

Appunti di Sistemi Elettronici sulla programmazione

LA PROGRAMMAZIONE

La programmazione costituisce una parte fondamentale dell'informatica. Infatti solo attraverso di essa si apprende la logica che ci permette di comunicare al computer le azioni da effettuare.

Il computer non dispone di una conoscenza a priori e senza il nostro aiuto non è in grado di svolgere alcuna operazione. Tutto quello che vogliamo che esso compia deve essere specificato nei minimi dettagli; in altre parole è compito del programmatore trasmettere i termini del problema al computer insieme con le operazioni da eseguire.; solo a questo punto l'utente può richiedere alla macchina di svolgere una determinata azione.

L'algoritmo

Il termine di algoritmo deriva dal nome di un matematico arabo Al-Khuwarizmi, vissuto nel IX secolo. Quando ci troviamo di fronte ad un problema, tentando di sviluppare un procedimento di soluzione cerchiamo l'algoritmo; cioè

L'algoritmo è un insieme di operazioni (non solo di natura matematica) che portano alla soluzione di un problema.

L'algoritmo deve possedere caratteristiche di completezza e logicità: deve prevedere ogni azione necessaria al raggiungimento della soluzione e la sequenza delle operazioni deve rispettare la logica del problema.

Esempio di problema generico:

“ Dato un distributore automatico e un gettone, prelevare una bibita”.

La sequenza delle operazioni che portano al completamento dell'obiettivo è la seguente:

Inizio:

- 1) Verificare che il distributore sia in funzione (luce verde accesa);
- 2) In caso positivo, inserire il gettone nell'apposita fessura;
- 3) Selezionare una bibita tra quelle che presentano una luce rossa (distributore pieno);
- 4) Attendere la discesa della bottiglia;
- 5) Prelevare la bottiglia.

Fine

Questo è uno dei tanti problemi rappresentabili attraverso una sequenza ordinata di operazioni elementari.

Esempio di problema matematico:

“ Dati due numeri A e B in ingresso calcolare la sua somma C e visualizzarla sullo schermo”.

La sequenza delle operazioni che portano al completamento dell'obiettivo è la seguente:

Inizio:

- 1) Inserire il primo numero A;
- 2) Leggere il numero A;
- 3) Inserire il secondo numero B
- 4) Leggere il numero B;
- 5) Eseguire la somma tra A e B;
- 6) Assegna alla variabile C il risultato della somma;
- 7) Comunicare sullo schermo il risultato C

Fine.

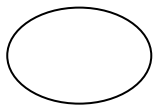
Diagrammi di flusso o FLOW-CHART

Nel diagramma di flusso ad ogni istruzione corrisponde un particolare simbolo grafico.

La forma dei simboli dipende dal tipo di istruzione considerata e ogni blocco è collegato ad un altro mediante una freccia che individua la sequenza di istruzioni. Diventa così possibile schematizzare graficamente i vari passaggi di un algoritmo scrivendo all'interno di ogni forma geometrica una sola istruzione.

Simbologia per i diagrammi di flusso

Nei diagrammi di flusso o FLOW-CHART vengono utilizzati i seguenti simboli:



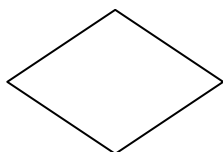
Inizio/Fine
programma



Operazioni interne



Operazioni di ingresso e uscita (I/O)



Operazioni di selezione

In particolare si tenga presente che:

- Per **operazioni di Input/Output** si intendono generalmente tutte le operazioni che permettono di trasferire dati dall'interno del calcolatore al mondo esterno e viceversa. Si considerano di conseguenza come dispositivi di input la tastiera e il mouse, come dispositivi di output il video e la stampante, ma anche le memorie di massa non volatili (floppy disk, hard disk, nastri) sono considerati dispositivi di I/O.
- Le **operazioni interne** sono tutte quelle operazioni di calcolo o più in generale di elaborazione dell'informazione che non siano quindi operazioni di I/O.
- Ogni diagramma di flusso deve avere **un solo punto di inizio**, ma può avere nessuno, uno o molti punti di fine.

