Algoritmi e Diagrammi di flusso

Prof. Orazio Mirabella

SOMMARIO

- La soluzione di un problema
- Cos'è un algoritmo
- Esempi di problemi algoritmici
- Struttura di selezione
- Struttura di iterazione
- Esempi di algoritmi

Come definire la soluzione di un problema?

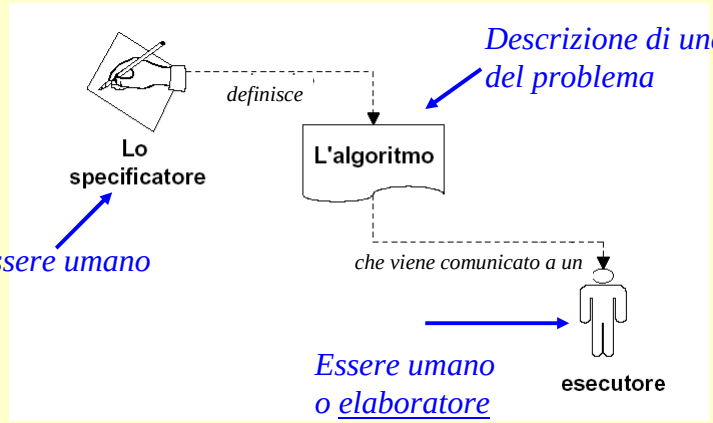
- Dato un problema identificare gli *elementi in ingresso* (ciò che si ha a disposizione), gli *elementi in uscita* (ciò che si vuole produrre risolvendo il problema) e le *azioni* che permettono di risolverlo (procedura di soluzione del problema)

Concetto di algoritmo

- esprimere la procedura di soluzione del problema in un linguaggio comprensibile all'esecutore

Concetto di programma, se l'esecutore e' il calcolatore

La soluzione "eseguibile" di un problema



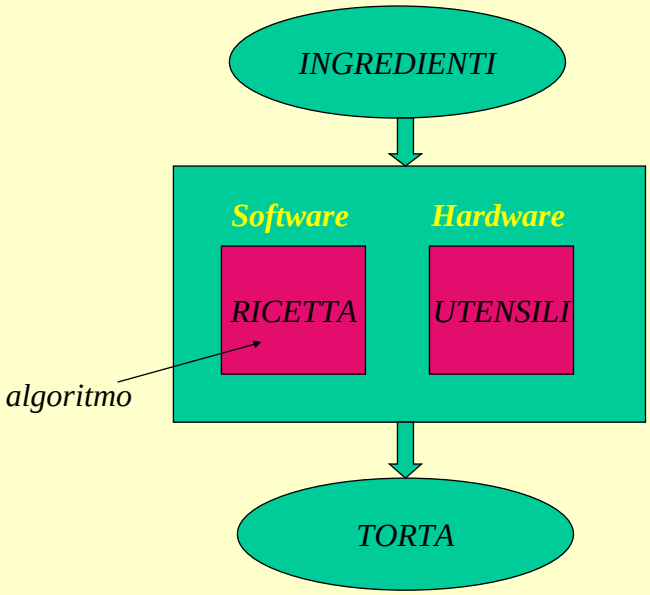
Origine della parola "algoritmo"

- Da Muhammad ibn Muza, detto al-Khowarizmi, un matematico arabo-persiano (825 d.c.) a cui è attribuita l'invenzione di un insieme di precise regole di calcolo per le quattro operazioni.
- In latino Algorismus: una procedura per risolvere un problema matematico in un numero finito di passi che implicano frequenti ripetizioni di un'operazione.

