

Programmazione

Algoritmo

- Percorso logico che conduce alla soluzione di un problema
- Caratteristiche:
 - **Ordinabilità**: è costituito da una sequenza ordinata di passi
 - **Eseguibilità**: ogni singolo passo deve essere praticamente possibile
 - **Assenza di ambiguità**: ogni operazione elementare deve essere nota e condivisa tra ideatore e esecutore dell'algoritmo
 - **Terminazione**: l'algoritmo deve concludersi in un numero finito di passi

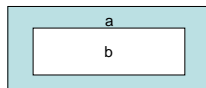
Scrittura di un algoritmo

- Non esiste un unico algoritmo che risolve un dato problema.
- **Obiettivo**: scrivere algoritmi efficienti e leggibili

Metodologia top-down

- Scomposizione di un problema generale in sotto-problemi più semplici
- I sotto-problemi corrispondono a parti funzionalmente indipendenti dello stesso algoritmo
- **NOTA**: la scomposizione può continuare a diversi livelli

Esempio pratico



- **Problema**: calcolo dell'area in azzurro
- **Soluzione** in macro-passi:
 - < Acquisisci i dati sui rettangoli a e b >
 - < Calcola area rettangolo "a" >
 - < Calcola area rettangolo "b" >
 - < Sottrai i due valori >

Procedure e Funzioni

- Per implementare l'algoritmo descritto, occorre far uso di **procedure e funzioni**
- **Procedura**: modulo di programma indipendente che può essere utilizzato da altri moduli per mezzo di una *chiamata*.
- **Funzione**: differisce dalla procedura in quanto prevede il passaggio di *parametri in input* e la restituzione di un *output*. La chiamata di una funzione viene effettuata nella parte destra di una istruzione di assegnamento.

Funzioni in MATLAB

Prima riga di un M-file che definisce una funzione:

```
function [variabili di output] = nome_funzione(variabili di input)
```

NOTE:

- "function" deve essere scritto in caratteri minuscoli
- variabili di output tra parentesi quadre
- variabili di input tra parentesi tonde e separati da una virgola
- "nome_funzione" deve essere uguale al nome del file.m in cui sarà salvata la funzione

Esempio: funzione che calcola l'area rettangolo

Funzione che dati in input base e altezza calcola e restituisce in output l'area del rettangolo:

```
function [area] = area_rett (base, altezza)

area = base * altezza;
```

Esempio: chiamata della funzione

```
val_area = area_rett(10, 8)
```

- Al momento della chiamata:
la funzione "area_rett" viene eseguita assegnando 10 alla variabile "base" e 8 alla variabile "altezza".
- Al termine dell'esecuzione della funzione:
Il valore di output "area" verrà assegnato alla variabile "val_area"

Variabili locali

- Tutte le variabili interne ad una funzione, sono **locali**:
 - sono utilizzabili esclusivamente all'interno della funzione.
 - vengono cancellate al termine della funzione
 - Non sono visibili al di fuori della funzione
 - Se la funzione viene chiamata più volte, le variabili della nuova chiamata non avranno alcuna relazione con quelle della chiamata precedente

Passaggio di parametri "per valore"

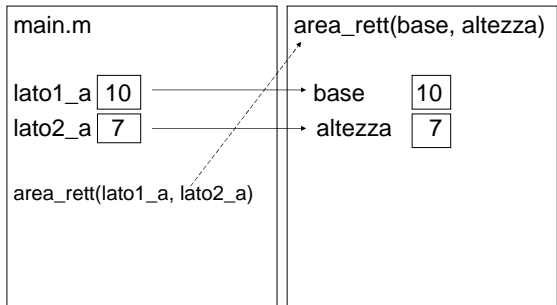
- Al momento della chiamata, il chiamante deve fornire il valore da associare ai parametri di input della funzione.
- Tale "passaggio di parametri" si dice **per valore**, in quanto comporta la copiatura del valore passato sui parametri interni

Nel nostro esempio:

Script "main.m":

```
lato1_a = 10;
lato2_a = 7;
lato1_b = 8;
lato2_b = 5;
area_a = area_rett (lato1_a, lato2_a);
area_b = area_rett (lato1_b, lato2_b);
differenza = area_a - area_b;
```

Graficamente:



Effetto

- All'atto della chiamata il parametro esterno "lato1_a" viene copiato nel parametro interno "base"
- Eventuali modifiche del parametro interno "base" non avranno alcuna ripercussione sul valore del parametro esterno "lato1_a"

Valori di output delle funzioni

- Le funzioni possono restituire uno o più output
- Esempio:
Funzione che dato in input il raggio restituisce l'area e la circonferenza del cerchio.