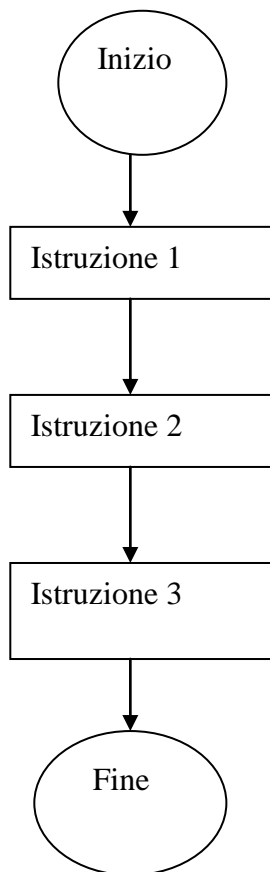
1) Struttura sequenziale

**La struttura sequenziale svolge le operazioni una di seguito all'altra:
esempio:**

Inizio

- 1) esegui istruzione 1
- 2) esegui istruzione 2
- 3) esegui istruzione 3
- 4)
- 5) Esegui istruzione n

Fine



2) Schema di selezione binaria o booleano

La struttura di controllo è binaria o booleana se ci permette di fare una selezione tra due soli casi possibili. Lo schema di selezione binaria ha quindi due sole uscite: cioè VERO/FALSO, oppure SI/NO, ACCESO/SPENTO

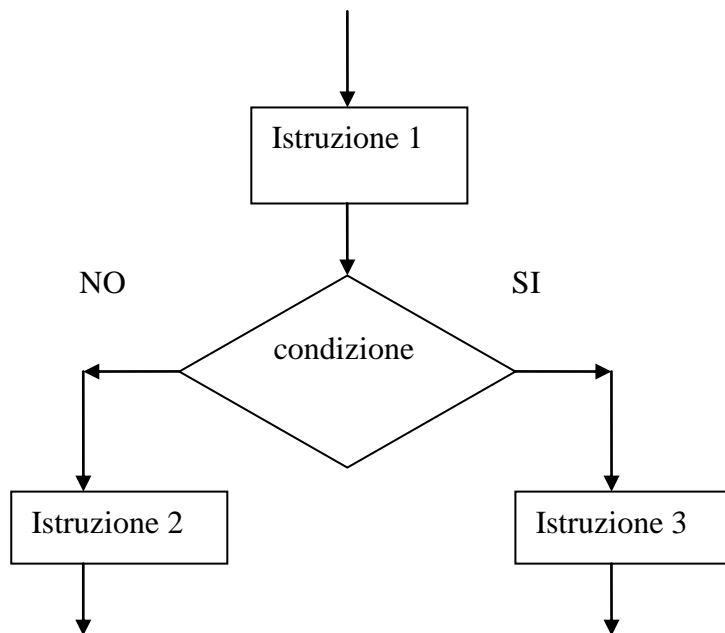


Figura di selezione binaria

algoritmo:

inizio

esegui istruzione 1

se condizione è vera **allora**

 esegui istruzione 3

altrimenti

 esegui istruzione 2

fine

3) Struttura di selezione multipla

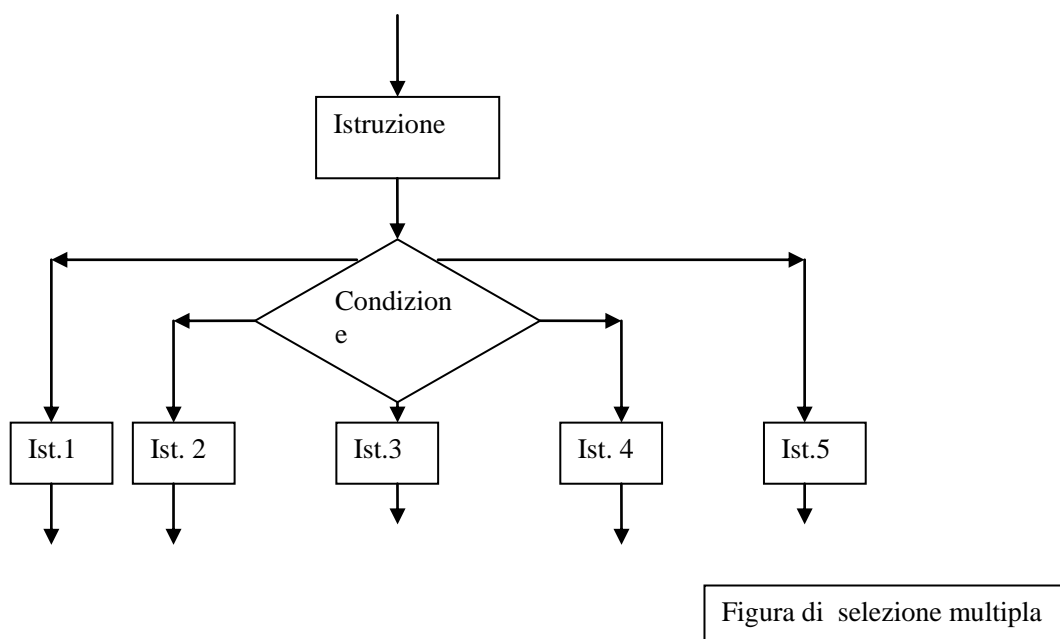
Si vuole adesso analizzare un'altra struttura di controllo, che risulta da una estensione del costrutto "SE...allora...altrimenti", cioè la selezione multipla. La selezione binaria dà la possibilità di scegliere tra due alternative (e la condizione di test è quindi booleana), la selezione multipla permette di selezionare più cammini differenti; la condizione di test non è più booleana, ossia a due valori, bensì a più valori. Nella selezione multipla le uscite dal blocco di confronto sono più di due:

algoritmo:

inizio:

```

se condizione 1 è vera allora
    esegui istruzione 1
altrimenti esegui istruzione 2
se condizione 2 è vera allora
    esegui istruzione 2
altrimenti esegui istruzione 3
se condizione 3 è vera allora
    esegui istruzione 3
altrimenti esegui istruzione 4
....
se condizione n è vera allora
    esegui istruzione n
altrimenti ...
....
Fine
    
```



4) SCHEMA DI ITERAZIONE

Lo schema di iterazione è uno schema in cui un'azione o un gruppo di azioni può essere ripetuta un numero di volte finito o infinito. **Nella risoluzione di un problema può essere necessario ripetere un insieme di operazioni elementari (LOOP = CICLO) fino a quando non si raggiunge una determinata condizione.**

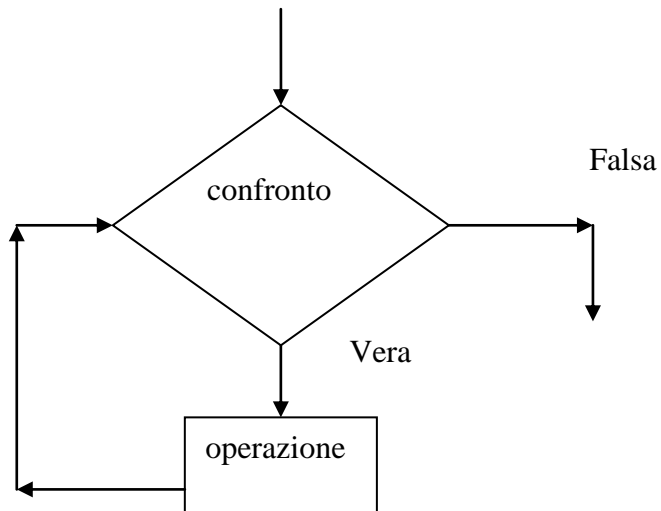


Figura:Schema di iterazione

a) CONTROLLO DI TESTA o schema di iterazione indefinita

Lo schema di iterazione indefinita (cioè non conosciamo a priori il numero di ripetizioni necessarie) viene anche detto con controllo di testa o con controllo di entrata. In questo caso se all'inizio del ciclo la condizione è subito FALSA il ciclo non viene mai eseguito. Esso ha il seguente schema:

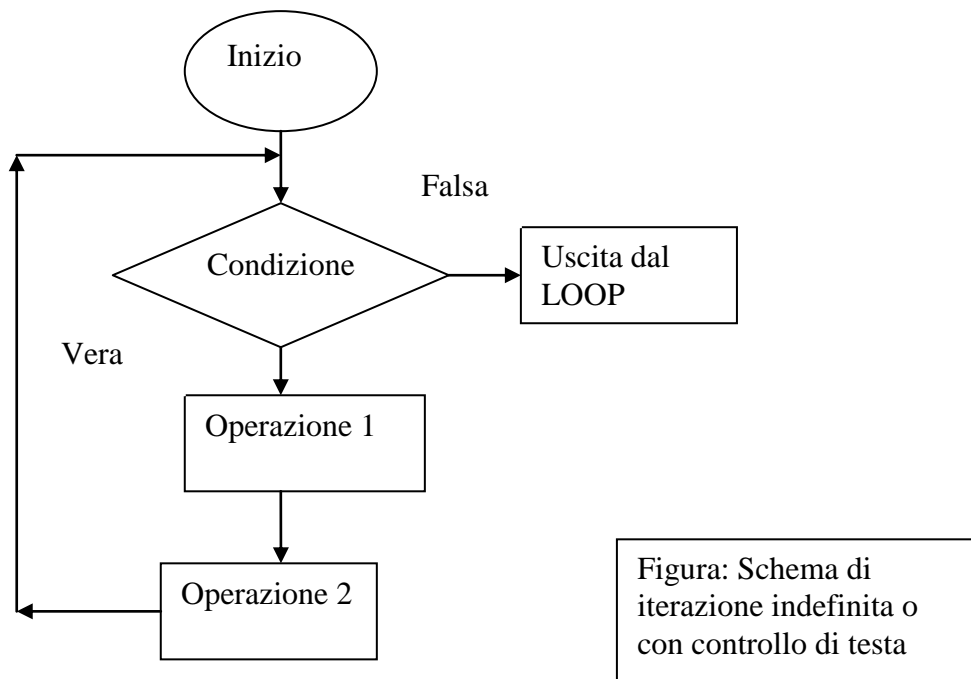


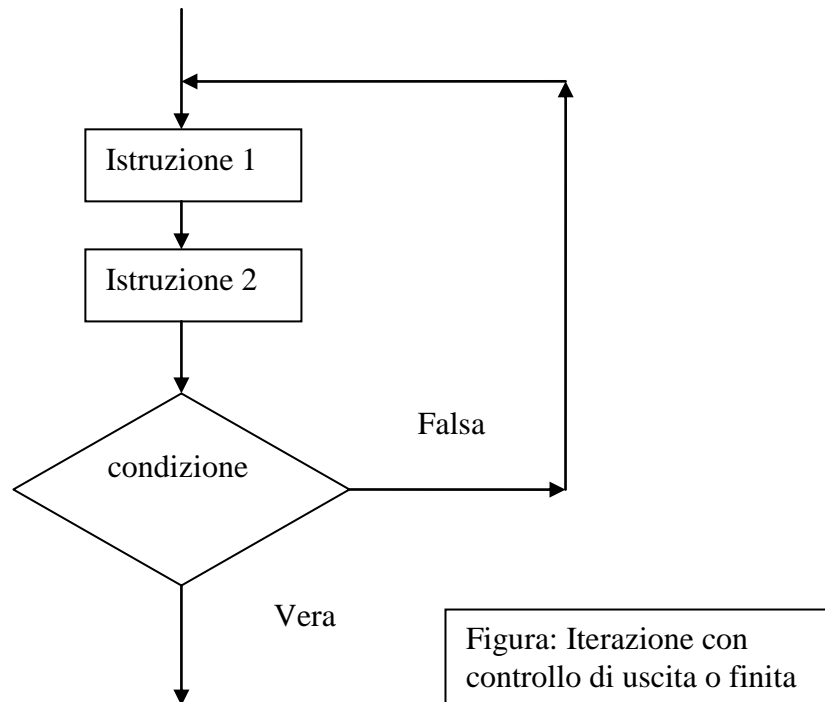
Figura: Schema di iterazione indefinita o con controllo di testa

algoritmo:

```
inizio  
mentre condizione = Vera  
  esegui operazione 1  
  esegui operazione 2  
  ...  
fine
```

B) CONTROLLO DI CODA o schema di iterazione finita

In questo caso conosciamo a priori il numero di ripetizioni necessario; questa ulteriore struttura condizionale è l'iterazione con controllo di uscita o con controllo di coda.



Algoritmo:

ripeti

 esegui istruzione 1

 esegui istruzione 2

Finchè condizione = VERA

Cioè: ripeti la sequenza di comandi specificata, finchè condizione non diventa vera (ripet... finchè in italiano è ambiguo, perché a volte assume il significato di mentre; in questo caso invece qui corrisponde all'inglese repeat ... until).