Programma del corso

- *Introduzione agli algoritmi*
- *Rappresentazione delle Informazioni*
- *Architettura del calcolatore*
- *Reti di Calcolatori*
- ***Elementi di Programmazione***

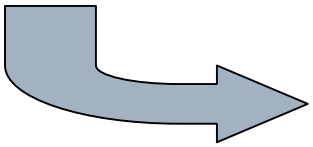
Algoritmi e programmi

- **Algoritmo**

  Sequenza finita di passi che risolve in tempo finito un problema.

- **Codifica**

  Fase di scrittura di un algoritmo attraverso un insieme ordinato di frasi (“istruzioni”), scritte in un qualche linguaggio di programmazione, che specificano le azioni da compiere.



Programma

Testo scritto in accordo con la sintassi e la semantica di un linguaggio di programmazione.

Cenni di programmazione

- Il computer esegue programmi
- Un programma eseguibile dal computer è una sequenza di istruzioni comprensibili da quel computer
- Usando sequenze diverse di istruzioni, e dati diversi, possiamo far fare al computer le cose più disparate

Cenni di programmazione

- Il linguaggio di programmazione che viene compreso da un calcolatore è il linguaggio macchina (**assembler**)

- Però scrivere programmi in assembler è scomodo, perché il linguaggio è molto distante da quello umano.

- Inoltre, un programma in assembler gira solo su un tipo di cpu
  - sarebbe comodo poter usare lo stesso programma su cpu diverse senza doverlo riscrivere ogni volta

Linguaggi di programmazione ad alto livello

- I linguaggi di programmazione ad alto livello permettono di scrivere programmi con una notazione adatta agli esseri umani, e in alcuni casi molto intuitiva.
- Usando degli opportuni traduttori (compilatori ed interpreti) lo stesso programma può essere usato su macchine diverse
- Fortran, Cobol, Pascal, Ada, C, C++, Java, Lisp, ML, Prolog,...

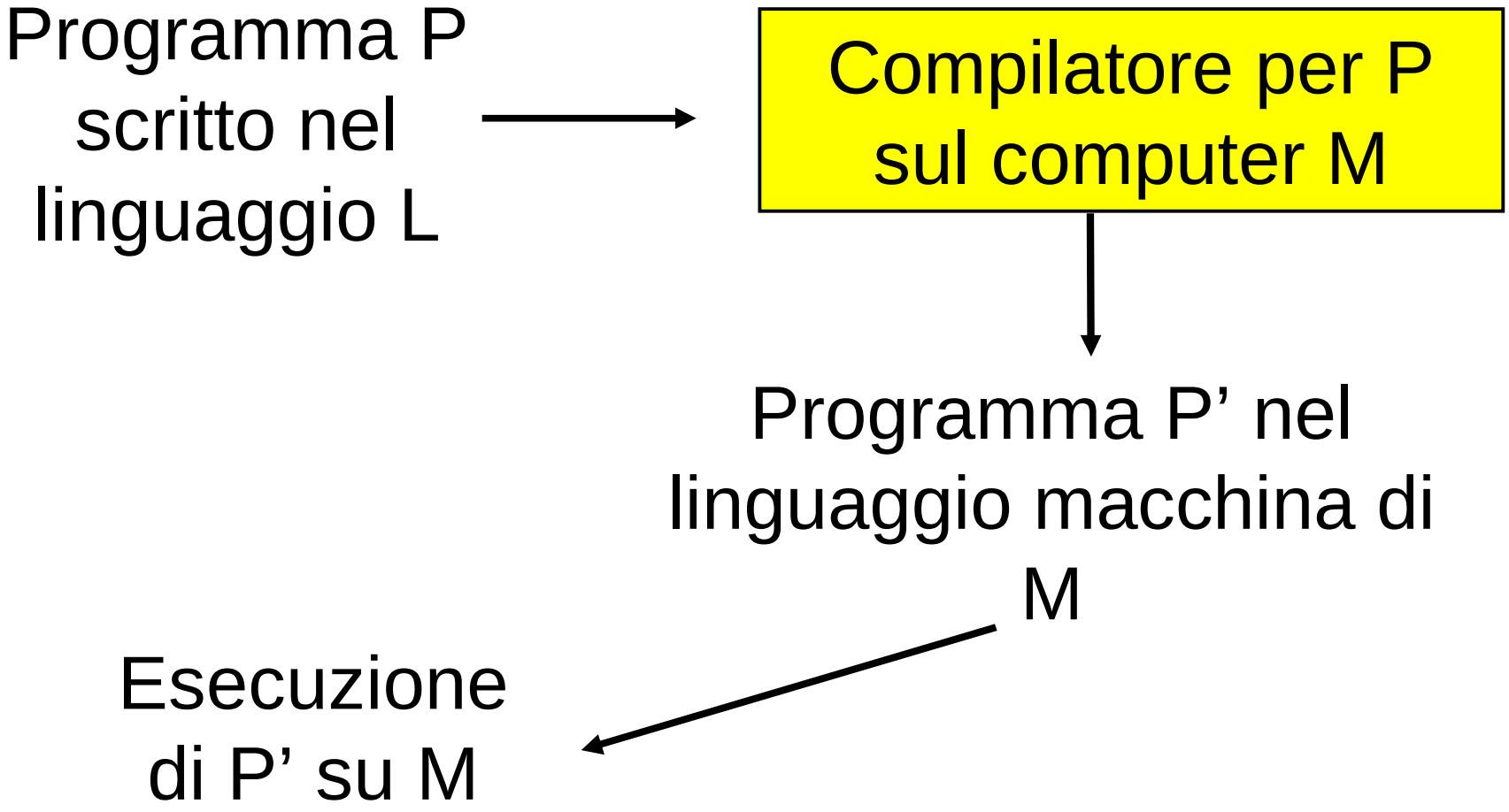
Linguaggi di programmazione ad alto livello

