 10 hanno distanza di 2 rispetto il numero 5 (ans: 3 e 7)

Ci ricordiamo la sintassi del operatore **for**

```
for i=1:5, disp(i); end; % Stampa i numeri da 1 a 5
```

```
for i=1:10
    disp(i);
    isdiff2(i,5)
end
```

File
try10.m

CONSOLA
>> try10;

```
for i=1:10
    if isdiff2(i,5),
        fprintf('%d ',i);
    end
end
```

26

Esempio: Flusso, FOR in FOR

Compito: riportare tutte le copie di numeri da 1 a 10 che hanno distanza 2 uno rispetto l'altro (ans: 1 e 3; 2 e 4, ecc.)

Hint: Utilizzare due cicli for, il secondo "annidato" nel primo.

```
for i=1:10
    for j=1:10
        if isdiff2(i,j)
            fprintf('%d - %d\n',i,j);
        end
    end
end
```

File
try1010.m

CONSOLA
>> try1010;

matlab@stoianov.it

27

Esempio: Flusso, CASE

Compito: 1) Implementare una funzione `diff2str` che riporta:
'UNO' se la differenza tra due numeri è 1 e
'DUE' se la differenza tra i numeri è 2.
2) Verificare il suo funzionamento nella console.

Strumenti: (a) `abs(x)` una funzione che dà $|x|$: il *valore assoluto* di x
(b) `round(x)` una funzione che dà il *valore intero* più vicino a x

```
function r = diff2str(i,j)
switch (abs(i-j))
    case 0, r='ZERO';
    case 1, r='UNO';
    case 2, r='DUE';
    otherwise, r='TANTO';
end
```

File
chediff.m

```
CONSOLA
>> diff2str(3, 5)
ans
    'DUE'
```

E' corretto il funzionamento del programma su tutti i dati, ad esempio, (4,4) ?

matlab@stoianov.it

28

Esempio: Flusso, while

Compito: produrre una sequenza di lettere 'x' di lunghezza casuale

Strumento: (1) `rand` una funzione che da un *numero casuale tra 0 a 1*

```
GoOn=1;           % Valore iniziale della condizione
while GoOn       % Continuare finche la condizione è true
    fprintf('x'); % Fa quello che deve fare (scrive "x")
    GoOn=(rand<0.8); % Andare avanti con 80% di probabil.
end
fprintf('\n');    % Mandare il cursore alla prossima riga
```

Fondamentale di scrivere commenti per poter leggere la logica del programma

matlab@stoianov.it

29

Esercizio: Flusso, IF / CASE

Compito: Calcolare la superficie totale di N elementi grafici, di tipo disco o quadrato, i grandezza ed il tipo di quali sono richiesti dal utente.

- | | | |
|--|--|--|
| 1) Implementare una funzione che calcola la superficie secondo il tipo di elemento grafico | <pre>if type=='d' ... else ... end</pre> | <pre>switch <selector> case 'd', ...; case 's', ...; otherwise, ...; end</pre> |
| 2) Scrivere un m-script che: | | |
| (a) chiede quanti elementi (N) ci sono | | <pre>for i=1:N</pre> |
| (b) fa un ciclo da 1 a N (1:N) | | <pre>...</pre> |
| (c) chiede info per ciascun elemento | | <pre>end</pre> |
| (d) calcola sa supecie di questo elemento | | |
| (e) calcola la somma totale | | |

matlab@stoianov.it

30