Concetto di algoritmo

- Sequenza di azioni (prescrizioni) per effettuare un compito
- La descrizione di una serie di operazioni la cui esecuzione permette di risolvere un problema
- La *definizione di un algoritmo* e' basata sulla disponibilita' di **INPUT** e implica la loro **TRSFORMAZIONE** per la produzione di un **OUTPUT**
- *Esempi:*
  - ➔*Ricetta per realizzazione di un piatto*
  - ➔*Istruzioni di montaggio di un elettrodomestico*
  - ➔*Prelievo bancomat*

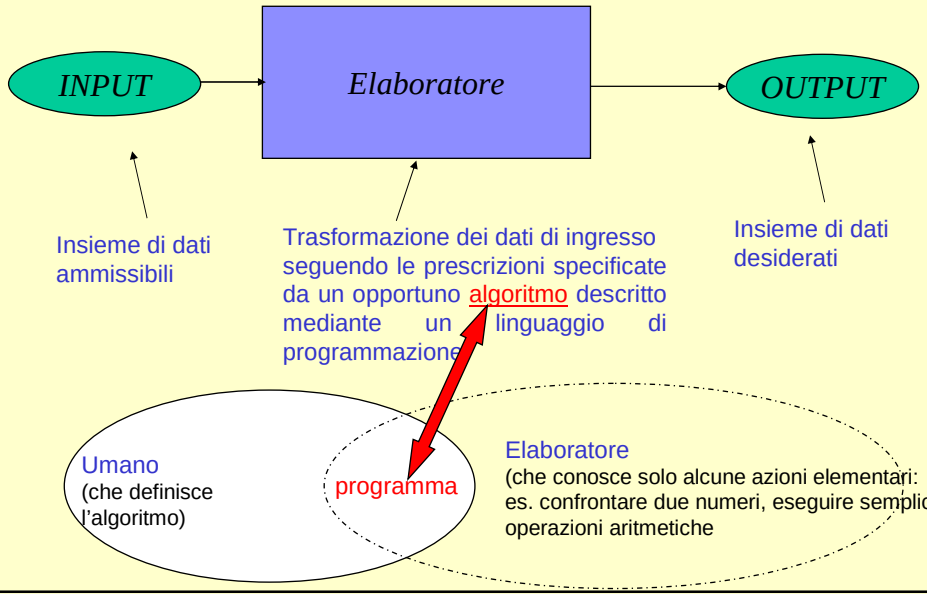
Esempio di algoritmo: una ricetta



Concetto di algoritmo

- Gli algoritmi sono espressi mediante un linguaggio e sono realizzati da un esecutore (uomo o macchina)
- Nell'elaboratore gli algoritmi per la soluzioni di particolari problemi sono espressi mediante un ***linguaggio di programmazione***
- Linguaggio di programmazione = linguaggio formale che l'elaboratore e' in grado di comprendere
- Un programma e' la descrizione di un algoritmo ed e' costituito da una sequenza di istruzioni che il calcolatore e' in grado di comprendere ed eseguire

Concetto di algoritmo



Concetto di algoritmo

- **Algoritmo** = sequenza di passi, definiti con precisione, che portano alla realizzazione di un compito.
- In mathematics and computing, an **algorithm** is a procedure (a *finite* set of *well-defined* instructions) for accomplishing some task which, given *an initial state*, will *terminate* in a defined *end-state*.
- Calcolatori – esecutori di algoritmi

Esempi di problemi algoritmici

Problema 1.

Problema aritmetico: calcolo su un numero fisso di dati

Input: due numeri interi J e K

Output: il numero J+K

Problema 2.

Problema aritmetico: calcolo su un numero variabile di dati

Input: un intero positivo K

Output: la somma di tutti gli interi da 1 a K

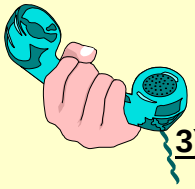
Problema 3.

