

Strutture Ciclo

Consentono di realizzare *cicli di elaborazione*, ossia la *esecuzione ripetuta* di una sequenza di istruzioni:

... Esempi ...

calcolare le paghe dei dipendenti di una azienda

noto il numero N dei dipendenti, ripetere N volte la sequenza di “calcolo della paga”

superare un esame universitario

si ripete il “sostenere l’esame” fino a quando non si è promossi (voto \geq 18)

giocando a carte ...

Si continua a distribuire le carte fin quando ciascun giocatore ha 5 carte

Strutture Ciclo

si suddividono in:

STRUTTURE CICLICHE ITERATIVE

il numero di volte per il quale la sequenza viene ripetuta non è noto a priori, ma è condizionato dal verificarsi di un determinato evento durante l’esecuzione

STRUTTURE CICLICHE RIPETITIVE

il numero di volte per il quale viene ripetuta la esecuzione della sequenza è noto “a priori”

Strutture Ciclo

... *Esempi* ...

Iterative

superare un esame universitario

si ripete a "sostenere l'esame" fino a quando non si è promossi (voto \geq 18)

cercare una parola in un dizionario

continua a sfogliare le pagine del dizionario fin quando non trovi la
parola voluta

Ripetitive

calcolare le paghe dei dipendenti di una azienda

noto il numero N dei dipendenti, ripetere N volte la sequenza di "calcolo
della paga"

Strutture Ciclo

Strutture iterative

Due forme fondamentali:

Ciclo a condizione iniziale (while):

fin quando un'espressione condizionale è VERA, una
sequenza di istruzioni è eseguita più volte (il ciclo termina
quando l'espressione condizionale è FALSA)

Ciclo a condizione finale (repeat-until):

una sequenza di istruzioni è eseguita 1 o più volte finché
un'espressione condizionale diventa VERA (il ciclo
continua se l'espressione condizionale è FALSA)

La semantica può essere diversa in linguaggi diversi

Strutture Ciclo

Ciclo a condizione iniziale

while (<espressione>) S;

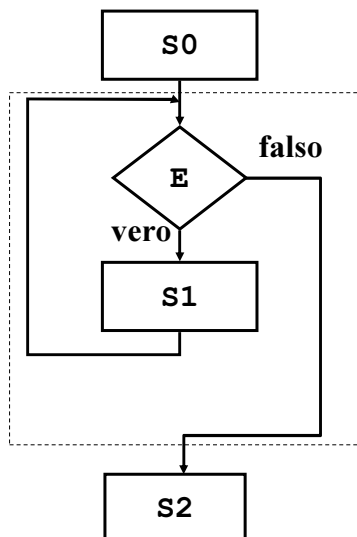
<espressione> è una espressione booleana;
S è una sequenza di istruzioni

semantica:

- mentre <espressione> è *vera* ripeti l'esecuzione della sequenza S (l'esecuzione o la riesecuzione della sequenza S verrà sospesa quando <espressione> risulterà falsa)
- la sequenza S può non essere eseguita affatto
- dal ciclo si esce con <espressione> *falsa*

Strutture Ciclo

Ciclo a condizione iniziale



```
S0;  
while (E)  
S1;  
S2;
```

possibili sequenze sono:

S0 S2
S0 S1 S2
S0 S1 S1 S2
S0 S1 S1 S1 S2
.....

la struttura è 1 in / 1 out

Esempio

```

.....
int x=3;
while (x<7)
begin
write("%d",x);
x=x+1;
end;
write("FINE");
.....

```

x	x<7	write("%d",x)
3	si	3
4	si	4
5	si	5
6	si	6
7	no	FINE

Elementi di Informatica
Prof. G. A. Di Lucca - Univ. del Sannio

7

Strutture Ciclo

Ciclo a condizione iniziale

... **Attenzione** ... **Errori Frequenti**
c'è il rischio che un errato uso della struttura porti verso *infinite ripetizioni*

Esempi:

```

...
while (X > 0)
A = B + C;
...

... poiché il valore di X non viene
in alcun caso modificato.....

```

```

...
while(i > n)
begin ...
i = i + 1;
end
...

... i varia, ma essendo il suo
valore iniziale già > n...

