

Metodo di Lavoro:

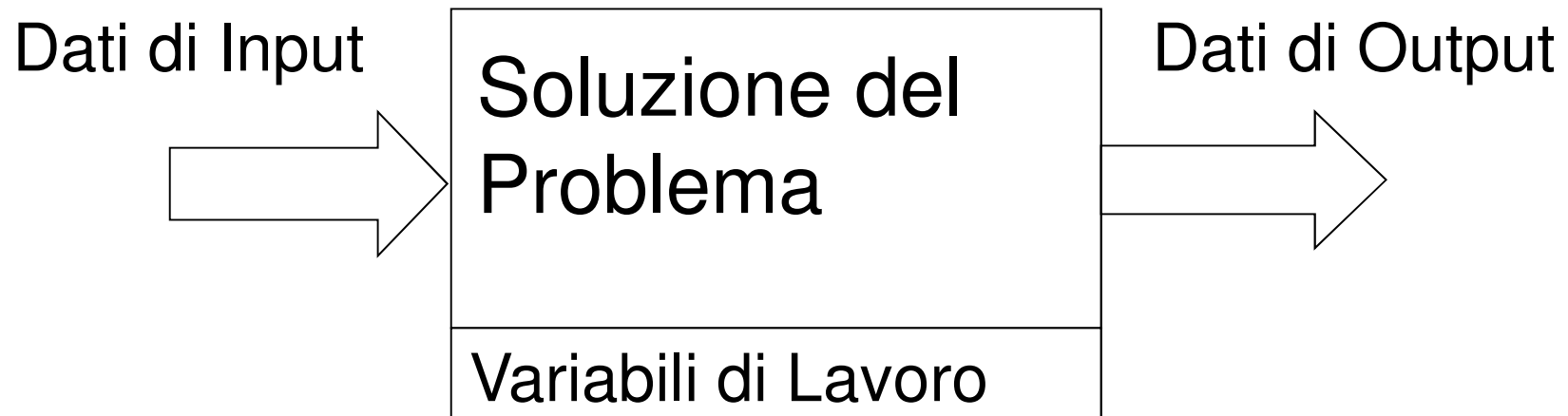
PASSO1: Descrizione del Problema;

PASSO2: Stesura Algoritmo Risolutivo;

PASSO3: Definizione Programma Risolutivo

Passo 1 Descrizione del Problema

- Definire i dati di Input del Problema;
- Definire i dati di Output del Problema;
- Definire le Variabili di Lavoro temporanee o di calcolo.



- Soluzioni Adottate

Passo 2 Stesura Algoritmo Risolutivo

- Definire una serie di Istruzioni da eseguire secondo un bel definito ordine.
- Scomporre eventuali operazioni complesse in una serie di operazioni elementari eseguibili dall'esecutore

Un algoritmo presuppone sempre la presenza di un esecutore (macchina, persona, ecc...)

Es: **Ricetta da Cucina**
 Istruzioni per eseguire la somma di due numeri
 Indicazioni per eseguire un lavoro a maglia
 Istruzioni per il calcolo dell'area di un triangolo

Passo 2 Stesura Algoritmo Risolutivo

L'algoritmo risolutivo viene rappresentato attraverso:

- pseudocodice e/o
- diagramma a blocchi

Es: Calcolare la somma di due numeri (due cifre):