- Un linguaggio artificiale per scrivere programmi
- Un linguaggio preciso e rigoroso. Occorre rispettare:
  - la sintassi (regole di composizione dei simboli del linguaggio per ricavarne le istruzioni)
  - la semantica (significato delle istruzioni)
- Il computer è meno tollerante agli “errori” di un umano

Il compilatore

- Un programma scritto in un linguaggio ad alto livello è detto **programma sorgente**.
- Per essere eseguito su un computer, va tradotto nel linguaggio macchina del computer.
- Il **compilatore** è un programma che esegue la traduzione, producendo il **programma oggetto**, ossia una sequenza di istruzioni macchina
- Il compilatore segnala anche eventuali errori di sintassi nella scrittura del programma sorgente

Il compilatore



L’interprete

- In alternativa alla compilazione, un programma sorgente può essere **interpretato**.
- Un **interprete** è un programma che non produce alcun programma oggetto, ma legge il ogni istruzione del programma sorgente e genera le istruzioni macchina corrispondenti, che vengono passate all’hardware per l’esecuzione.

Compilatori vs interpreti

- In un programma compilato, la traduzione avviene una sola volta, e poi il programma oggetto può essere eseguito quante volte si vuole
- In un programma interpretato, la traduzione avviene tutte le volte che si esegue il programma
- Molti linguaggi permettono entrambe le scelte
- Attualmente, i computer sono così potenti che anche la compilazione di lunghi programmi non richiede molto tempo.

Dall'algoritmo al programma

Il concetto di algoritmo

- Un algoritmo è una sequenza di passi necessari per risolvere un problema o eseguire una computazione
- In alcuni casi, lo stesso problema/computazione può essere risolto in modi diversi, ai cui corrispondono diversi algoritmi
- Un programma non è altro che la descrizione di un algoritmo scritta nel linguaggio di programmazione scelto.

Esempio di algoritmo:

ricerca di una voce nell'elenco telefonico

- Sia **cognome** la voce da cercare
- Sia **E** l'elenco da "sfogliare"

Ripeti

- se E è vuoto allora cognome non esiste, termina.
- dividi a metà l'elenco E
- esamina tutte le voci della pagina che hai di fronte
- se trovi cognome allora annota il numero e termina.
- se cognome < prima voce pagina
  - considera come E la prima metà dell'elenco

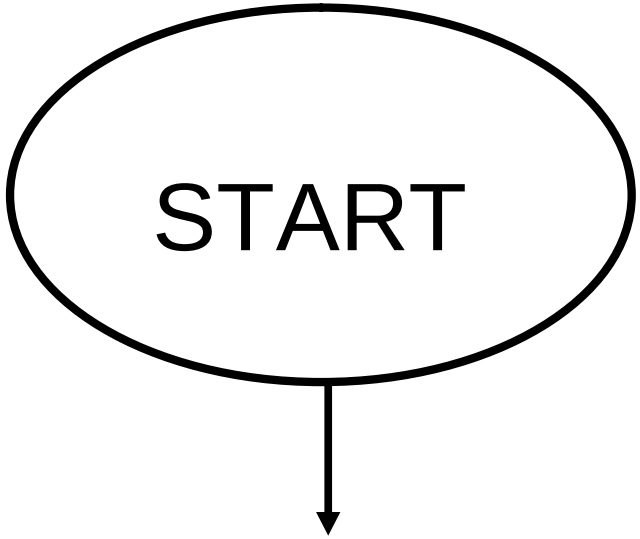
  altrimenti
  - considera come E la seconda metà dell'elenco