Problema non aritmetico

Input: una lista L di parole in una lingua fissata

Output: la lista L in ordine alfabetico

Esempio di algoritmo: effettuare una telefonata



1) Sollevare il ricevitore
2) Attendere il segnale di linea
3) Comporre il numero
Ripeti
Componi una cifra
Finchè *numero composto*

sottoalgoritmo

iterazione

4) Attendere risposta
Se la linea è libera allora
Se l'interlocutore risponde allora
5) Condurre la conversazione
6) Deporre il ricevitore
Altrimenti
5) Deporre il ricevitore
Altrimenti (se la linea è occupata)
5) Deporre il ricevitore

selezione

Algoritmo: struttura di selezione

- Permette la prescrizione di un azione sulla base del verificarsi di una condizione

  SE condizione
  ALLORA istruzione1
  ALTRIMENTI istruzione2

- Esempio:

  SE il numero e' occupato
  ALLORA ricomponi numero
  ALTRIMENTI inizia conversazione

Algoritmo: struttura di iterazione

- Permette la prescrizione di ripetizione di un azione sino al momento in cui si verifica una determinata condizione

  RIPETI istruzione
  FINCHE' condizione

- Attenzione: il controllo viene fatto dopo l'esecuzione dell'istruzione; uscita per condizione verificata (vera)

- Esempio:

  RIPETI componi numero sulla tastiera
  FINCHE' utente cercato ha segnale libero

Algoritmo: struttura di iterazione

- Altre espressioni di struttura di iterazione

  RIPETI **istruzione**

  MENTRE **condizione**

- Esempio:

  RIPETI **componi numero sulla tastiera**

  MENTRE **utente cercato ha segnale occupato**

Algoritmi: ancora esempi …

- Definiamo degli algoritmi per i seguenti problemi:
  1. *Trovare il massimo fra 2 numeri interi positivi x e y*
  2. *Trovare il massimo fra 3 numeri interi positivi x, y e z*
  3. *Trovare il massimo fra N numeri interi positivi*
- Assumiamo le seguenti operazioni elementari:
  - ➔ Somma (+), sottrazione (-)
  - ➔ Stabilire se un numero e' maggiore o minore di 0
  - ➔ Leggere/scrivere un numero dallo schermo
- … dove ragionevole:
  - ➔ definiamo dei sottoalgoritmi e utilizziamo le strutture di controllo

Il massimo (maggiore) fra 2 numeri interi x e y

- Algoritmo **max**
  1. Leggi i valori di x e y dall'esterno
  2. Calcola la differenza d fra x e y (d=x-y)
  3. Se d e' maggiore di 0
     Allora stampa "il massimo e' …" seguito dal
     valore di x
     Altrimenti stampa "il massimo e' …" seguito
     dal valore di y
  4. Termina l'esecuzione

Il massimo (maggiore) fra 2 numeri interi x e y

- *x*, *y*, *d* sono le '**variabili**', cioè i *contenitori* per i dati coinvolti nell'elaborazione. Ogni variabile ha un nome (x, y…) e un valore: il dato contenuto o *memorizzato* in essa
- Dobbiamo specificare esplicitamente come e quando l'esecuzione termina!

Il massimo fra 3 numeri interi x, y e z

- Possiamo sfruttare l'algoritmo max come ***"sottoalgoritmo"***

- Algoritmo **max_3**
  1. Leggi i valori di x, y e z dall'esterno
  2. Se x e' maggiore di y (usando l'algoritmo max)
     Allora trova il massimo fra x e z (con max) e termina
     Altrimenti trova il massimo fra y e z (con max) e termina

Il massimo fra N numeri interi

- Possiamo ancora sfruttare l'algoritmo max come sottoalgoritmo

- **Idea**: trovare prima il maggiore fra i primi due numeri, poi confrontare il risultato con il terzo, poi con il quarto, ecc …

- Utilizziamo la struttura di controllo iterativa *ripeti …finchè* per effettuare le operazioni di max su tutti i numeri in ingresso

Il massimo fra N numeri interi

- Algoritmo **max_N**

1. Leggi il valore N dall'esterno
2. Leggi i primi due numeri x e y dall'esterno
3. Trova il massimo m fra i primi due numeri (con max)
4. Ripeti (fintanto che esaminati meno di N numeri)
   a. Leggi un nuovo numero x
   b. Trova il massimo fra m e x usando l'algoritmo max
   c. Assegna il valore del massimo a m

   Finche' letti N numeri
5. Stampa "il massimo e' …" seguito dal valore di m
6. Termina l'esecuzione