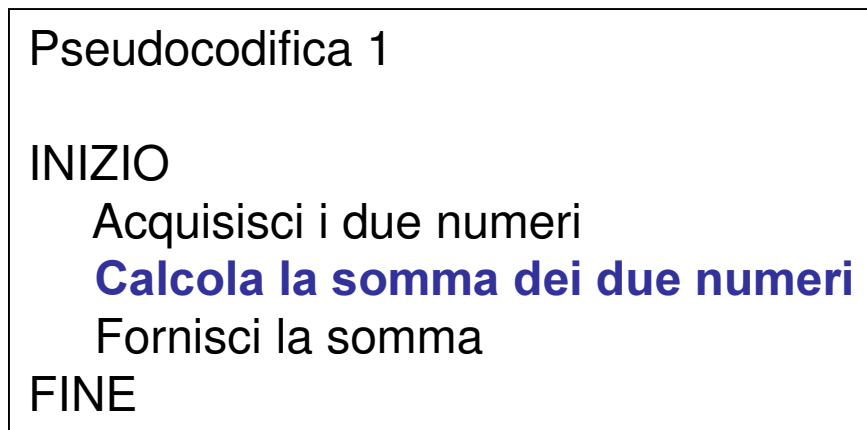
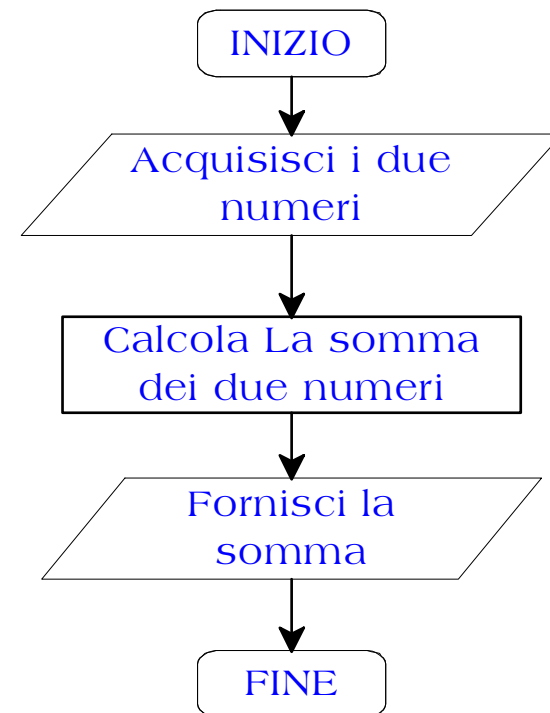


Diagramma a blocchi 1



L'esecutore è capace di fare solo la somma di numeri ad una cifra:

Es: Calcolare la somma di due numeri composti da una cifra:

Pseudocodifica 2

INIZIO

Acquisisci i due numeri

Metti in colonna i numeri

Addiziona le unità

Se la somma è maggiore di 9

**allora Cambia dieci unità con una decina
e riportale nella colonna delle decine**

Scrivi il risultato delle unità

Addiziona le decine

Fornisci la somma

FINE

Diagramma a blocchi 1

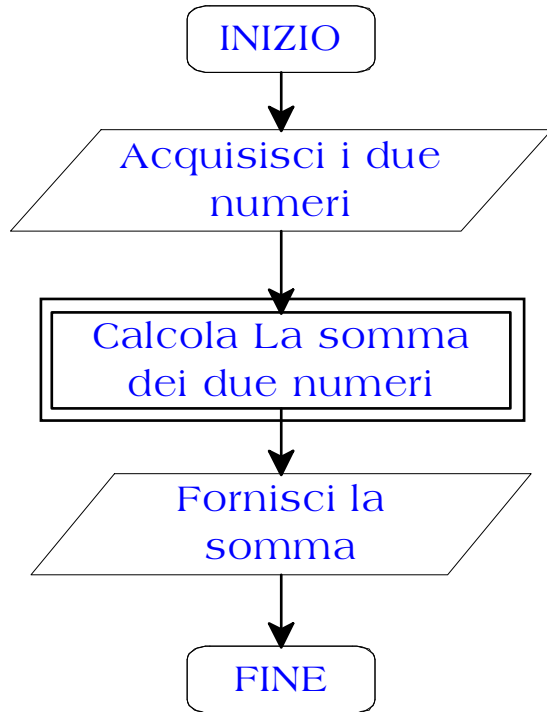
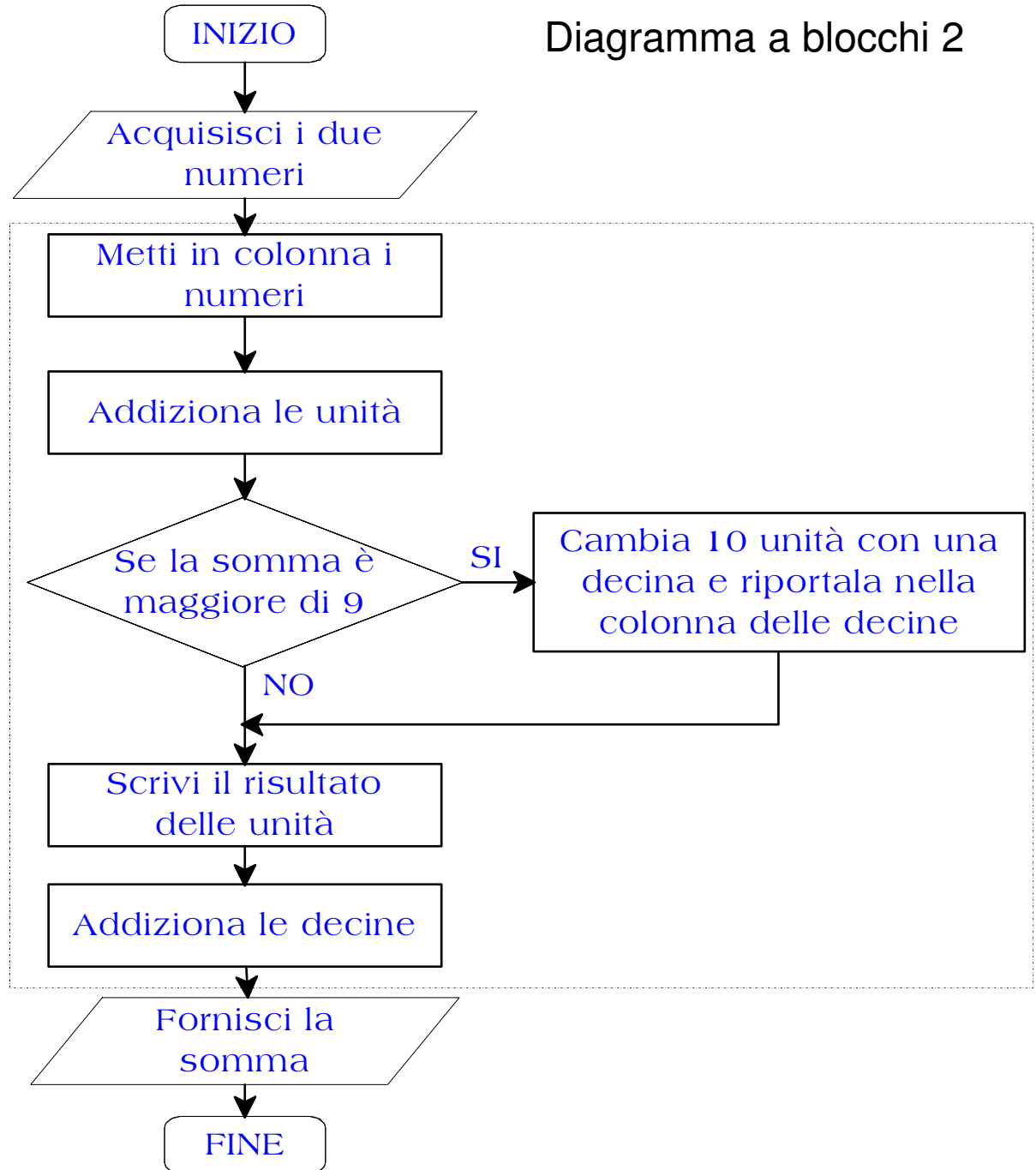


