LABORATORIO DI PROGRAMMAZIONE
Corso di laurea in matematica

8 – STRUTTURE DI CONTROLLO (1)

Marco Lapegna
Dipartimento di Matematica e Applicazioni
Universita' degli Studi di Napoli Federico II

wpage.unina.it/lapegna

Dalla precedente lezione

Il pascal-like e' uno pseudo linguaggio per la descrizione degli algoritmi

Le variabili sono nomi dati alle locazioni di memoria, i cui valori possono essere modificati durante l'algoritmo attraverso il loro nome

- Le variabili vanno **dichiarate**
- Le variabili vanno **definite** prima di essere usate

- Operatore di **assegnazione** "="
- Istruzione read per **leggere** dall'unita' di input
- Istruzione print per **stampare** sull'unita' di output

```
begin trapezio
var   Bmag, Bmin  : real
var   H,  Area : real
read  Bmag,  Bmin
read  H
Area = Bmag + Bmin
Area = Area * H
Area = Area /2
print  Area
end trapezio
```

L'algoritmo per il calcolo dell'area del trapezio

Un esempio

Scambio di due variabili

- Dati di input: A e B (ad es. 5 e 3)
- Dati di output: le stesse variabili A e B con i valori invertiti (3 e 5)

La coppia di istruzioni

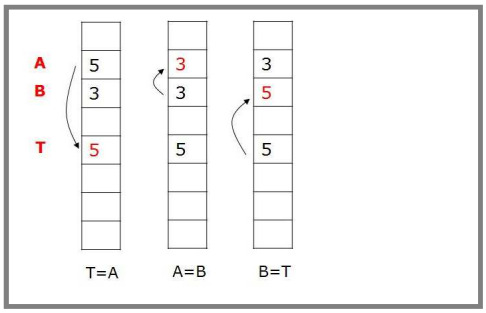
A = B
B = A

Non funziona !!
(si perde subito il valore di A)

necessaria una terza variabile (ad es. T) per salvare il valore di A ed assegnarlo successivamente a B

```
begin scambio
var  A,  B,  T: integer
read A, B
T = A
A = B
B = T
print A, B
end scambio
```

Algoritmo per lo scambio di due variabili



Evoluzione delle variabili A, B e T in memoria

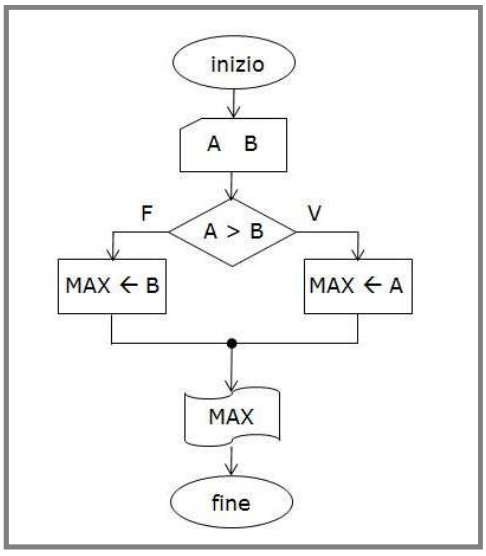
Struttura di selezione

Spesso in un algoritmo occore eseguire operazioni diverse a secondo del valore (vero o falso) di una condizione logica

Esempio:
Il massimo tra due numeri

Dati di **input**: i due numeri (A e B)
Dati di **output**: il massimo (MAX)

Idea: si confrontano i due numeri (A e B). Se il primo e' maggiore del secondo il massimo (MAX) e' A, altrimenti e' B



Flow chart del massimo di due numeri

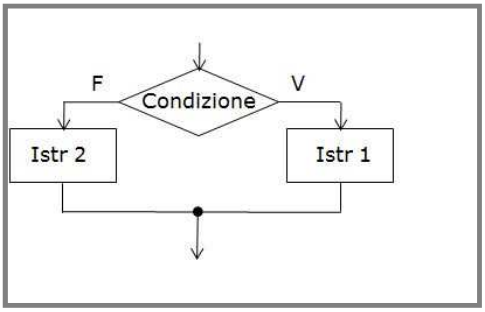
Struttura di selezione

In Pascal like la struttura di selezione e' realizzata con la struttura

" if-then-else-endif "