SOMMARIO

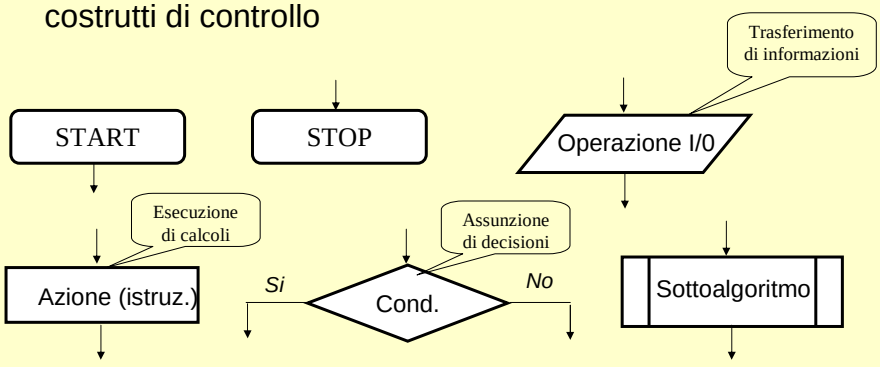
- Rappresentazione di un algoritmo
- Caratteristiche di un algoritmo
- Concetto di programma

Rappresentazione di un algoritmo: Diagrammi di flusso

- Sono grafici che permettono di esprimere un algoritmo in modo schematico e intuitivo
- Per rappresentare un algoritmo occorre rappresentare:
  - ➔Passi necessari
  - ➔Loro corretta sequenza
- I diagrammi di flusso sono una descrizione piu' efficace e meno ambigua di una descrizione a parole

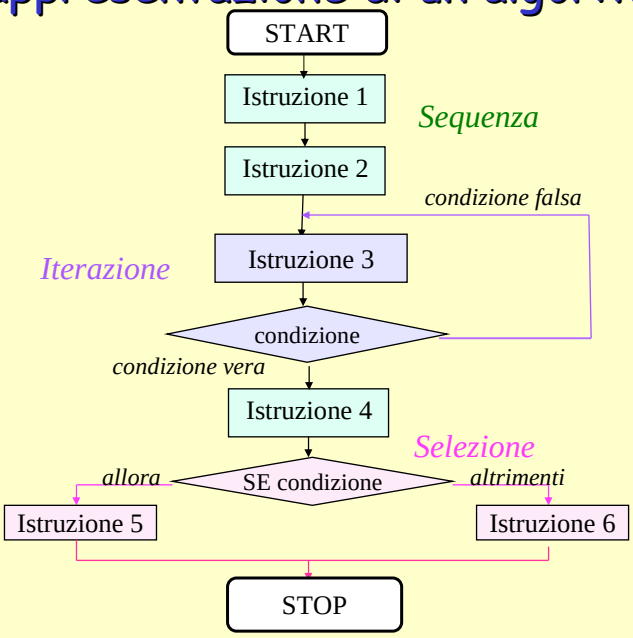
Diagrammi di flusso

- Si costruiscono a partire da un certo numero di ***blocchi base*** che rappresentano le operazioni elementari ed i costrutti di controllo



- I blocchi base vengono collegati tramite "frecce" che collegano un'azione alla successiva all'interno dell'algoritmo

