

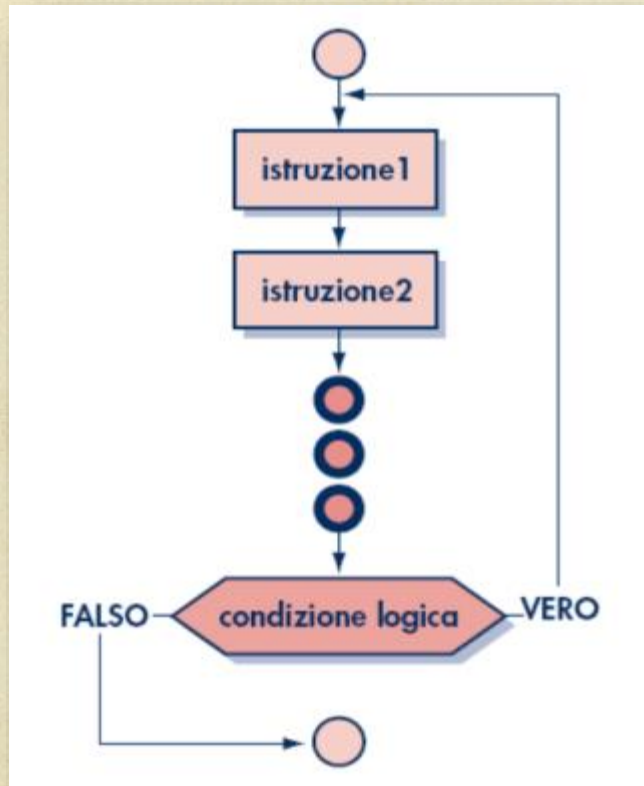
Iterazione

A. Ferrari

Iterazioni (cicli)

- L'iterazione è una struttura che consente di ripetere più volte l'esecuzione di un insieme di istruzioni.
- Due tipi di iterazione:
 - iterazioni enumerative: consentono di eseguire un blocco di istruzioni un numero predefinito di volte.
 - iterazioni guidate da eventi: non è noto a priori quante volte saranno eseguite.
Se la condizione specificata è vera (o è ancora vera) il blocco viene eseguito, altrimenti si passa all'istruzione successiva al termine del blocco.

Iterazioni postcondizionali



- L'iterazione postcondizionale è quella fondamentale, infatti tutte le altre possono essere definite a partire da questa.
- Il blocco delle istruzioni che fanno parte dell'iterazione viene eseguito almeno una volta poi si verifica la condizione e si ripete il blocco fintanto che la condizione è vera.
- Sono soddisfatti i requisiti della programmazione strutturata: un solo punto d'ingresso e un solo punto d'uscita.

do ... while

▶ Sintassi:

```
do
    <istruzione>
while (<espressione>);
```

▶ L'istruzione continua a essere eseguita fintanto che <espressione> risulta vera.

▶ Nel caso, molto frequente, in cui il corpo del ciclo sia costituito da più istruzioni, si utilizza una coppia di parentesi graffe per racchiudere l'insieme delle istruzioni che fanno parte del ciclo.

```
do
{
    <istruzione1>;
    <istruzione2>;
    ...
    <istruzioneN>;
} while (<espressione>);
```

Esercizio 1

Iterazione guidata da eventi

- Scrivere un programma che richiede in input un numero intero e lo continua a richiedere finché non viene inserito un numero pari.