In generale

- Viene valutata **l'espressione logica (if)**
- Se e' vera viene eseguito il blocco di istruzioni **prima di "else" (Istr 1)**
- se e' falsa viene eseguito il blocco di istruzioni **dopo di "else" (Istr 2)**
- L'istruzione **"endif"** segnala la fine della struttura



Flow chart della struttura di selezione

```
....
if (condizione) then
    Istr 1
else
    Istr 2
endif
....
```

Struttura di selezione in pascal-like

Algoritmo del massimo in pascal like

L'algoritmo del massimo
in pascal like e' allora

- Se (A > B) si assegna il valore di A alla variabile MAX
- Altrimenti si assegna il valore di B alla variabile MAX
- Dopo l'istruzione **"endif"** l'esecuzione dell'algoritmo riprende regolarmente

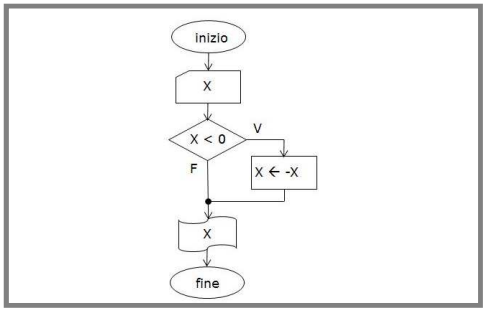
```
begin massimo
var A, B, MAX: real
read A, B
if (A > B) then
    MAX = A
else
    MAX = B
endif
print MAX
end massimo
```

Algoritmo del massimo in pascal like

Struttura di selezione semplificata

in alcuni casi e' necessario eseguire istruzioni solo se la condizione e' vera, ma **nessuna istruzione se la condizione e' falsa**

In questo caso manca il comando "else" e le relative istruzioni



il valore assoluto in flow chart

Esempio:
Valore assoluto di un numero

Dati di **input**: il numero X

Dati di **output**: lo stesso numero X, eventualmente cambiato di segno **se e' negativo**

```
begin valass
var X: real
read X
if (X < 0) then
    X = -X
endif
print X
end valass
```
il valore assoluto in pascal like

operatori relazionali e logici

Nella valutazione dell'espressione logica e' possibile verificare la veridicita' di **una o piu' condizioni basate su operatori relazionali**

> (maggiore) < (minore) == (uguale) /= (diverso)

combinate tra loro mediante **operatori logici**

AND OR NOT

Esempio:
(A > B) AND (X == 0)

Osservazioni
- Il risultato dell'espressione logica deve essere sempre **vero** o **falso**
- gli operatori relazionali vengono valutati prima degli operatori logici

Struttura di iterazione

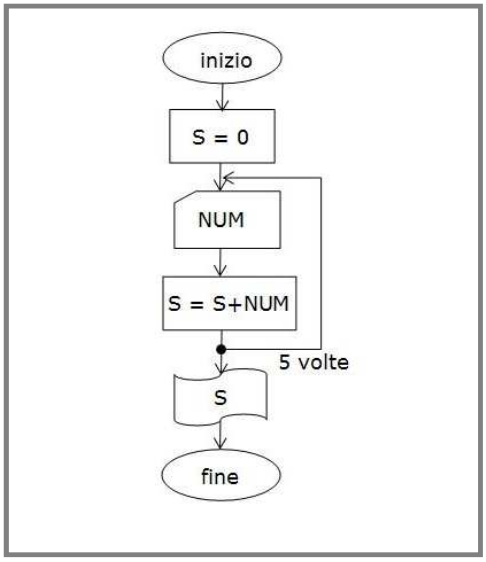
Spesso in un algoritmo e' necessario **ripetere piu' volte lo stesso blocco di istruzioni**

Esempio:
Somma di 5 numeri

Dati di **input**: i 5 numeri

Dati di **output**: la somma

Idea: leggere un numero alla volta (NUM) e sommarlo ad una variabile inizialmente posta uguale a zero (S)