Scrivo in un file che chiamo `circle.m`:

```
function [a, c] = circle ( r )
a = pi * r .^ 2;
c = 2 * pi * r;
```
- La chiamata sarà:

```
[area, circonferenza] = circle(9)
```
- **Effetto:** il valore di "a" verrà assegnato a "area" e il valore di "c" verrà assegnato a "circonferenza"

In laboratorio

- Implementare lo script "main.m" e la funzione "area_rett"
- E rispondere alle seguenti domande:
 - Bloccare l'esecuzione all'interno della funzione. Quanto vale la variabile "base"?
 - Bloccare l'esecuzione nel main, dopo la prima chiamata. Quanto vale la variabile "base"?
 - Cambiare i nomi ai parametri interni "base" e "altezza". Cambia il funzionamento?

Funzioni interne

Esponenziali
Logaritmiche
Complesse
Numeriche
Trigonometriche
Iperboliche
Input/Output

Esponenziali e Logaritmiche

`sqrt(x)` restituisce la radice quadrata di x
Se l'argomento è un numero negativo restituisce la soluzione immaginaria positiva
Es: `log(-9)` restituisce $0 + 3.0000i$
`exp(x)` restituisce e^x
`log(x)` restituisce il logaritmo naturale di x
`log10(x)` restituisce il logaritmo decimale di x

Funzioni numeriche

<code>round(x)</code>	restituisce l'intero più vicino
<code>fix(x)</code>	restituisce x troncando la parte decimale
<code>ceil(x)</code>	restituisce x arrotondato per eccesso
<code>floor(x)</code>	restituisce x arrotondato per difetto
<code>abs(x)</code>	restituisce il valore assoluto

Esempi:

	<code>s = 1.1;</code>	<code>r=1.6</code>	<code>z=-1.1</code>	<code>g=-1.6</code>
<code>round</code>	restituisce <code>s=1;</code>	<code>r= 2;</code>	<code>z=-1;</code>	<code>g= -2</code>
<code>fix</code>	restituisce <code>s=1;</code>	<code>r=1;</code>	<code>z= -1;</code>	<code>g= -1</code>
<code>ceil</code>	restituisce <code>s=2;</code>	<code>r= 2;</code>	<code>z=-1;</code>	<code>g= -1</code>
<code>floor</code>	restituisce <code>s=1;</code>	<code>r=1;</code>	<code>z=-2;</code>	<code>g= -2</code>
<code>abs</code>	restituisce <code>s=1.1;</code>	<code>r=1.6;</code>	<code>z=1.1;</code>	<code>g=1.6</code>

Funzioni trigonometriche

NOTA 1: Le funzioni trigonometriche operano in modalità radianti.

Es. `cos(5)` restituisce il coseno dell'angolo di 5 radianti non di 5 gradi

NOTA 2: l'argomento va scritto all'interno di una coppia di parentesi tonde.

Es. `cos2` è scorretto, `cos(2)` è corretto

NOTA 3: \sin^2x va scritto come `(sin(x))^2`

Funzione di Input

- `input_num = input('inserisci un numero: ')`
- `input_parola = input('inserisci una parola','s')`
- NOTE:
 - l'input viene assegnato ad una variabile
 - Se l'input è una stringa, bisogna aggiungere il parametro 's' alla chiamata

Funzione di Output

- `disp('Area del rettangolo: ')`
visualizza la stringa di caratteri tra virgolette
- `disp(x)`
visualizza il contenuto della variabile x
- NOTA: la funzione "disp" non ha output!!!

Funzione di Output

- Per visualizzare testo e variabili insieme?
- E' necessario convertire tutto l'output in un'unica stringa
- `disp(['Area del rettangolo: ', num2str(x)])`
 - La funzione `num2str` serve a convertire un numero in una stringa
 - Le parentesi [] servono per concatenare 2 o più stringhe

In laboratorio

- Correggere lo script `main.m` in modo tale che:
 - I lati del rettangolo vengano chiesti in input all'utente mediante la funzione "input"
 - L'area della regione azzurra venga visualizzata mediante la funzione "disp", opportunamente introdotta da una frase esplicativa

Esercizio: Conversione Euro / Lire

- Scrivere un programma che
 - richieda in input un valore in Euro,
 - lo converta in lire italiane tramite un'apposita funzione di conversione
 - visualizzi nel main il valore in lire ottenuto

Esercizio: Conversioni di temperature

- Scrivere un programma che chieda in input un valore in gradi centigradi, lo converta in gradi Fahrenheit tramite un'apposita funzione e visualizzi il valore ottenuto
- Analogamente, scrivere un programma che dato un valore in gradi centigradi, lo converta in Kelvin

Esercizio: Differenza di Tempi

- Scrivere un programma
 - che chieda in input due tempi espressi in ore, minuti e secondi,
 - ne calcoli la differenza, esprimendola in secondi tramite un'opportuna funzione
 - Ne visualizzi la differenza

Esercizio: Scambio di variabili

- Scrivere un programma che:
 - Chieda in input due valori a e b
 - Chiami una funzione che ne scambia i valori
 - Visualizzarli.