



Cicli e Vettori

Alessandra Giordani

agiordani@disi.unitn.it

Lunedì 12 marzo 2012

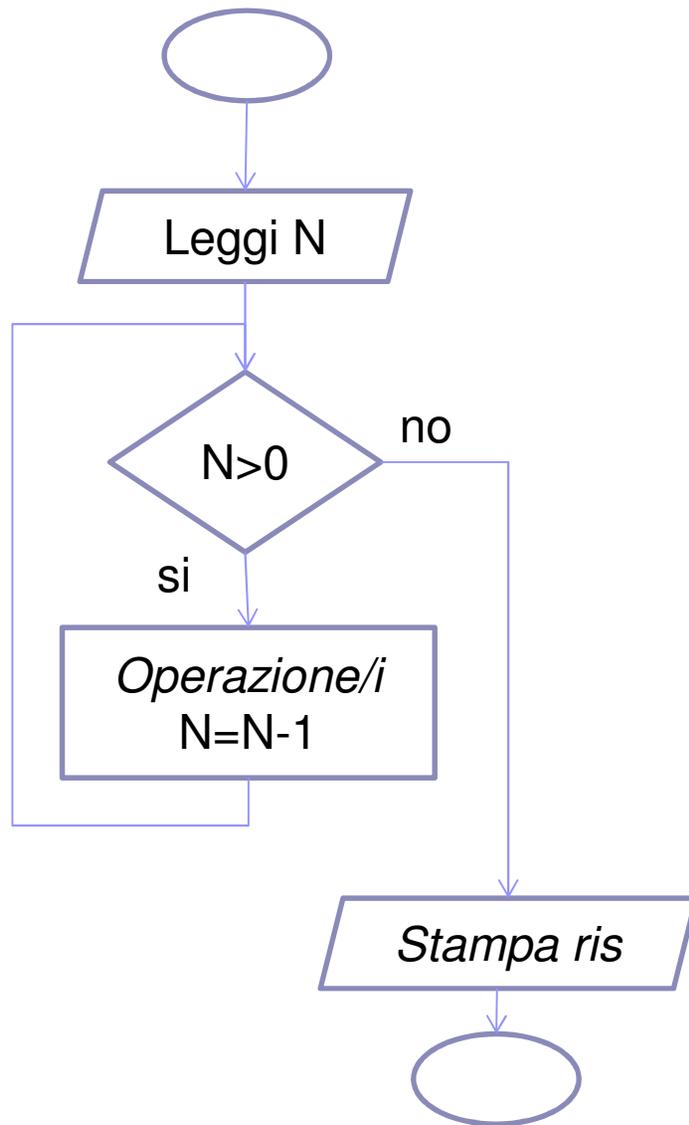
<http://disi.unitn.it/~agiordani/>



I cicli (**while** e **do-while**)