Soluzione 1

```
/*
 * Richiede in input un valore intero pari poi lo
 * visualizza
 */
#include <stdio.h>
int main()
{
    int valore;
    printf("Inserisci un numero pari ");
    do
        scanf("%d",&valore);
    while ((valore%2)!=0);
    printf("%d e' un numero pari",valore);
}
```

Soluzione 2

```
/*
 * Richiede in input un valore intero pari poi lo
 * visualizza
 */
#include <stdio.h>
int main()
{
    int valore;
    do
    {
        printf("Inserisci un numero pari ");
        scanf("%d",&valore);
    }
    while ((valore%2)!=0);
    printf("%d e' un numero pari",valore);
}
```

Soluzione 3

```
/*
 * Richiede in input un valore intero pari poi lo
   visualizza
 */
#include <stdio.h>
int main()
{
    int valore;
    do
    {
        printf("Inserisci un numero pari ");
        scanf("%d",&valore);
        if((valore%2)!=0)
            printf("il numero inserito non e' pari ...\n");
    }
    while ((valore%2)!=0);
    printf("%d e' un numero pari",valore);
}
```

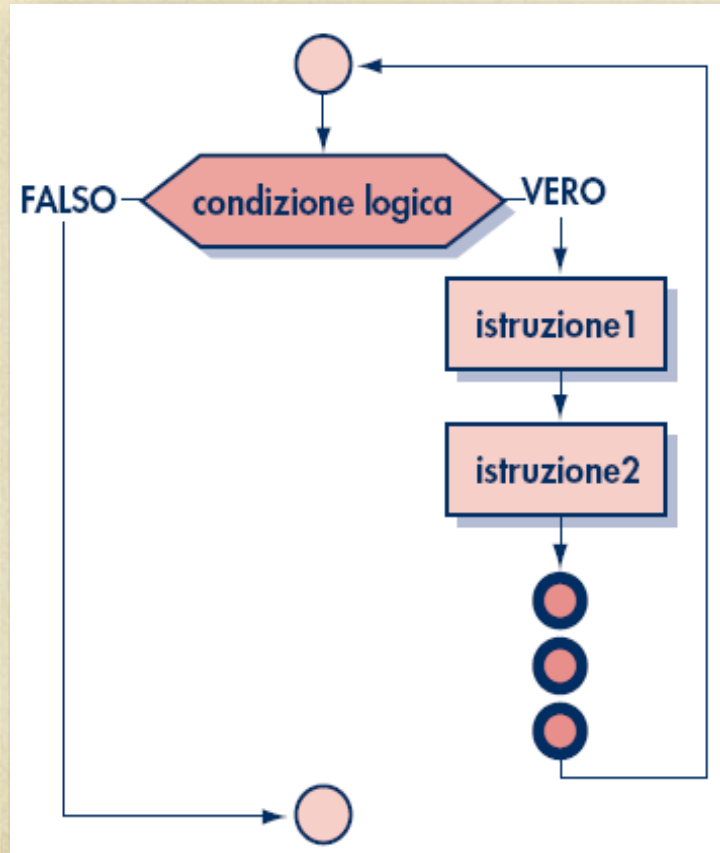

Esercizio 2

Iterazione enumerativa

- Realizzare un programma che determini la potenza intera di un numero utilizzando soltanto l'operatore prodotto.
Il valore della potenza (l'esponente) e il numero (la base) sono immessi dall'utente.

```
#include <stdio.h>
int main()
{
    int base,esponente,potenza;
    int contatore;    //conta quante volte viene eseguito il ciclo
    printf("inserisci la base ");
    scanf("%d",&base);
    printf("inserisci l' esponente");
    scanf("%d",&esponente);
    potenza = 1;
    contatore= 0;
    do
    {
        potenza = potenza * base;
        contatore = contatore +1;    //numero di prodotti eseguiti
    }
    while (contatore<esponente);
    printf("potenza = %d ",potenza);
}
```

Iterazioni precondizionali



- Le iterazioni precondizionali hanno la condizione in testa al ciclo.
- La logica è quella di verificare una condizione e continuare a eseguire il blocco delle istruzioni che fanno parte dell'iterazione fintanto che questa risulti vera.
- Il blocco di istruzioni può, nel caso in cui la condizione risulti subito falsa, non essere mai eseguito.