Rappresentazione di un algoritmo

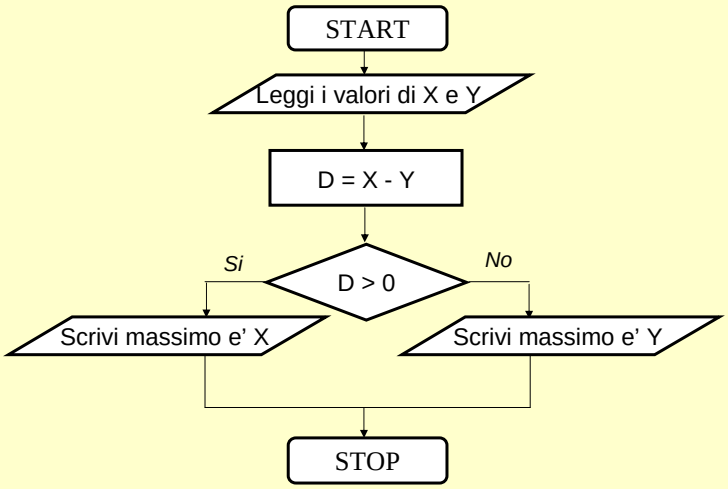


Diagrammi di flusso

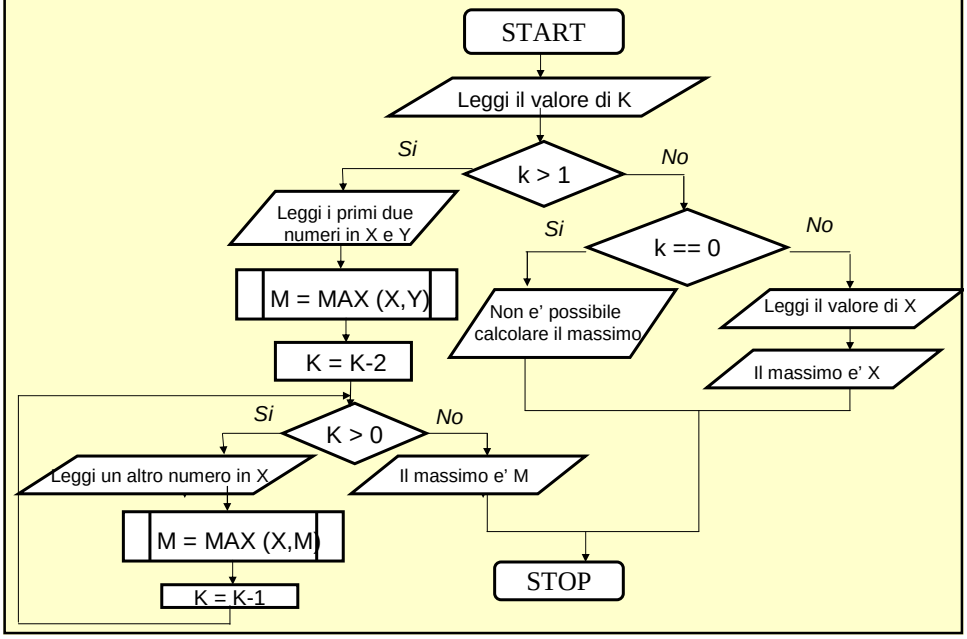
- Realizzare i diagrammi di flusso per i seguenti algoritmi:

1. *Trovare il massimo fra 2 numeri interi positivi x e y*
2. *Trovare il massimo fra 3 numeri interi positivi x, y e z*
3. *Trovare il massimo fra N numeri interi positivi*

Massimo fra due numeri interi positivi X e Y



Massimo fra N numeri interi positivi



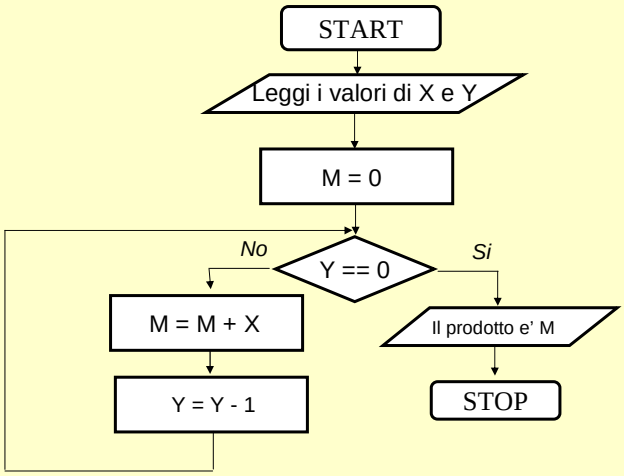
Esercizio

- Si chiede di realizzare il diagramma di flusso dell'algoritmo che risolve il prodotto di due numeri col metodo delle addizioni successive.