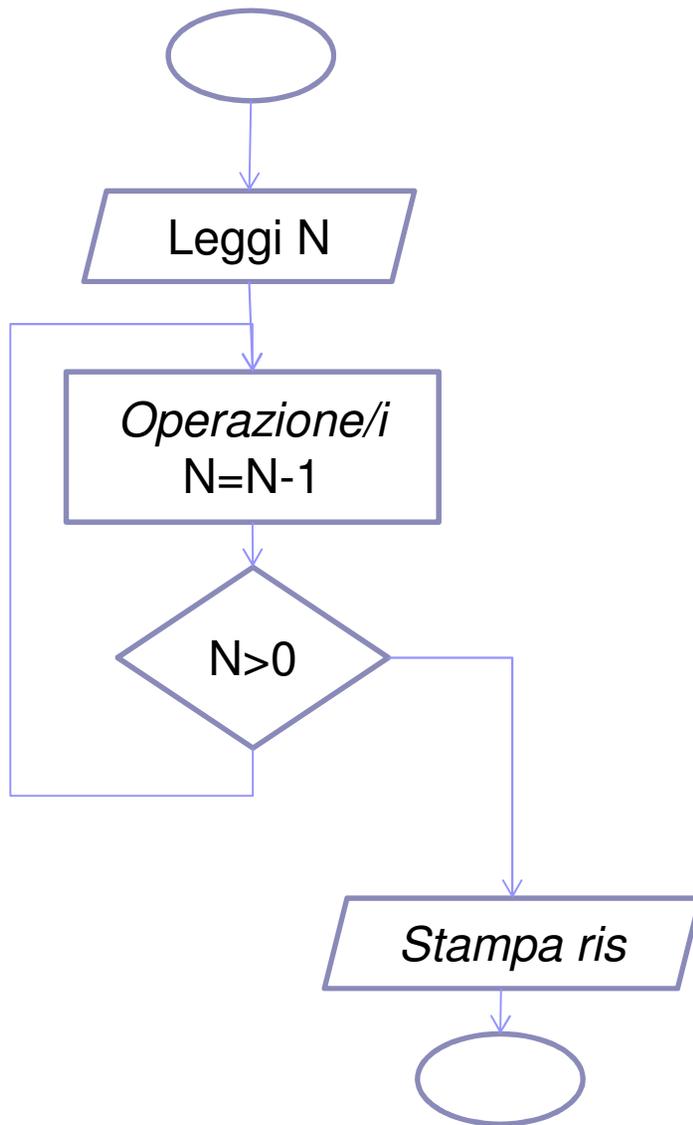
- Quando uso i cicli?
 - Quando voglio ripetere la stessa **operazione** più volte (in generale, N volte)
- Cosa è N?
 - E' un numero dato in input (*es. letto da tastiera*)
 - Ad ogni ciclo va modificato (*es. decrementato*)
 - Quando N è nullo esco dal ciclo
- Differenza tra **while** e **do-while**?
 - **do-while** esegue almeno una volta l'operazione, anche se N è nullo da principio

Ciclo while



1. Leggo il numero N
2. Se $N \leq 0$ vai a 6
3. Eseguo operazione/i
4. Decremento N
5. Se N è positivo torna a 3
6. Stampo il risultato dell'operazione/i

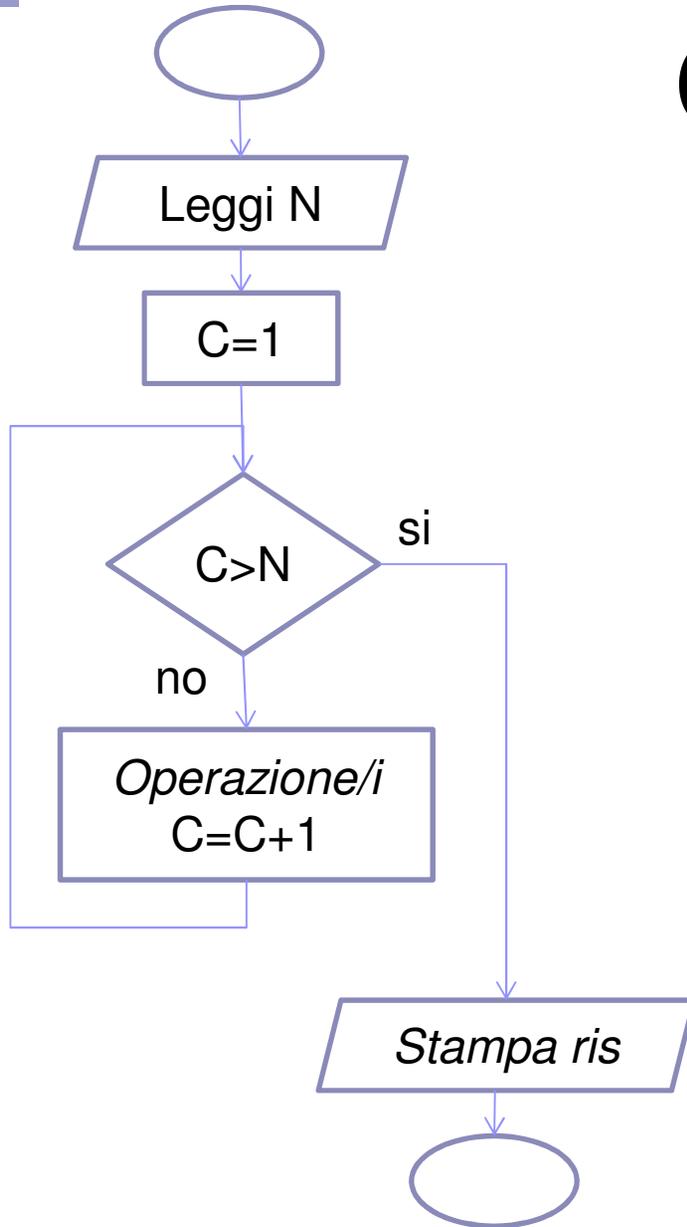
Ciclo do-while



1. Leggo il numero N
2. Eseguo operazione/i
(almeno una volta anche quando N non è positivo)
3. Decremento N
4. Se N è positivo torna a 2
5. Altrimenti stampo il risultato dell'operazione/i