```

Elementi di Informatica
Prof. G. A. Di Lucca - Univ. del Sannio

8

Strutture Ciclo

Ciclo a condizione iniziale

... Assicurarsi della **Convergenza** del ciclo ...

- assicurarsi, in prima istanza, che nel corpo del while venga alterato il valore di almeno una delle informazioni coinvolte in <espressione>
- assicurarsi, in ogni caso, che in un numero finito di iterazioni, <espressione> **diventi falsa**
- è anche necessario che a tutte le informazioni usate nell'<espressione> sia stato assegnato un valore prima della struttura while ...

Esempio

Analisi e Specifica

Definizione del problema: data una sequenza di numeri interi, terminante con il numero 0, trovare quello con valore massimo

Definizione dei dati del problema:

I: un numero della sequenza

Pi: il numero 0 termina la sequenza

U: MAX, il numero con il valore massimo, di tipo intero

Pu: nessuna

Esempio

Tabella delle variabili di ingresso

Nome variabile	Descrizione	Tipo
Num	numero intero della sequenza	INT

Tabella delle variabili di uscita

Nome variabile	Descrizione	Tipo
MAX	Il numero con valore massimo	INT

Esempio

Progettazione

Descrizione del metodo di elaborazione:

- è letto il valore del primo numero della sequenza di interi;
- si considera il primo numero come quello massimo;
- se il numero immesso è diverso da zero si legge il numero successivo nella sequenza e lo si confronta con il massimo trovato finora e
- si continua a fare ciò (lettura di un nuovo numero e confronto con il massimo) fin quando non è immesso il valore zero;
- si stampa il valore massimo così trovato

Esempio - il programma LDP

```
Program Trova_massimo;  
// Trova MAX in sequenza  
int Num, MAX;  
begin  
  write( "immetti un numero");  
  read("%d" Num);  
  MAX=Num;  
  while (Num<>0)  
  begin  
    write ("immetti un numero ");  
    read("%d" Num);  
    if (Num>MAX) then MAX=Num;  
  end;  
  write ( "Il valore massimo è ", MAX );  
end.
```

Elementi di Informatica
Prof. G. A. Di Lucca - Univ. del Sannio

13

Esempio - il programma C

```
#include <stdio.h>  
main ()  
{  
  // Trova MAX in sequenza  
  int Num, MAX;  
  printf( "immetti un numero \n");  
  scanf("%d" , &Num);  
  MAX=Num;  
  while (Num!=0)  
  {  
    printf ("immetti un numero \n");  
    scanf("%d" , &Num);  
    if (Num>MAX) MAX=Num;  
  }  
  printf ( "Il valore massimo è %d \n", MAX );  
}
```

Attenzione !! Se sono immessi tutti numeri negativi risulterà che MAX sarà uguale a zero. Ma il valore zero che chiude la sequenza non fa parte della sequenza stessa e quindi non deve essere considerato. Una soluzione per ovviare a ciò è riportata nel lucido successivo

Elementi di Informatica
Prof. G. A. Di Lucca - Univ. del Sannio

14

Esempio - il programma C

```
#include <stdio.h>
main ()
{
// Trova MAX in sequenza
int Num, MAX;
printf( "immetti un numero \n");
scanf("%d" , &Num);
MAX=Num;
while (Num!=0)
{
if (Num>MAX) MAX=Num;
printf( "immetti un numero \n");
scanf("%d" , &Num);
}
printf ( "Il valore massimo è %d \n", MAX );
}
```

.... Basta effettuare il confronto tra Num e MAX prima della lettura di un nuovo valore per Num ...

Esempio - il programma C

```
#include <stdio.h>
main ()
{
// Trova MAX in sequenza
int Num, MAX;
printf( "immetti un numero \n");
scanf("%d" , &Num); Num=4
MAX=Num; Max=4
while (Num!=0)
{ if (Num>MAX) MAX=Num; Max=4 Max=82
printf( "immetti un numero \n"); Num=82 Num=34 Num=25 Num=0
scanf("%d" , &Num);
}
printf ( "Il valore massimo è %d \n", MAX );
```

```
}
printf ( "Il valore massimo è %d \n", MAX );
```

Strutture Ciclo

Ciclo a condizione finale

repeat S until (<espressione>);

