



LABORATORIO DI PROGRAMMAZIONE
Corso di laurea in matematica

10 – STRUTTURE DI CONTROLLO REPEAT E WHILE

Marco Lapegna

Dipartimento di Matematica e Applicazioni

Universita' degli Studi di Napoli Federico II

wpage.unina.it/lapegna

Marco Lapegna –
Laboratorio di Programmazione
10. Strutture repeat e while

Dalla precedente lezione

Strutture di controllo e variabili strutturate

Struttura di selezione **"if-then-else-endif"**

Struttura di iterazione **"for-endfor"**

Definiscono l'ordine con cui vengono eseguite le istruzioni di un algoritmo

Variabili strutturate (**array**)

Permettono l'accesso a dati organizzati in vettori o matrici attraverso un indice

Sono concetti universali presenti in ogni linguaggio di programmazione

Dato un array di nomi (un elenco) di lunghezza N determinare la posizione nell'array di un nome da cercare

Dati di input:

- N (lunghezza dell'elenco)
- Elenco(1),...,Elenco(N) (l'elenco)
- NOME (nome da cercare)

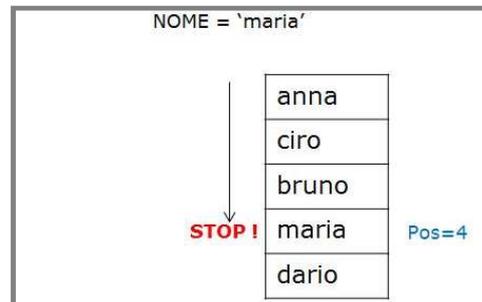
Dati di output

- Pos (la posizione di NOME in Elenco)

IDEA

Esaminare uno alla volta gli elementi di Elenco, e confrontarli con il NOME

RICERCA SEQUENZIALE



Ricerca di 'maria' nell'array Elenco

Ci si deve arrestare quando:

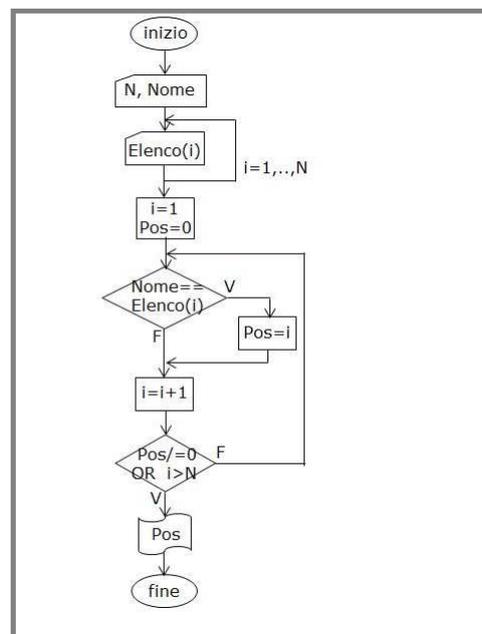
- Si trova il NOME nell'Elenco oppure
- Si scorre tutto l'Elenco ma NOME non è presente (si assume Pos=0)

Esempio1: N = 5; NOME = 'maria';
Elenco=('anna', 'ciro', 'bruno', 'maria', 'dario')

Pos = 4 (i=4 iterazioni)

Esempio2: N = 5; NOME = giacomo;
Elenco=('anna', 'ciro', 'bruno', 'maria', 'dario')

Pos = 0 (i=5 iterazioni)



Flow chart della ricerca sequenziale

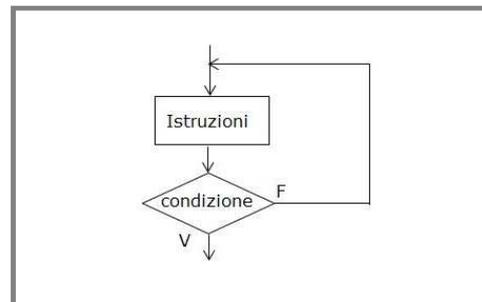
La struttura "repeat-until" in pascal like