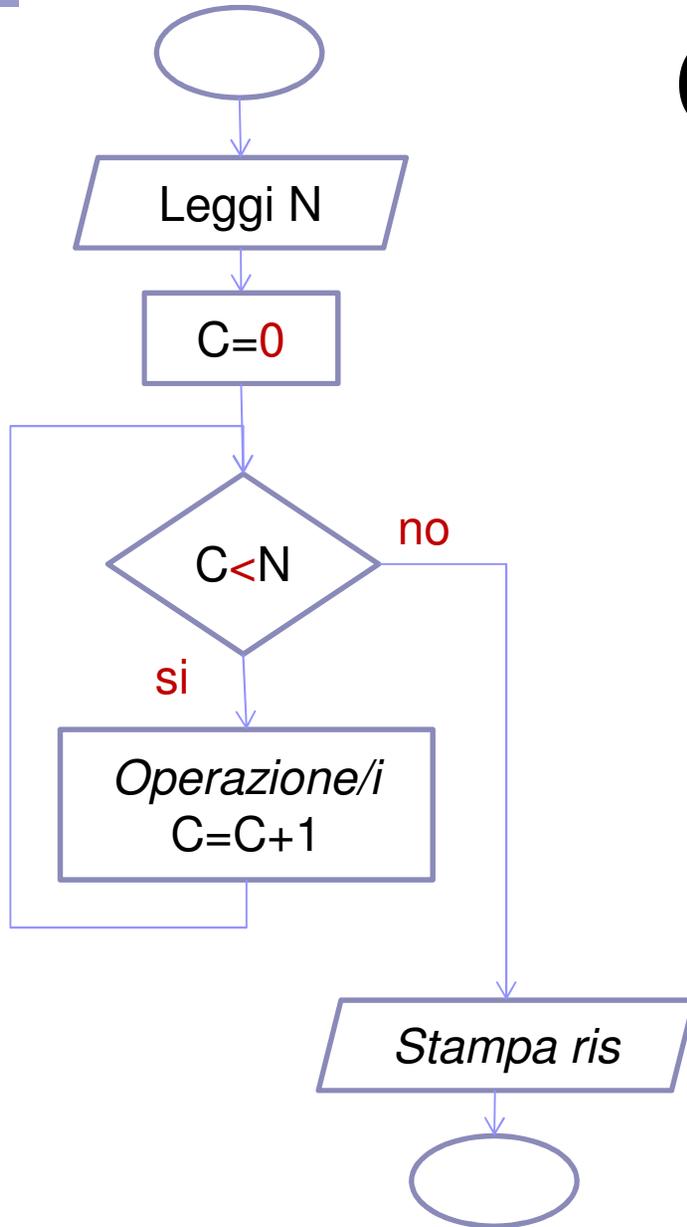
no

si



Ciclo **while** (ver.2)

1. Leggo il numero N
2. **Use un contatore C** inizialmente uguale a 1
3. Se $C > N$ vai a 7
4. Eseguo operazione/i
5. **Incremento C**
6. Se $C \leq N$ torna a 3
7. Stampo il risultato dell'operazione/i



Ciclo **while** (ver.3)

1. Leggo il numero N
2. **Uso un contatore C** inizialmente **uguale a 0**
3. Se $C \geq N$ vai a 7
4. Eseguo operazione/i
5. **Incremento C**
6. Se $C < N$ torna a 3
7. Stampo il risultato dell'operazione/i



Gli indici

- Servono per contare a che iterazione/posizione siamo arrivati
- Naturalmente viene spontaneo partire da 1 e contare N volte (1,2,...,N)
- MA in C si parte da 0 e si conta sempre N volte, fino ad N-1 (0,1,2,...,N-1)
- Quindi attenzione anche alle condizioni di uscita dal ciclo ($i < N$ invece che $i \leq N$)



Esercizio 1: Media di N numeri

- Leggere N
 - Leggere N numeri
 - Farne la somma ($\text{somma} = \text{somma} + \text{num}$)
 - Calcolare la media (somma/N)
 - Stampare il risultato (media)
-
- Disegnare il diagramma di flusso



Soluzioni

- Ci possono essere più diagrammi di flusso
- Ciclo while normale
- Ciclo while con contatore
- Ciclo do-while
 - Supponendo $N > 0$
 - Ma se $N = 0$?