<espressione> è una espressione booleana;

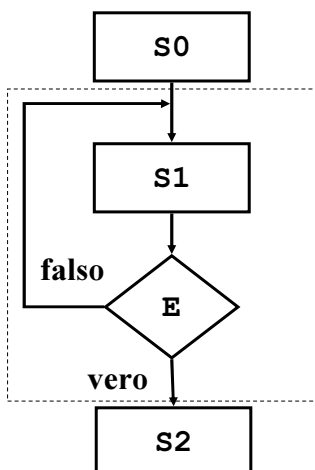
S è una sequenza di istruzioni

semantica:

- ripeti l'esecuzione di S finché <espressione> risulta vera; ovvero, dopo una esecuzione di S, la stessa verrà ripetuta se <espressione> è falsa
- la sequenza S viene eseguita almeno una volta
- dal ciclo si esce con <espressione> vera

Strutture Ciclo

Ciclo a condizione finale



```
S0;  
repeat  
  S1  
until (E);  
S2;
```

le possibili sequenze sono:

S0 S1 S2
S0 S1 S1 S2
S0 S1 S1 S1 S2
.....

la struttura è 1 in / 1 out

Esempio

```

.....
int x=3;
repeat
    write("%d",x);
    x=x+1;
until (x>=7)
write("FINE");
.....

```

x	X>=7	write("%d",x)
3		3
4	no	4
5	no	5
6	no	6
7	si	FINE

Elementi di Informatica
Prof. G. A. Di Lucca - Univ. del Sannio

19

Strutture Ciclo

Ciclo a condizione finale

... esiste anche un'altra forma ...

do S while (<espressione>);

<espressione> è una espressione booleana; S è una sequenza di istruzioni

semantica:

- ripeti l'esecuzione di S finché <espressione> risulta **falsa**; ovvero, dopo una esecuzione di S, la stessa verrà ripetuta se <espressione> è **vera**
- la sequenza S viene eseguita almeno una volta
- dal ciclo **si esce** con <espressione> **falsa**

20

Strutture Ciclo

Ciclo a condizione finale

```

    S0;
    do
        S1
    while (E);
    S2;
  
```

le possibili sequenze sono:
 S0 S1 S2
 S0 S1 S1 S2
 S0 S1 S1 S1 S2

la struttura è 1 in / 1 out

NB: nel linguaggio C per il ciclo a condizione finale esiste solo la struttura do-while

Elementi di Informatica
Prof. G. A. Di Lucca - Univ. del Sannio

21

Esempio

```

    .....
    int x=3;
    do
        write("%d", x);
        x=x+1;
    while(x<7)
    write("FINE");
    .....
  
```

x	x<7
3	
4	si
5	si
6	si
7	no

write("%d", x)

3

4

5

6

FINE

Elementi di Informatica
Prof. G. A. Di Lucca - Univ. del Sannio

22

Esempio

```

.....
int x=7;
do
    write("%d", x);
    x=x+1;
while (x<7)
write("FINE");
.....

```

x	x<7	write("%d\n", x)
7		7
7	no	FINE

Elementi di Informatica
Prof. G. A. Di Lucca - Univ. del Sannio

23

Strutture Ciclo

Ciclo a condizione finale

... Errori frequenti e convergenza del ciclo ...

- anche per il *do-while (repeat-until)* c'è il rischio che un errato uso della struttura porti verso *infinite ripetizioni*
- problemi di convergenza analoghi a quelli riscontrati per il costrutto *while ...* ma nel *do-while (repeat-until)* l'espressione condizionale viene valutata alla fine e non all'inizio del ciclo
 - assicurarsi, in prima istanza, che nel corpo del *do-while* venga alterato il valore di almeno una delle informazioni coinvolte in <espressione>
 - assicurarsi, in ogni caso, che in un numero finito di iterazioni <espressione> diventi *falsa*

24

Esempio

Analisi e Specifica

Definizione del problema: fornito in input un numero intero dire quante volte esso è presente in una sequenza di numeri interi, terminante con il numero 0

Definizione dei dati del problema:

I: un numero intero della sequenza; il numero intero da cercare

Pi: il numero zero termina la sequenza; il numero da cercare è diverso da zero

U: il numero di volte per cui il numero da cercare è nella sequenza

Pu: nessuna

Esempio

Tabella delle variabili di ingresso

Nome variabile	Descrizione	Tipo
Num	numero intero della sequenza	INT
Cerca	numero da cercare	INT

Tabella delle variabili di uscita

Nome variabile	Descrizione	Tipo
Volte	Il numero di presenze di Cerca nella sequenza	INT

Esempio

Progettazione

Descrizione del metodo di elaborazione:

- è letto il valore del numero da cercare nella sequenza;
- è letto un numero della sequenza di interi;
- si confronta il numero letto con quello da cercare
- se essi sono uguali si incrementa di una unità il contatore delle presenze del numero nella sequenza
- si continua dal secondo punto fino all'immissione dello zero
- si stampa il valore del contatore delle presenze del numero nella sequenza

Esempio - il programma C

```
#include <stdio.h>
main ()
{
    // trova e conta in sequenza
    int Num, Cerca, Volte;
    do
        { printf("immetti il numero da cercare \n");
          scanf("%d", &Cerca);
        }
    while (Cerca!=0);
```

Esempio - il programma C

```
Volte=0;
do
{ printf ( "immetti un numero della sequenza \n");
  scanf("%d", &Num);
  if (Num==Cerca) Volte=Volte+1;
}
while (Num!=0);
printf ("Numero %d presente %d volte \n", Cerca, Volte);
}
```

Relazione tra while e do-while ...

.... con notazione C:

```
while (<espressione>
{
    S1;
}
```

```
while (Num < 10)
{
    Num++;
}
```

.... è equivalente a:

```
if (<espressione>
do {
    S1;
}
while (<espressione>);
```

.... è equivalente a:

```
if (Num < 10)
do {
    Num++;
}
while (Num < 10);
```

Relazione tra do-while e while ...

```
do
{
    S1;
}
while (<espressione>;
```

.... è equivalente a:

```
S1;
while (<espressione>
{
    S1;
}
```

```
do
{
    scanf("%d", &Num);
}
while (Num!=0);
```

.... è equivalente a:

```
scanf("%d", &Num);
while (Num!=0)
{
    scanf("%d", &Num);
}
```

Relazione tra repeat-until e while ...

.... con notazione LDP:

```
repeat
    S1;
until (<espressione>;
```

.... è equivalente a:

```
S1;
while ( not <espressione>
    S1;
```

```
repeat
    a=a+1;
until (a>35);
```

.... è equivalente a:

```
a=a+1;
while ( a<=35)
    a=a+1;
```

Relazione tra repeat-until e while ...

```
while (<espressione>
      S1;
```

```
while ( x < 26)
      x=x+y;
```

.... è equivalente a:

.... è equivalente a:

```
if (<espressione>
repeat
      S1;
until (not <espressione>);
```

```
if ( x < 26 )
repeat
      x=x+y;
until ( x >= 26 );
```

Strutture Ciclo

Strutture ripetitive

.... strutture del tipo

ripeti N volte S;

N è un intero

S è un'istruzione (sequenza di istruzioni)

semantica:

S viene eseguita N volte prima di procedere con la esecuzione della prima istruzione successiva alla struttura

In quasi tutti i linguaggi è presente nella forma a "conteggio delle ripetizioni" (*ciclo a conteggio*) ...

... linguaggi diversi offrono forme diverse di tale struttura...

Strutture Ciclo

Ciclo a conteggio

for (<espressione1>; <condizione>; <espressione2>) S;

- <espressione1>: è l'espressione usata per inizializzare la **variabile di controllo** del ciclo
- <condizione>: è l'espressione logica per verificare la terminazione del ciclo
- <espressione2>: è l'espressione usata per determinare il **passo di incremento** della variabile di controllo
- S è l'istruzione, o la sequenza, che deve essere ripetuta

Strutture Ciclo

Ciclo a conteggio: un tipico formato

for (cont = *inf*; cont <= *sup* ; cont = cont + *step*) S;

cont, *inf*, *sup* e *step* sono, tipicamente, variabili di tipo intero, o comunque di un tipo ordinato;

S è la sequenza da ripetere;

step > 0 (per la convergenza del ciclo)