In pascal-like una tale struttura di iterazione e' realizzata con la struttura

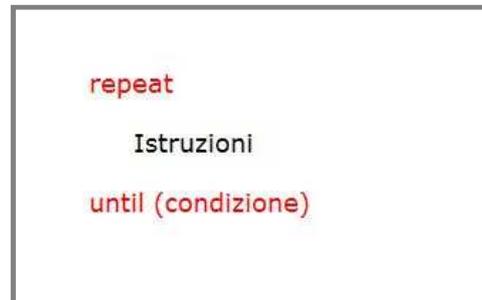
"repeat-until "

In generale

- Vengono eseguite ripetutamente le **Istruzioni** comprese tra **"repeat"** e **"until"**
- Ci si arresta quando la **condizione** posta **alla fine della struttura risulta vera**



Flow chart della struttura "repeat-until"



La struttura "repeat-until" in pascal-like

Ricerca sequenziale in pascal-like

Osservazione:

In generale la condizione di arresto puo' essere la **combinazione di piu' operazioni logiche** legate dagli operatori OR, AND e NOT

L'algoritmo per la ricerca sequenziale in pascal like e' allora

```
begin ricbin
var i, N, Pos : Integer
var Elenco(10), NOME : array of character
read N
Pos = 0
i = 1
repeat
    if ( Elenco(i) == NOME ) then
        Pos = i
    endif
    i = i+1
until ( Pos /= 0 OR i > N)
print Pos
end ricbin
```

Ricerca sequenziale in pascal-like

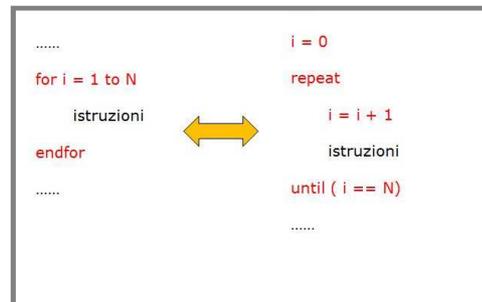
Osservazione:

In generale la condizione di arresto puo' essere la combinazione di piu' operazioni logiche legate dagli operatori OR, AND e NOT

La struttura “repeat-until” e' piu' flessibile e generale della struttura “for-endfor”

Una struttura “repeat-until” con una condizione di arresto solo sul numero di passi e' equivalente ad una struttura “for-endfor”

L'indice *i* va gestito esplicitamente



Equivalenza tra “for-endfor” e “repeat-until”

La struttura repeat-until e' utile per far si che un dato di input sia corretto

Esempio:

Leggere un array di lunghezza *N*, assicurandosi prima che $N > 0$

Idea :

Leggere ripetutamente *N* fino a che $N > 0$

```
begin leggi  
  var A(100) array of float  
  var i, N :integer  
  repeat  
    read N  
  until ( N > 0)  
  for i = 1 to N  
    read A(i)  
  endfor  
end leggi
```

Letture con controllo di un dato di input

Esempio: calcolo del M.C.D.

M.C.D. tra A e B (non entrambi nulli)

Metodo delle divisioni successive (o di Euclide)

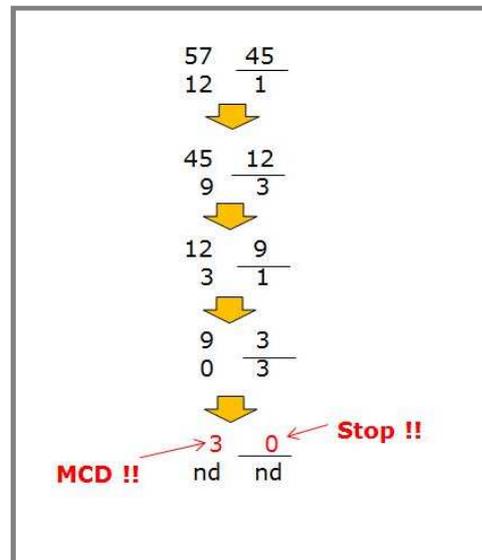
Si DIVIDE il MAGGIORE di essi per il MINORE;

Se il RESTO della divisione è ZERO, il M.C.D. è il numero MINORE (ovvero il divisore);

Se il RESTO della divisione è DIVERSO da ZERO si DIVIDE il numero MINORE per tale RESTO.