Il ciclo while

- Il linguaggio C mette a disposizione il ciclo while

```
while (<espressione>)
```

```
<istruzione>;
```

- L'istruzione continua a essere eseguita fino a quando <espressione> risulta vera.
- Nel caso, molto frequente, in cui il corpo del ciclo sia costituito da più istruzioni, si utilizza una coppia di parentesi graffe per racchiudere l'insieme delle istruzioni che fanno parte del ciclo.

Esempio - while 1

```
/*  
 * Richiede in input un valore intero pari poi lo  
   visualizza  
*/  
#include <stdio.h>  
int main()  
{  
    int valore;  
    printf("Inserisci un numero pari ");  
    scanf("%d",&valore);  
    while ((valore%2)!=0)  
        scanf("%d",&valore);  
    printf("%d e' un numero pari",valore);  
}
```

Esempio - while 2

```
/*
 * Richiede in input un valore intero pari poi lo
   visualizza
 */
#include <stdio.h>
int main()
{
    int valore;
    printf("Inserisci un numero pari ");
    scanf("%d",&valore);
    while ((valore%2)!=0)
    {
        printf("Il numero non e' pari, reinserirlo");
        scanf("%d",&valore);
    }
    printf("%d e' un numero pari",valore);
}
```

Esercizi

- Realizza un programma che determini la potenza intera di un numero utilizzando soltanto l'operatore prodotto.
Il valore della potenza (l'esponente) e il numero (la base) sono immessi dall'utente.
Utilizzare il ciclo while
- Si riceve come dato d'ingresso una sequenza di numeri terminante per 0, i numeri sono al massimo 100, non è conosciuta a priori la lunghezza della sequenza.
Visualizza il valore del numero maggiore e di quello minore

Iterazioni enumerative

- In molti casi, è noto a priori quante volte il blocco di istruzioni interne al ciclo debba essere eseguito
- Le strutture iterative presentate possono essere utilizzate anche per risolvere questo tipo di problemi.
- Viene utilizzata una variabile (contatore) che ha la funzione di “contare” il numero di ripetizioni;
- Il contatore viene inizializzato prima del ciclo con un valore di partenza, e poi incrementato a ogni esecuzione;
- La condizione di ciclo controlla il raggiungimento del numero di iterazioni voluto.

Problema

- Problema: calcolare la somma, il prodotto e la media di n valori immessi dall'utente.
- Input: la lunghezza n della sequenza dei valori in ingresso, gli n valori della sequenza.
- Output: somma, prodotto e media dei valori immessi.

Il ciclo for

- Ogni linguaggio presenta uno o più costrutti di programmazione per implementare le iterazioni enumerative
- Il C fornisce il ciclo for che, in realtà, è un costrutto molto potente e può essere utilizzato per qualsiasi tipo di iterazione.
- Per semplicità, lo utilizzeremo solo per implementare iterazioni enumerative.

for - sintassi

- `for(<istruzione1>;<espressione>;<istruzione2>)`
 <istruzione da ripetere>;
- <istruzione1>
 - viene eseguita prima del ciclo e viene utilizzata per inizializzare la variabile contatore
- <espressione>
 - è la condizione logica che deve essere soddisfatta per eseguire il corpo del ciclo
- <istruzione2>
 - viene eseguita ad ogni iterazione ed è utilizzata per incrementare/decrementare la variabile contatore
- <istruzione da ripetere>
 - è il corpo del ciclo che normalmente è costituito da più istruzioni racchiuse in una coppia di parentesi graffe

Esempi

```
/* tabellina del 2 */
#include <stdio.h>
int main()
{
    int ind;        //indice
    int prodotto;
    for(ind=1;ind<=10;ind++)
    {
        prodotto=2*ind;
        printf("%d
        ",prodotto);
    }
}
```

```
/* tabellina del 2 inverso*/
#include <stdio.h>
int main()
{
    int ind;        //indice
    int prodotto;
    for(ind=10;ind>=1;ind--)
    {
        prodotto=2*ind;
        printf("%d
        ",prodotto);
    }
}
```