Flow chart dell'algoritmo per la somma di 5 numeri

Struttura di iterazione in pascal like

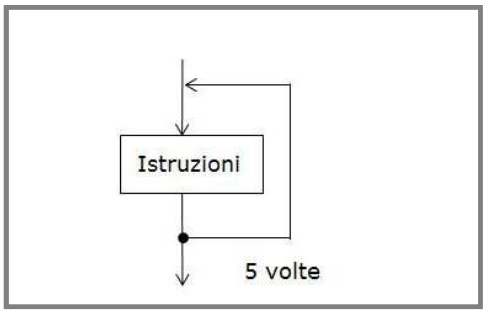
In Pascal like la struttura di iterazione e' realizzata con la struttura

" for - endfor "

In generale

- Vengono eseguite le istruzioni tra **"for"** e **"endfor"** per tutti i valori dell'indice **I** compresi tra il primo e l'ultimo valore

- L'indice I e' una variabile a tutti gli effetti e va dichiarata



Flow chart della struttura di iterazione "for-endfor"

```
....
for  I = 1 to 5
    Istruzioni
endfor
....
```

La struttura di iterazione "for-endfor" in pascal like

Algoritmo della somma in pascal like

L'algoritmo della somma
in pascal like e' allora

- La coppia di istruzioni
  read NUM
  S = S + NUM
  Viene ripetuta 5 volte con valori dell'indice
  I = 1, 2, 3, 4, 5

- Dopo l'istruzione **"endfor"** l'esecuzione dell'algoritmo riprende regolarmente

```
begin somma
var S, NUM: real
var I : integer
S = 0
for I = 1 to 5
    read NUM
    S = S + NUM
endfor
print S
end somma
```

Algoritmo della somma in pascal like

Esempio di esecuzione

Dati di input: 5; 6; 1; 8; -1

Primo passo (I=1)
NUM = 5 S = S+NUM (S = 5)

Secondo passo (I=2)
NUM = 6 S = S+NUM (S = 11)

Terzo passo (I=3)
NUM = 1 S = S+NUM (S = 12)

Quarto passo (I=4)
NUM = 8 S = S+NUM (S = 20)

Quinto passo (I=5)
NUM = -1 S = S+NUM (S = 19)

Risultato: 19 (ultimo valore di S)

I	S			NUM			
1	5			5			Primo passo
2	11			6			Secondo passo
3	12			1			Terzo passo
4	20			8			Quarto passo
5	19			-1			Quinto passo

Evoluzione dei dati in memoria nel corso dell'algoritmo della somma

In generale

La struttura di iterazione "for – endfor" esegue il blocco di istruzioni per tutti i valori dell'indice I compresi tra le variabili intere Start e End

Start: primo valore della variabile I

End: ultimo valore della variabile I

E' possibile specificare un valore opzionale anche negativo (**Passo**) che rappresenta l'incremento della variabile I (se manca, Passo=1)

Esempio: Start=11, End=5, Step=-2

for I = Start to End step Passo

produce la sequenza di valori per l'indice I:
11, 9, 7, 5

```
begin algoritmo
var I, Start, End, Passo: integer
...
for I = Start to End step Passo
    istruzioni
endfor
...
end algoritmo
```

Sintassi generale della
struttura di iterazione "for-endfor"_

Algoritmo della somma (II versione)

E' possibile avere una versione piu' generale dell'algoritmo

Somma di N numeri

In questo caso prima della struttura "for-endfor" si definisce la variabile N, che si usa come valore finale dell'indice I

si ottiene un algoritmo valido per sommare qualunque insieme di numeri

```
begin somma
var S, NUM: real
var I , N : integer
read N
S = 0
for I = 1 to N
    read NUM
    S = S + NUM
endfor
print S
end somma
```

Algoritmo della somma di N numeri in pascal like

Strutture innestate

Le strutture di controllo possono essere innestate l'una nell'altra

- La prima struttura che si apre deve essere l'ultima che si chiude
- In caso di piu' strutture "for-endfor" e' necessario utilizzare indici diversi
- Spostare verso destra le istruzioni interne ad una struttura aiuta la leggibilita' dell'algoritmo

```
begin  algoritmo
var I, J, K : integer
...
for  I  =  1 to N
    for J = 1 to M
        for K = 1 to L
            if  (condizione ) then
                istruzioni
            endif
        endfor
    endfor
endfor
...
end  algoritmo
```