Si continua così fino ad ottenere una divisione per ZERO. Il M.C.D. cercato è il DIVIDENDO dell'ultima divisione.

Esempio: M.C.D.(57,45) = 3



Il metodo delle divisioni successive per il MCD

Un possibile algoritmo

E' necessario effettuare
la divisione (intera) tra A e B

Calcolare

- Il quoziente Q
- Il resto R

Rinominare A e B per la successiva divisione

Ripetere finche' B==0

PROBLEMA

Cosa accade se il dato di input B=0?

```
begin MCD
var A, B, Q, R, MCD : integer
read A, B
repeat
  Q = A/B
  R = A - B*Q
  A = B
  B = R
until ( B == 0)
MCD = A
print A
end MCD
```

Un possibile algoritmo in pascal like per il MCD tra A e B

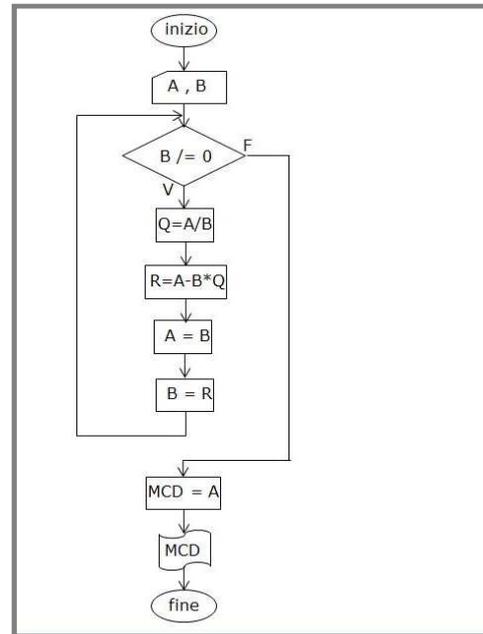
Se il minore dei due numeri (cioè B) è uguale a zero, non è possibile eseguire l'algoritmo

Ma il MCD con B=0 è definito !

Es. M.C.D. (5,0) = 5 (cioè A)

Soluzione:

Eeguire il test su B **PRIMA** della divisione



Flow chart per il MCD (seconda versione)

La struttura di iterazione

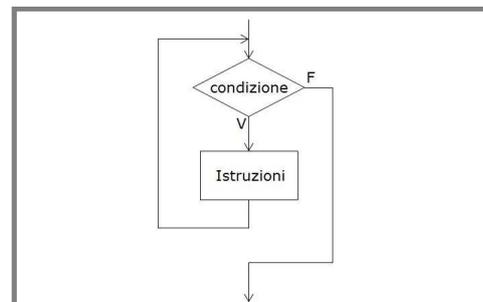
while-endwhile

Esegue ripetutamente le istruzioni comprese tra "while" e "endwhile", ma a differenza del repeat-until

PRIMA valuta la condizione

DOPO esegue le istruzioni

Le istruzioni vengono eseguite se la condizione risulta VERA



Flow chart della struttura while-endwhile

while (condizione)

Istruzioni

endwhile

La struttura while-endwhile in pascal like

Algoritmo del MCD in pascal like

L'algoritmo del MCD puo' allora essere riscritto utilizzando la struttura while-endwhile

In questo modo l'algoritmo puo' essere utilizzato anche quando il dato di input B=0

```
begin MCD2
var A, B, Q, R, MCD : integer
read A, B
while (B /= 0)
  Q = A/B
  R = A - B*Q
  A = B
  B = R
endwhile
MCD = A
print A
end MCD
```