Diagramma a blocchi 2



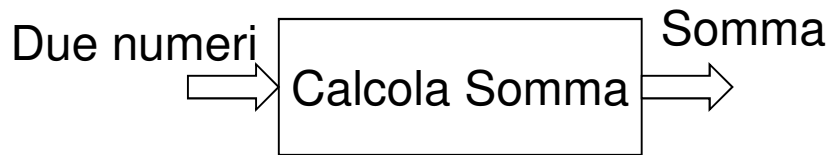
Passo 3 Definizione Programma risolutivo

- Trasformare l'algoritmo risolutivo in un linguaggio comprensibile da parte della macchina (linguaggio di programmazione *.c)
- Tradurre il programma in linguaggio macchina (*.exe)

Considerazioni:

- Se si vuole scrivere un programma allora anche il passo 2 viene sviluppato in vista dell'esecutore macchina.
- Non tutti gli algoritmi risolutivi possono essere trasformati, attraverso semplici passaggi, in un programma eseguibile (Es: Ricetta per la torta)
- La persona che si occupa di scrivere programmi, in linguaggio di programmazione, viene chiamato programmatore.

PASSO 1



Dati Input: a, b
Dati Output: Somma
Variabili interne:

Descrizione della strategia risolutiva:
Si acquisiscono prima i due numeri a e b, si calcola la somma dei due ed infine si fornisce il risultato

PASSO 2

Pseudocodifica

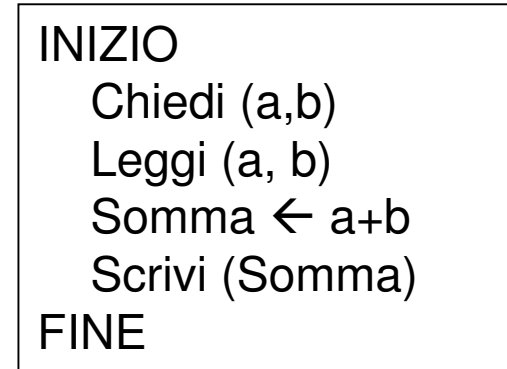
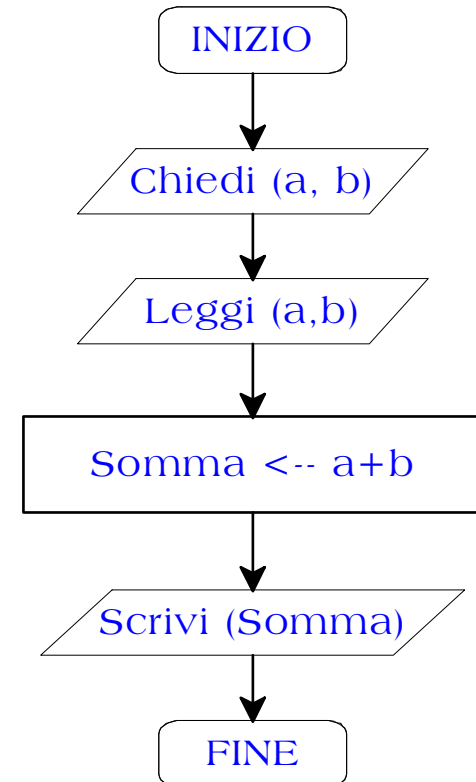


Diagramma a blocchi



PASSO 3

Programma risolutivo

File: somma.c

```
/* Somma.c: somma di due numeri */  
  
#include <stdio.h>  
  
main () {  
    int a, b, Somma;  
    printf("Inserisci il primo numero a: ");  
    scanf("%d",&a);  
    printf("Inserisci il secondo numero b: ");  
    scanf("%d",&b);  
    Somma=a+b;  
    printf("La somma risulta essere: %d \n", Somma);  
    system ("PAUSE");  
}
```

Da riga di comando DOS:

```
gcc somma.c -o somma.exe
```

ALGORITMO

Caratteristiche di un algoritmo:

- 1) **finito** → Composto da un numero finito di istruzioni
- 2) **esaustivo** → Comprendere tutti i possibili casi
- 3) **riproducibile** → Medesimi dati iniziali conducono a medesimi risultati finali