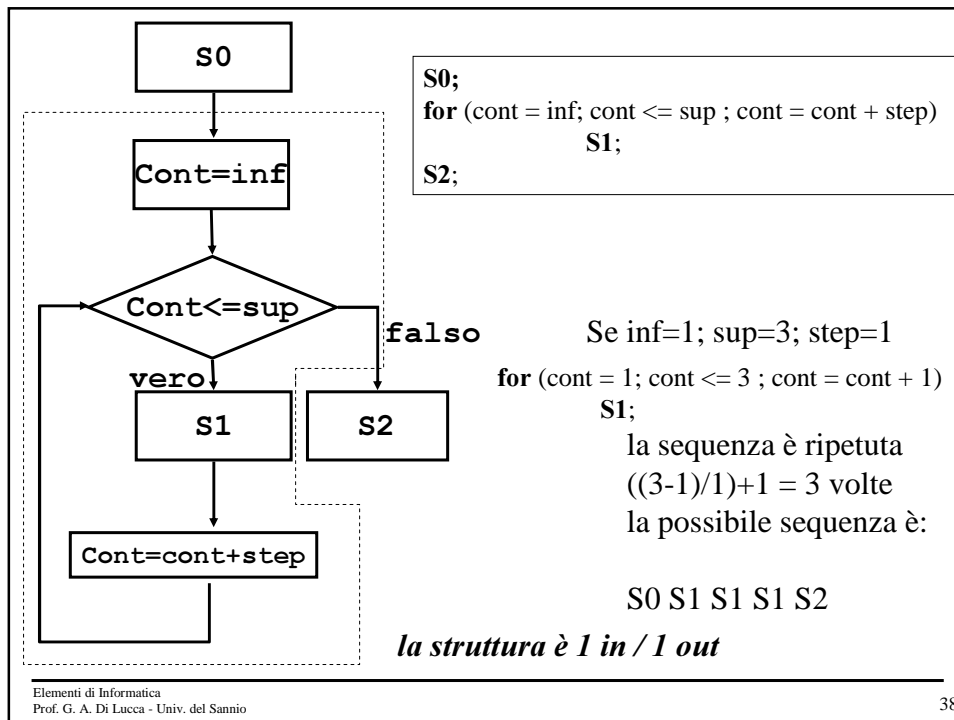
- se *inf* > *sup* il ciclo non viene percorso e la sequenza S non viene eseguita;
- se *inf* <= *sup*, S viene eseguita un numero di volte pari a $((sup - inf) / step) + 1$;
 - la prima volta con cont = *inf*,
 - la seconda con cont = *inf* + *step*,
 - la terza con cont = *inf* + 2 * *step*, ...
- il ciclo termina quando cont > *sup*

Strutture Ciclo

Ciclo a conteggio

for (cont = inf; cont <= sup ; cont = cont + step) S;

- cont, inf, sup e step possono essere usati in S ma il loro valore non va modificato
- il valore di cont deve essere ridefinito se la variabile viene usata dopo il ciclo



Esempio

```

{.....
int x;
for (x=1; x<=5; x++) printf("%d\n",x);
printf("FINE");
}

```

x	x<=5	printf("%d\n",x)
1	si	1
2	si	2
3	si	3
4	si	4
5	si	5
6	no	FINE

Elementi di Informatica
Prof. G. A. Di Lucca - Univ. del Sannio

39

Relazione tra cicli for e while ...

```

for (init; test; step)
{
    S1;
}

```

.... è equivalente a:

```

init;
while (test)
{
    S1;
    step;
}

```

```

for (i = 1; i<10; i++)
{
    printf("%d\n",i);
}

```

.... è equivalente a:

```

i=1;
while (i<10)
{
    printf("%d\n",i);
    i++;
}

```

Elementi di Informatica
Prof. G. A. Di Lucca - Univ. del Sannio

40

Teorema di Bohm e Jacopini

... problema: quali “strutture di controllo” occorrono per descrivere un qualsiasi tipo di algoritmo?

Teorema di Bohm e Jacopini

Qualsiasi algoritmo può essere descritto usando solo tre tipi di strutture di controllo:

- una di sequenza*
- una di selezione*
- una di ciclo*

Ad es.

la sequenza

solo la struttura if per la selezione

solo la struttura while per i cicli