Soluzione

- Si chiami X il valore del moltiplicando ed Y il valore del moltiplicatore e sia M il risultato.
  - ➔ Si inizializza il valore M a 0
  - ➔ Si ripetono le seguenti operazioni fintanto che Y e' diverso da 0:
    - ✓ Si sommi il valore di X al valore di M e si chiami ancora il risultato M (M=M+X)
    - ✓ Si sottragga 1 dal valore di Y, e si chiami Y ancora il risultato (Y=Y-1)
  - ➔ Sia M il risultato del prodotto

Prodotto di due numeri X e Y col metodo delle addizioni successive



Esercizio

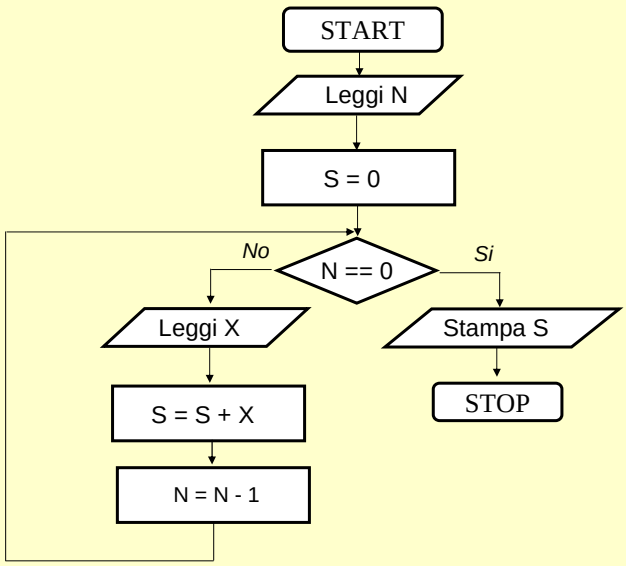
- Si chiede di rappresentare con un diagramma di flusso l'algoritmo che effettua la somma di N numeri interi.

Soluzione

- Problema: leggere K, e quindi calcolare la somma di K valori letti dall'input
- Devo memorizzare K, la somma, e i valori letti *V1,V2,…,VK*
- Poiche uso ogni *Vi* una sola volta, mi bastano 3 variabili: *K, x ed S*
- ***x*** manterrà il valore *Vi* corrente
- Algoritmo informale:

leggo K,
inizializzo S con 0
per tutti i valori da V1 a VK,
leggo(x)
sommo x a S (S=S+x)
stampo S

Somma di N numeri interi



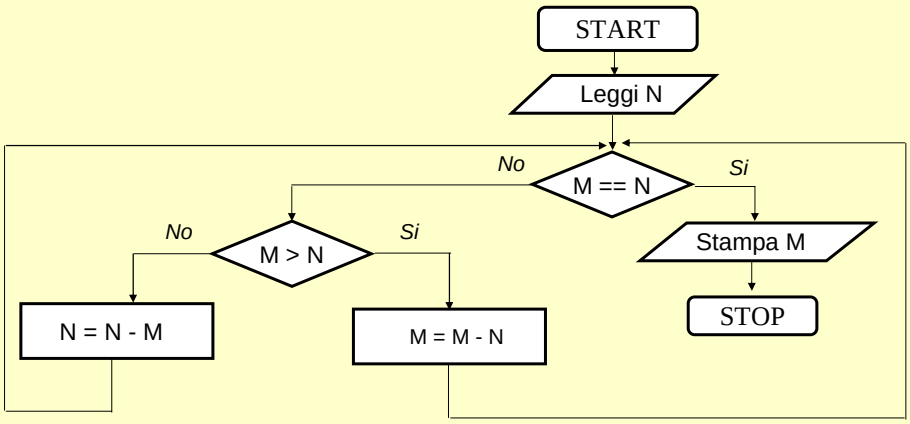
Esercizio

- Calcolare il massimo comune divisore tra due numeri interi letti da input, utilizzando l'algoritmo di *Euclide*:
  - ➔mcd(m,n)=m=n se n=M
  - ➔mcd(m,n)=mcd(m-n,n) se m>n
  - ➔mcd(m,n)=mcd(m,n-m) se n>m