Caratteristiche di ogni singola istruzione:

- 1) **realizzabile** → deve prevedere un'operazione fattibile
(non è realizzabile: la radice di un numero negativo, dividere un numero per lo zero, ecc..)
- 2) **elementare** → L'esecutore deve essere in grado di eseguirla
- 3) **precisa e non ambigua**
- 4) **durata limitata nel tempo**
(non è limitata nel tempo l'istruzione: sommare 2 ad un numero pari fino a farlo diventare dispari)
- 5) **risultato osservabile**
- 6) **deterministica** → compiere una determinata azione e non altre se inseriscono gli stessi dati iniziale.

Figure in gioco in un qualsiasi tipo di problema

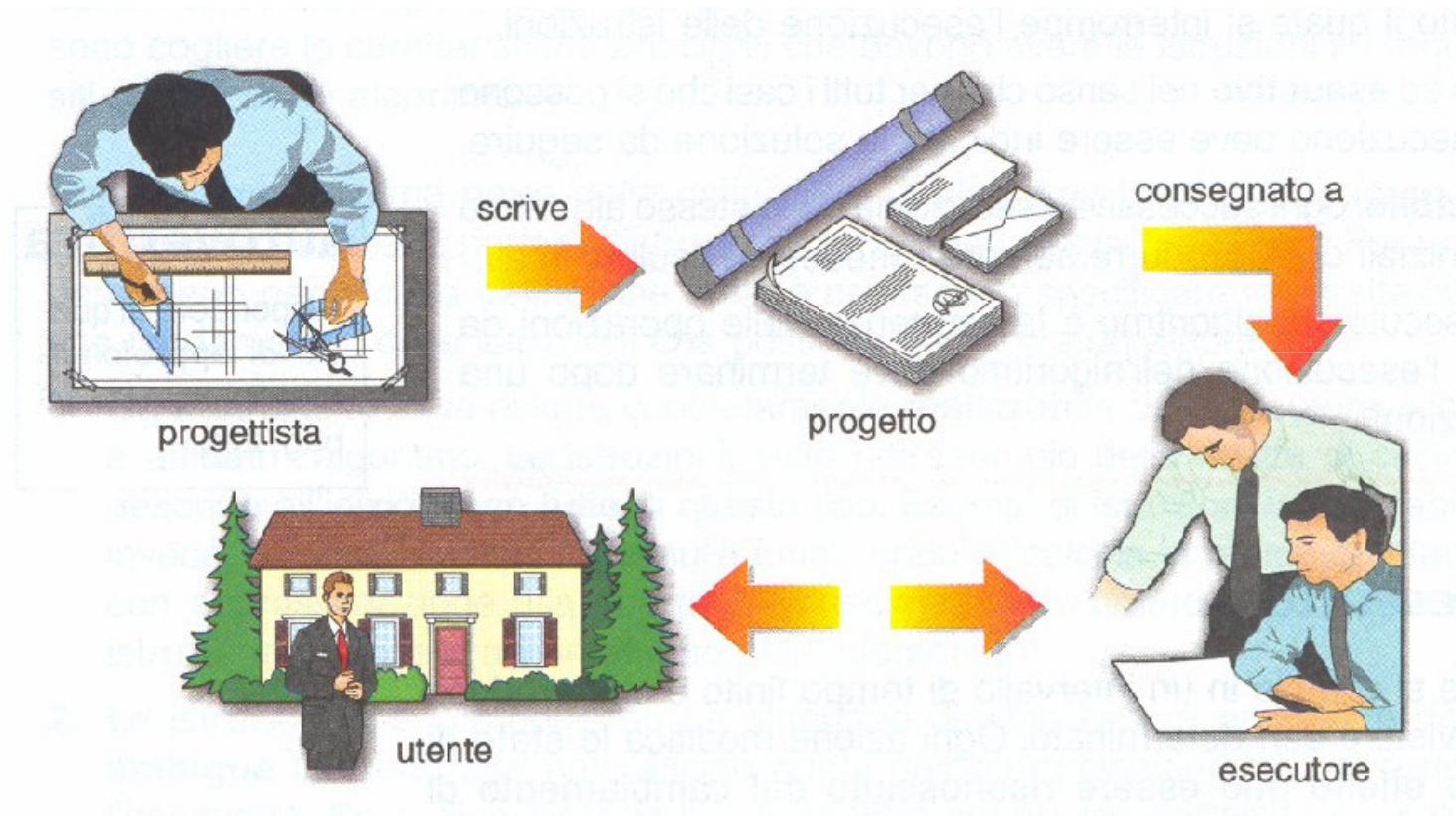
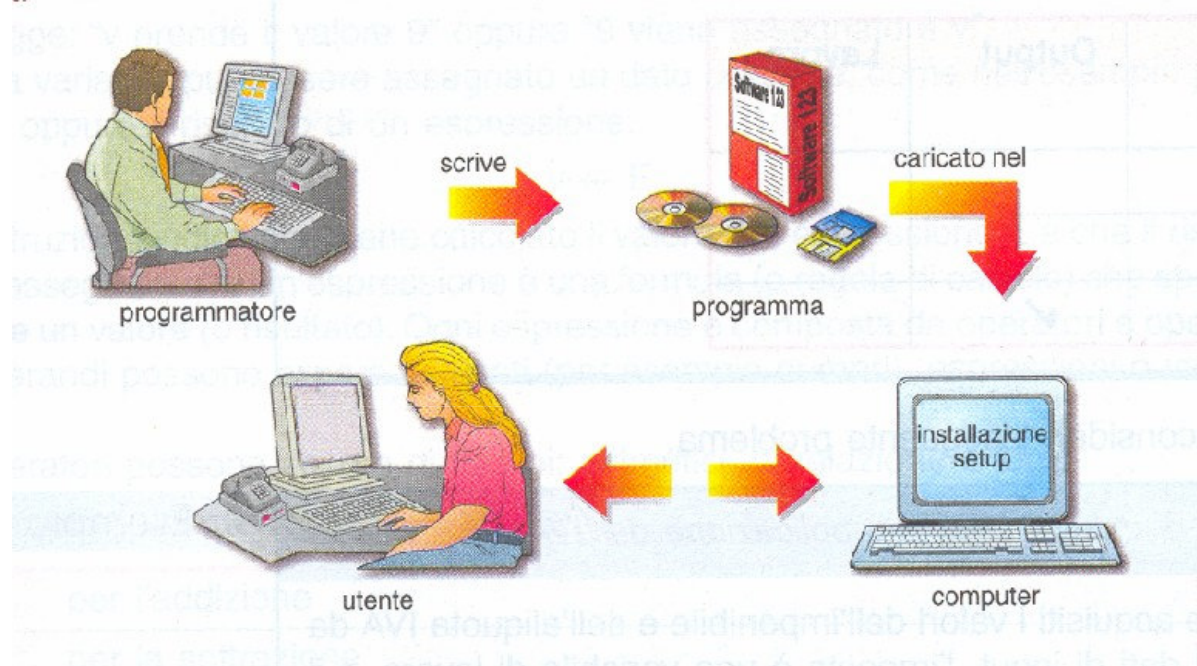


