

Laboratorio di programmazione per sistemi ciberfisici

4. Costrutti di iterazione

Enrico Martini
October 28, 2025

I Cicli

Consentono di eseguire ripetutamente una serie di istruzioni.

- ▶ `while`
- ▶ `do-while`
- ▶ `for`

Il ciclo while

Sintassi:

```
1 while ( condizione ){  
2     corpo del ciclo  
3 }
```

- ▶ Se il valore della condizione è vera ($\neq 0$), allora viene eseguito il corpo del ciclo
- ▶ Quando la condizione diventa falsa (0), il ciclo termina
- ▶ Il corpo del ciclo potrebbe non essere mai eseguito, se la condizione è falsa in partenza
- ▶ Il ciclo potrebbe non finire mai se la condizione rimanesse vera per sempre. **Deve essere modificata all'interno del ciclo**

Esempio

Scrivere un programma che somma una serie di numeri inseriti dall'utente (scrivi 0 per uscire)

```
1  int n, sum = 0;           // Dichiarazione variabili
2  scanf("%i", &n);         // Inserisco il primo numero
3  while(n != 0){           // Fintantochè n e' diverso da zero
4      sum += n;             // Sommo n a sum (sum e' un accumulatore)
5      scanf("%i", &n);      // Leggo il numero successivo
6  }
7  printf("Somma: %i\n", sum);
```

Il ciclo do-while

Sintassi:

```
1 do{  
2     corpo del ciclo  
3 } while( condizione );
```

- ▶ L'espressione di controllo viene testata **dopo** l'esecuzione del corpo del ciclo
- ▶ Il corpo del ciclo viene **sempre eseguito almeno una volta**, indipendentemente dalla condizione

Esempio 1

Scrivere un programma che somma una serie di numeri inseriti dall'utente (scrivi 0 per uscire)

```
1  int n, sum = 0;           // Dichiarazione variabili
2
3  do {
4
5      scanf("%i", &n);      // Leggo il numero
6      sum += n;             // Sommo n a sum (sum e' un accumulatore)
7
8  } while(n != 0);          // Fintantochè n e' diverso da zero
9
10 printf("Somma: %i\n", sum);
```

Esempio 2

Scrivere un programma che continui a chiedere all'utente un numero finchè non è positivo.

```
1  int n;  
2  
3  do{  
4  
5      printf("Inserisci un numero positivo: ");  
6      scanf("%i", &n);  
7  
8  } while(n < 0);  
9  
10 printf("Il numero %i e' positivo.\n",n);
```

Il ciclo for

Sintassi:

```
1 for ( espressione iniziale ; condizione ; espressione di fine corpo ){  
2     corpo del ciclo  
3 }
```

- ▶ Ideale per i cicli che hanno una **variabile contatore**
- ▶ L'espressione iniziale serve ad impostare i valori iniziali prima che cominci il ciclo. Contiene un **assegnamento o una dichiarazione** con assegnamento
- ▶ Il ciclo continua fintanto che la condizione è soddisfatta. Quando falsa, allora si esce dal ciclo
- ▶ L'espressione di fine corpo viene eseguita **dopo ogni esecuzione** del corpo del ciclo

Esempio

Scrivere un programma che stampi la tavola dei primi n quadrati, con n inserito dall'utente (usando il ciclo for)

```
1  int n;                                // Dichiarazione
2
3  printf("Inserisci n: ");
4  scanf("%i", &n);
5
6  for( int i = 1 ; i <= n ; i++ ){      // i va da 1 a n
7      printf("%i - %i\n", i, i*i);      // Stampo il valore di i
8  }
```

Esempi di conteggio:

da 0 a n-1		for(i = 0 ; i < n ; i++){ ... }
da 1 a n		for(i = 1 ; i <= n ; i++){ ... }
da n a 1		for(i = n ; i > 0 ; i--){ ... }
da n-1 a 0		for(i = n-1 ; i >= 0 ; i--){ ... }

- ▶ E' possibile utilizzare l'istruzione `break` per uscire da un ciclo da un qualsiasi punto
- ▶ E' possibile andare all'iterazione successiva tramite l'istruzione `continue`

Equivalenza for - while - do while

while:

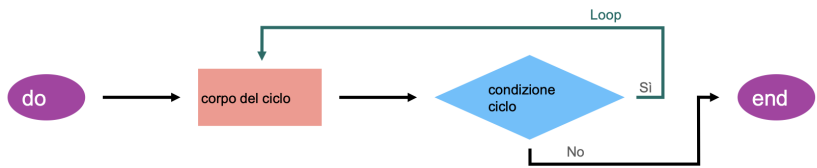
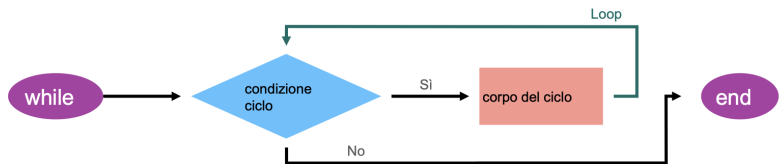
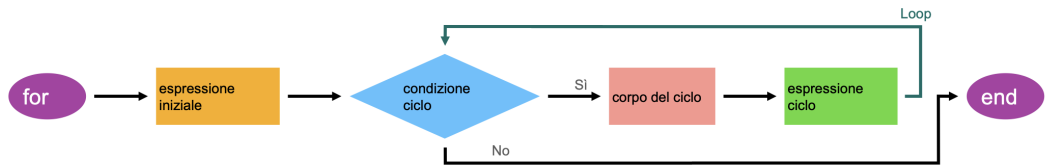
```
1 espressione iniziale
2 while( condizione ){
3     corpo del ciclo
4     espressione di fine corpo
5 }
```

do while:

```
1 espressione iniziale
2 do{
3     corpo del ciclo
4     espressione di fine corpo
5 } while( condizione );
```

for:

```
1 for ( espressione iniziale ; condizione ; espressione di fine corpo ){
2     corpo del ciclo
3 }
```



Esercizio 1

Scrivere un programma che stampi la tabellina di un numero n inserito dall'utente. La tabellina deve essere calcolata da 1 fino ad un valore y , sempre inserito dall'utente.

Esercizio 2

Scrivi un programma che, dato un intero N positivo inserito dall'utente, ne calcoli il numero triangolare (somma dei numeri da 1 a N).

Esercizio 3

Scrivi un programma che, dato un intero N positivo inserito dall'utente, ne calcoli il fattoriale (prodotto dei numeri da 1 a N).

Esercizio 4

Scrivere un programma che conta le cifre di un intero, inserito dall'utente.

Esercizio 5

Scrivere un programma che data base ed esponente ne calcoli la potenza.

Esercizio 6

Scrivere un programma che inverta le cifre di un numero intero inserito dall'utente.

Esercizio 7

Scrivere un programma che stampi il calendario di un mese, dati in input il numero di giorni e il giorno della settimana in cui comincia (1=lun, 7=dom).

Esercizio 8

Scrivere un programma che approssimi il valore di e (con n intero positivo immesso dall'utente):

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \dots + \frac{1}{n!} \quad (1)$$

Esercizio 9

Scrivi un programma che chieda all'utente un numero intero n e stampi la serie di fibonacci dei primi n numeri. La serie di Fibonacci inizia con 0 e 1, e ogni numero successivo è la somma dei due precedenti (1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144).

Esercizio 10

Scrivere un Programma C che stampi una matrice $n \times n$, dove n è un valore richiesto all'utente. In ogni cella della matrice va stampata la coppia (i, j) dove i è il numero di riga e j il numero di colonna.

Suggerimento: Utilizzare opportunamente le stringhe di formato (%2i) per impostare la larghezza di campo. Assumere che n sia composto al massimo da 2 cifre.

```
1 Inserire un numero intero: 4
2 ( 1, 1) ( 1, 2) ( 1, 3) ( 1, 4)
3 ( 2, 1) ( 2, 2) ( 2, 3) ( 2, 4)
4 ( 3, 1) ( 3, 2) ( 3, 3) ( 3, 4)
5 ( 4, 1) ( 4, 2) ( 4, 3) ( 4, 4)
```

Esercizio 11

Scrivere un programma che calcoli la serie armonica fino al n-esimo termine, controllando che n sia maggiore di 1.

$$r = \sum_{k=1}^n \frac{1}{k}$$

Esercizio 12

Scrivere un programma per il calcolo del coefficiente binomiale, definito come:

$$\binom{a}{n} = \prod_{k=1}^n \frac{a - k + 1}{k}$$

dove $a, n \in \mathbb{N}$ sono inseriti dall'utente, con $n < a$.