Algoritmo di *Euclide*

- Leggo m and n
- (*) Fino a che m diverso da n,
  - ➔se m>n allora sottraggo n ad m
  - ➔se n>m sottraggo m ad n
  - ➔torno a (*)
- Quando m=n stampo, ad es, n

Massimo comune divisore tra due numeri interi M ed N letti da input



Esercizio

- Calcolare il minimo comune multiplo tra due numeri interi letti da input, utilizzando l'algoritmo di *Euclide*:

  mcm(*n*,*m*) = *n* * *m* / MCD(*n*,*m*)

Caratteristiche di un algoritmo

- Un insieme finito di prescrizioni (istruzioni) che da' luogo ad una sequenza finita di operazioni
  - ➔ Termina dopo un numero finito di passi
  - ➔ Opera su input ammissibili per il problema considerato
  - ➔ Produce output desiderati che abbiamo una relazione con gli input specificati (siano funzione degli input)
  - ➔ Tutte le operazioni dell'algoritmo devono essere elementari (comprensibili all'esecutore) e poter essere eseguite in un tempo finito
- Osservazione: il concetto di azione elementare e' relativo all'esecutore!

Concetto di algoritmo: requisiti fondamentali

- **Univocita'**
  - ➔ La descrizione è effettuata in un linguaggio comprensibile all'esecutore e mediante un insieme finito di operazioni elementari, note all'esecutore
- **Completezza**
  - ➔ Tiene conto di tutte le possibili condizioni che si possono verificare durante la sua esecuzione
- **Finitezza**
  - ➔ In un numero finito di passi il compito viene portato a termine
- **Determinismo**
  - ➔ Risultati non dipendenti dalla esecuzione
- **Efficienza**
  - ➔ Minimo di operazioni

Concetto di programma

- Un linguaggio di programmazione consente di stabilire un dialogo (di comunicare) con l'elaboratore.
- Un linguaggio di programmazione e' un linguaggio comprensibile ad un elaboratore, per mezzo del quale si possono esprimere gli algoritmi.
- Un programma e'la rappresentazione formale di un algoritmo mediante un lingiaggio di programmazione
- Le istruzioni specificate in un programma possono richiedere dati comunicati dall'utente (dati di input - ingresso) e producono altri dati (dati di output - uscita)

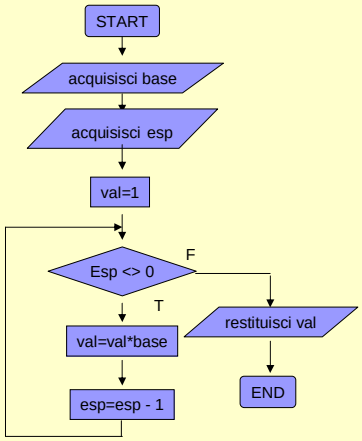
Concetto di programma

- Il linguaggio macchina e'il linguaggio che l'elaboratore e' in grado di comprendere
- Linguaggi di programmazione di alto livello:
  - ➔ Piu' facili da usare, in quanto piu' comprensibili
  - ➔ I programmi scritti in un linguaggio di alto livello sono tradotti da altri programmi in linguaggio macchina
- Esempi di linguaggi di alto livello:
  - ➔ ForTran (Formula Translator)
  - ➔ Cobol (Common business oriented language)
  - ➔ Pascal
  - ➔ C
  - ➔ Java
  - ➔ …

Esempio di algoritmi: calcolo dell'e-esima potenza

Il seguente schema realizza il percorso risolutivo del problema relativo al calcolo della potenza di un numero e la relativa traduzione in pseudocodifica.

Il problema si riferisce a numero (detto **base**) e valore della potenza (detto **esp**) scelti da un generico utente.




```
START

 acquisici base

acquisici esp

val = 1


 ESEGUI MENTRE  esp<>0
     val = val * base
     esp = esp - 1
   RIPETI
   restituisci val

END
```