Esempio di strutture di controllo innestate

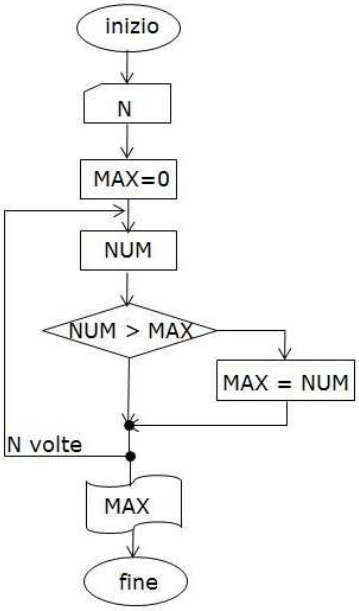
Esempio: il massimo di N numeri (positivi)

Dati di input:
N = 5 (la quantita' di numeri da esaminare)
NUM = 5; 6; 1; 8; 2

Risultato: Il massimo (MAX=8)

Idea: leggere un numero alla volta (NUM) e confrontarlo con la variabile che contiene il massimo dei numeri esaminati (MAX). Se NUM e' maggiore di MAX si cambia il valore di MAX

2 strutture innestate"
- "for - endfor" per leggere i numeri uno alla volta
- "if-endif" per confrontare NUM e MAX



Flow-chart per la ricerca del massimo

Algoritmo del massimo in pascal-like

Dati di input: 5; 6; 1; 8; 2

Primo passo (I=1)
NUM = 5 NUM > MAX ? (si) → MAX=5

Secondo passo (I=2)
NUM = 6 NUM > MAX ? (si) → MAX=6

Terzo passo (I=3)
NUM = 1 NUM > MAX ? (no)

Quarto passo (I=4)
NUM = 8 NUM > MAX ? (si) → MAX=8

Quinto passo (I=5)
NUM = 2 NUM > MAX ? (no)

Risultato: 8 (ultimo valore di MAX)

```
begin massimo
var MAX, NUM: real
var I , N : integer
read N
MAX = 0
for I = 1 to N
    read NUM
    if (NUM> MAX) then
        MAX = NUM
    endif
endfor
print MAX
end massimo
```

Algoritmo della ricerca del massimo in pascal-like

Riassumendo

Le strutture di controllo determinano l'ordine con cui vengono eseguite le istruzioni

Struttura di selezione "if-then-else"

Esegue in alternativa due blocchi di istruzioni
a secondo della veridicita' di una condizione logica

Struttura di iterazione "for-endfor"

Esegue piu' volte un blocco di istruzioni
per tutti i valori di un indice compresi in un intervallo

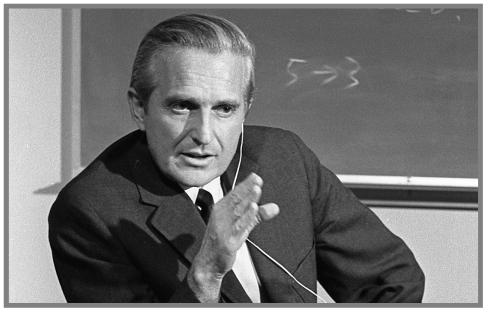
Le strutture di controllo possono essere innestate

Sono elementi universali degli algoritmi
presenti in tutti i linguaggi di programmazione

Un po' di storia (8)

Douglas Carl Engelbart (1925-2013)

- Ingegnere e inventore americano e' noto per i suoi studi sull'Interazione Uomo-Macchina. A tal fine fondo' l'istituto Augmentation Research Center (ARC) nel 1960 per sviluppare progetti e esperimenti con nuovi strumenti e tecniche.
- il principale prodotto dell'ARC fu l'oN-Line System (NLS), un sistema che gia' nel 1968, racchiudeva tutte le caratteristiche di un moderno personal computer: mouse, interfaccia grafica a finestre, word processing, teleconferenza, connessione di rete.
- L'NLS fu oggetto di una storica presentazione nota come "Mother of All Demos" che influenzo' profondamente le tecnologie alla base di Microsoft ed Apple 10 anni piu' tardi.
- Guidato dall'idea che attraverso il computer era possibile costruire un mondo migliore, vinse il Turing Award nel 1997 per la sua visione sull'interazione Uomo-Macchina e per aver realizzato le tecnologie necessarie



Englebart negli anni '70 (credit: SRI International)

Il primo mouse (court. Computer History Museum)