Figure in gioco in un problema da risolvere con l'ausilio di un elaboratore



Programmatore: Colui che organizza prepara e realizza un programma

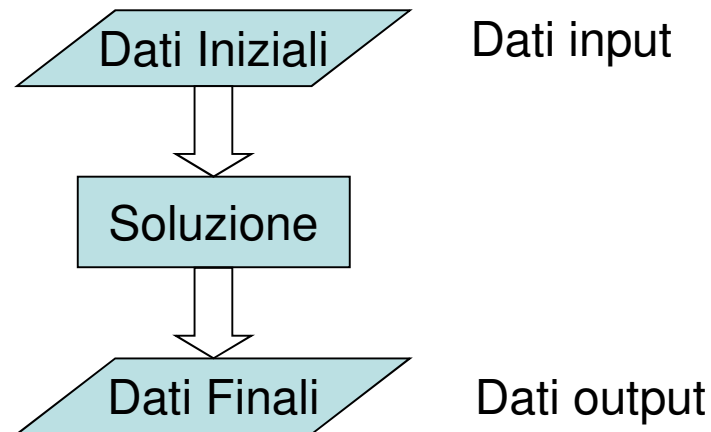
L'esecutore: Figura che provvede ad eseguire le azioni secondo le istruzioni di un algoritmo

Utente: Figura che attiva l'esecuzione dell'algoritmo, che fornisce i dati iniziali del problema e ne utilizza i risultati

Acquisire e comunicare i dati

La risoluzione di un problema implica generalmente tre operazioni fondamentali:

- L'acquisizione dei dati iniziali
- La soluzione del problema
- La comunicazione dei dati o risultati finali



L'esecutore deve quindi essere capace di:

- acquisire i dati in ingresso (variabili)
- svolgere la soluzione
- fornire i dati in uscita (variabili)

Nota: Nello svolgimento della soluzione, l'esecutore può utilizzare altri tipi di variabili dette variabili di lavoro temporanee o di calcolo.

Azioni da prevedere all'interno di un algoritmo:

Azioni di Input:

- leggi
- acquisisci
- accetta
- read
- scanf (*funzione prevista nel linguaggio C*)
- accept
- ecc..

Azioni di Output:

- scrivi
- comunica
- mostra
- write
- printf (*funzione prevista nel linguaggio C*)
- display
- ecc..

Le azioni di elaborazione sui dati vengono fatte utilizzando degli operatori.

Vi sono 4 tipi di operatori:

Operatori di Assegnamento:

- ←
- = (*Operatore usato nel linguaggio C*)
- :=

Operatori Aritmetici:

- +
- -
- *
- /

(Nel linguaggio C sono previsti oltre agli operatori presentati anche altri operatori aritmetici specifici)

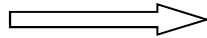
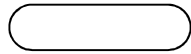
Operatori di Confronto:

- =
- == (*Operatore previsto nel linguaggio C*)
- < (*Operatore previsto nel linguaggio C*)
- <= (*Operatore previsto nel linguaggio C*)
- > (*Operatore previsto nel linguaggio C*)
- >= (*Operatore previsto nel linguaggio C*)
- <>
- != (*Operatore previsto nel linguaggio C*)

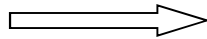
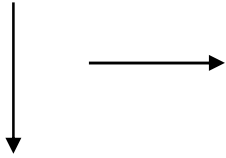
Operatori logici o connettivi logici:

- AND
- OR
- NOT
- && (*Operatore previsto nel linguaggio C "AND"*)
- || (*Operatore previsto nel linguaggio C "OR"*)
- ! (*Operatore previsto nel linguaggio C "NOT"*)

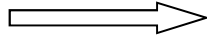
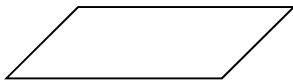
Simboli utilizzati per la stesura di un diagramma a blocchi (flow-chart)



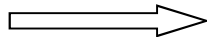
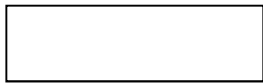
Blocco terminale (Inizio o Fine)



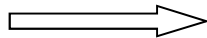
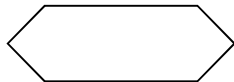
Indicano la sequenza delle operazioni



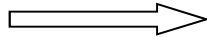
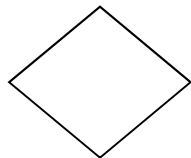
Blocco di Input/output



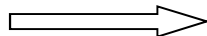
Blocco di Elaborazione



Blocco di Inizializzazione



Blocco Decisionale



Connessione o Richiamo