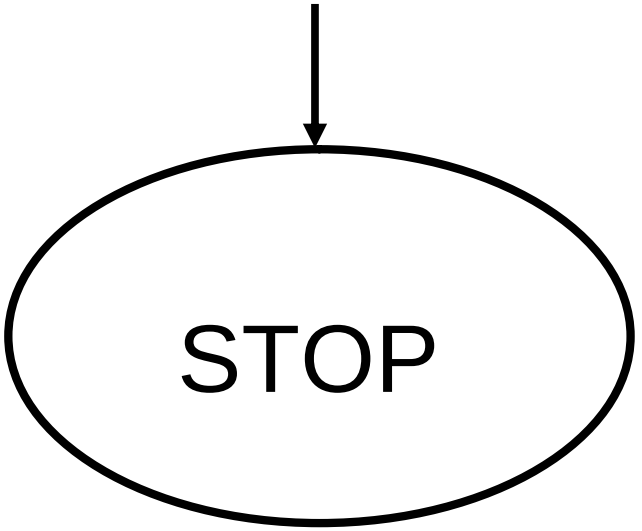
Fine

Diagrammi di flusso

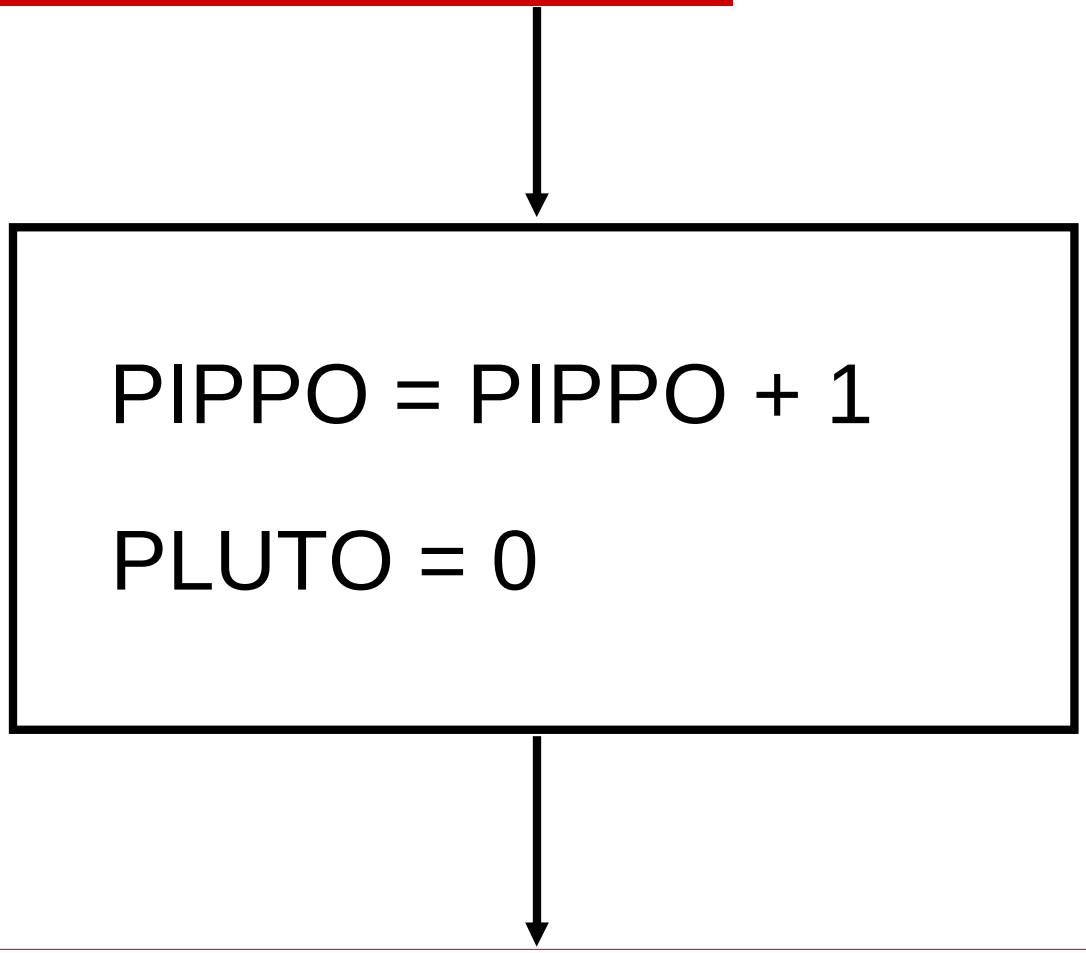
- Notazione grafica usata per descrivere in modo intuitivo le azioni di cui è fatto un algoritmo.
- Viene usata per descrivere i passi salienti di un algoritmo, senza doversi preoccupare dei dettagli sintattici del programma corrispondente
- Una volta che l'algoritmo è stato descritto con un diagramma di flusso, deve però essere trasformato nel programma corrispondente.
- Ogni azione è rappresentata da un **blocco**