Esercizio 2: calcolare la varianza

- Leggere N numeri
- Calcolarne la media
- Fare la somma degli scarti quadratici medi (differenza tra ogni numero e il valore della media elevato al quadrato)
- **PROBLEMA:**
 - Quanti cicli devo fare?
 - Come memorizzo i numeri senza rileggerli per la seconda volta?



Array e Matrici

- Gli array sono sequenze di dimensione finita di variabili omogenee
- Ciascun elemento di un array è una variabile
- E' possibile definire array uni-dimensionali (vettori) e bi-dimensionali (matrici)
- Ciascun elemento puo essere acceduto mediante il suo indice
 - `vettore[indice]`
 - `matrice[indice][indice]`
- Il primo elemento di un array di dimensione n ha indice 0, l'ultimo $n-1$



Soluzione esercizio 2

- Leggo N
- Ciclo di lettura degli N numeri e memorizzazione in un array (utilizzo un indice $0 \dots N-1$, es. *leggi $A[i]$*)
- Calcolo media (ciclo separato o in parallelo al precedente)
- Ciclo per il calcolo degli scarti (accedo al vettore ripartendo dall'indice 0)
- Stampo varianza



Dall'analisi alla programmazione

- Non vedremo subito la traduzione di questi diagrammi da pseudocodice/flow-chart a programmi in linguaggio C
 - MA possiamo iniziare con programmi più semplici (senza cicli).
 - Esempi
 - QUADRATO di un numero
 - SOMMA e MEDIA di due numeri
- Per ora NON letti da tastiera

Quadrato di un numero

```
int main()
```

Tipo della funzione
Nome della funzione

```
{  
    int quadrato;  
    int num;  
  
    num=2;  
    quadrato=num*num;  
}
```

Dichiarazione di una variabile
Istruzioni C eseguibili
Corpo della funzione



Nota sull'uso del compilatore:

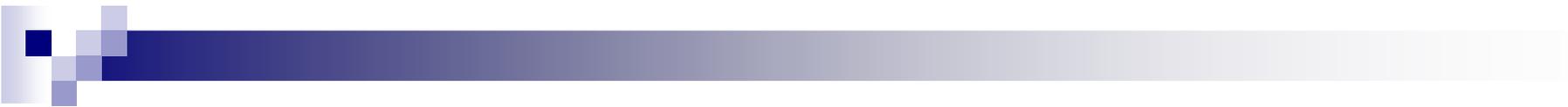
Specificare file di output

- Quando si invoca GCC è anche possibile specificare un nome per il file di output:

```
$ gcc <file sorgente C> -o <nome file di output>
```

- oppure, equivalentemente:

```
$ gcc -o <nome file di output> <file sorgente C>
```



Es. compilazione helloworld.c 2

- In questo modo, al posto del file "a.out" viene creato un file con il nome indicato. Ad esempio:

```
$ gcc helloworld.c -o helloworld
```

- compila il codice sorgente contenuto in "helloworld.c" in un file eseguibile chiamato "helloworld", nella directory corrente. Il file così compilato può essere eseguito con il comando:

```
$ ./helloworld
```



Quadrato di un numero

```
#include <stdio.h>

int main()
{
    int num;
    int quadrato;

    num = 2;
    quadrato = num * num;
    printf("Il quadrato e' %d\n", quadrato);
}
```

- \$ gcc quadrato.c -o quadrato
- \$./quadrato



Somma e media di 2 numeri

```
#include <stdio.h>
int main(){
int somma;
float media;
somma=100;
somma=somma+200;
media=somma/2;

printf("Somma %d\n",somma);
printf("Media %f\n",media);
}
```

- \$ gcc media.c -o media
- \$./media