Algoritmo del MCD in pascal like (seconda versione, con struttura while-endwhile)

Esecuzione dell'algoritmo

Esempio:

$A = 57$ $B = 45$

Dopo 4 iterazioni del ciclo while il MCD si trova nella variabile A

Esempio:

$A = 57$ $B = 0$

Il ciclo while non viene eseguito e il MCD si trova nella variabile A

	A	B	Q	R	memoria		
Prima del ciclo while	57	45					
Al termine del 1° passo	45	12	1	12			
Al termine del 2° passo	12	9	3	9			
Al termine del 3° passo	9	3	3	3			
Al termine del 4° passo	3	0	3	0			

Evoluzione dei dati in memoria nel corso dell'algoritmo

Quando utilizzare le tre strutture di iterazione

- for-endfor
- repeat-until
- while-endwhile

La scelta dipende da

- Numero di iterazioni noto a priori
- Necessita' di verificare la condizione prima o dopo le istruzioni da ripetere

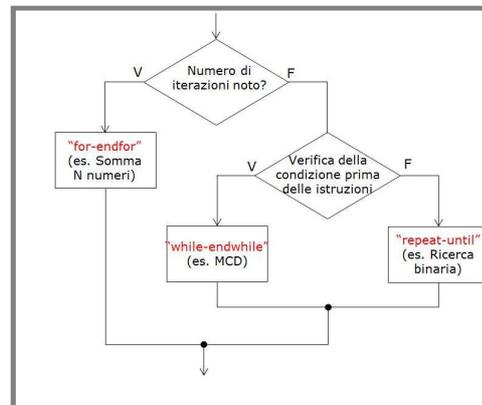


Diagramma di scelta della struttura di iterazione piu' opportuna

Problema: confronto tra stringhe di caratteri

Dati di input:

- N (intero)
- A e B (due array di caratteri di lunghezza N)

Dati di output:

- test (intero) che vale
 - test = 1 se i due array sono uguali
 - test = 0 se sono diversi

N.B. ogni componente contiene un solo carattere

Algoritmo di confronto tra stringhe

Idea:

Partendo dalla prima componente, confrontare coppie corrispondenti di componenti, fino a trovare una coppia diversa oppure fino alla fine degli array

Se dopo la serie di confronti anche le ultime due componenti sono uguali, allora le due stringhe sono uguali

```
begin confronta
var  (100), B(100) : array of float
var  i, N , test :integer

read N
for i = 1 to N
  read A(i), B(i)
endfor

test = 0
i = 0

repeat
  i = i+1
until ( A(i)≠B(i) or i >= N)

if ( A(i) == B(i) )
  test = 1
endif

print test
end confronta
```

Algoritmo in pascal like per il confronto tra stringhe di caratteri

riassumendo

Strutture di iterazione in pascal-like

- for-endfor
- repeat-until
- while-endwhile

Strutture universali disponibili in tutti i linguaggi di programmazione

Federico Faggin (1941)

- Italiano, laureato in fisica a Padova, lavora alla STMicroelettronica dove collabora alla definizione di tecnologie per circuiti integrati
- Nel 1970 si stabilisce negli Stati Uniti e lavora alla Intel dove dirige il progetto per lo sviluppo del 4004 primo esempio di moderna CPU (1971)
- L'Intel 4004 e' stato il primo microprocessore che integra in un solo chip ALU e CU, lavorava a 0.1 MHz e conteneva 2300 transistor. Apre le porte all'elettronica e all'informatica di massa
- Con la stessa tecnologia (MOS) dirige lo sviluppo di altre generazioni di processori e di memorie alla Intel.
- Fonda altre aziende di microprocessori: Zilog e Synaptic e riceve prestigiosi premi internazionali (Kioto Prize e National Medal of Technology)



F. Faggin nel 2009
(courtesy of Computer History Museum)



L'Intel 4004