Blocchi di flusso: inizio e fine algoritmo

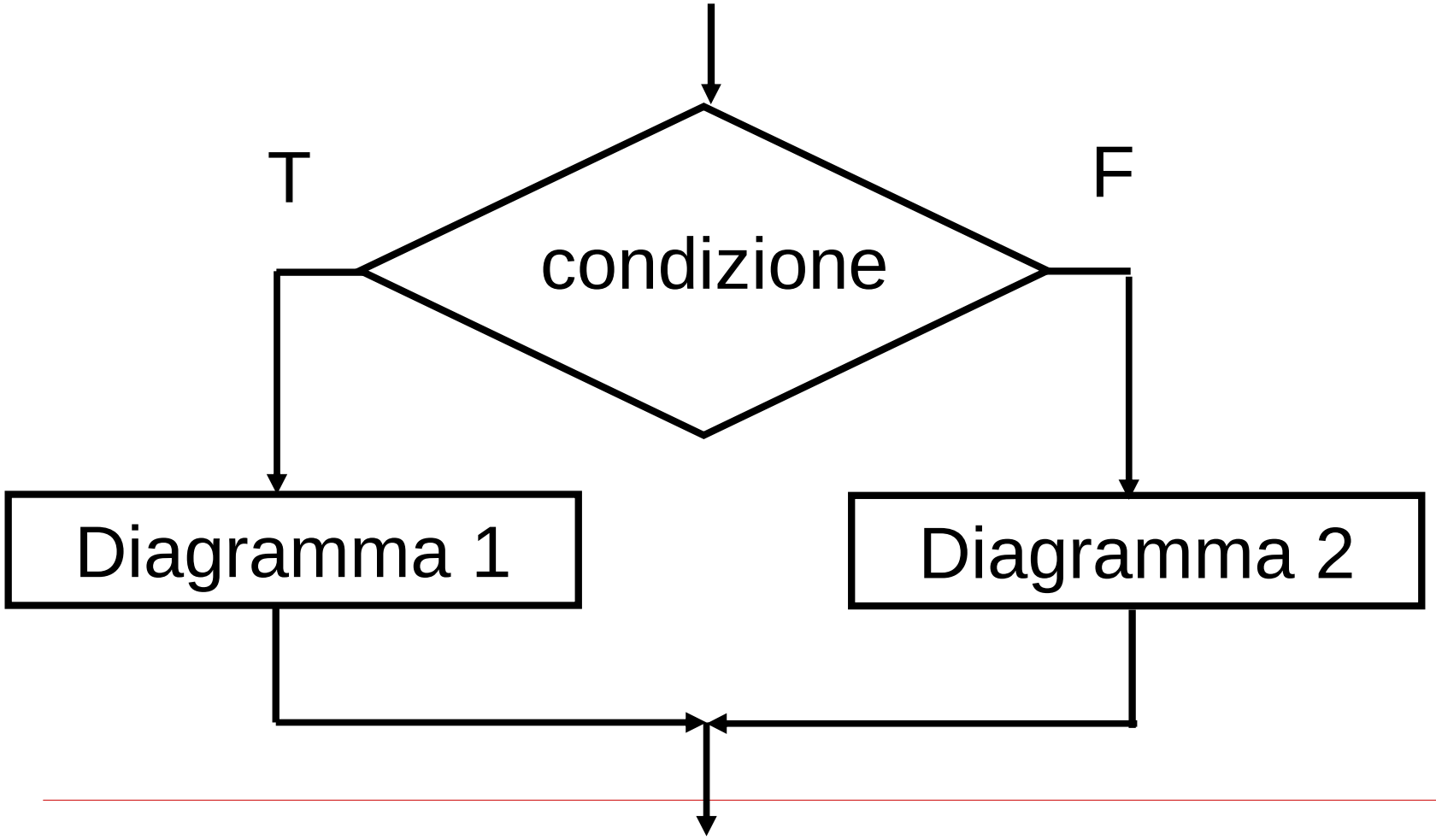




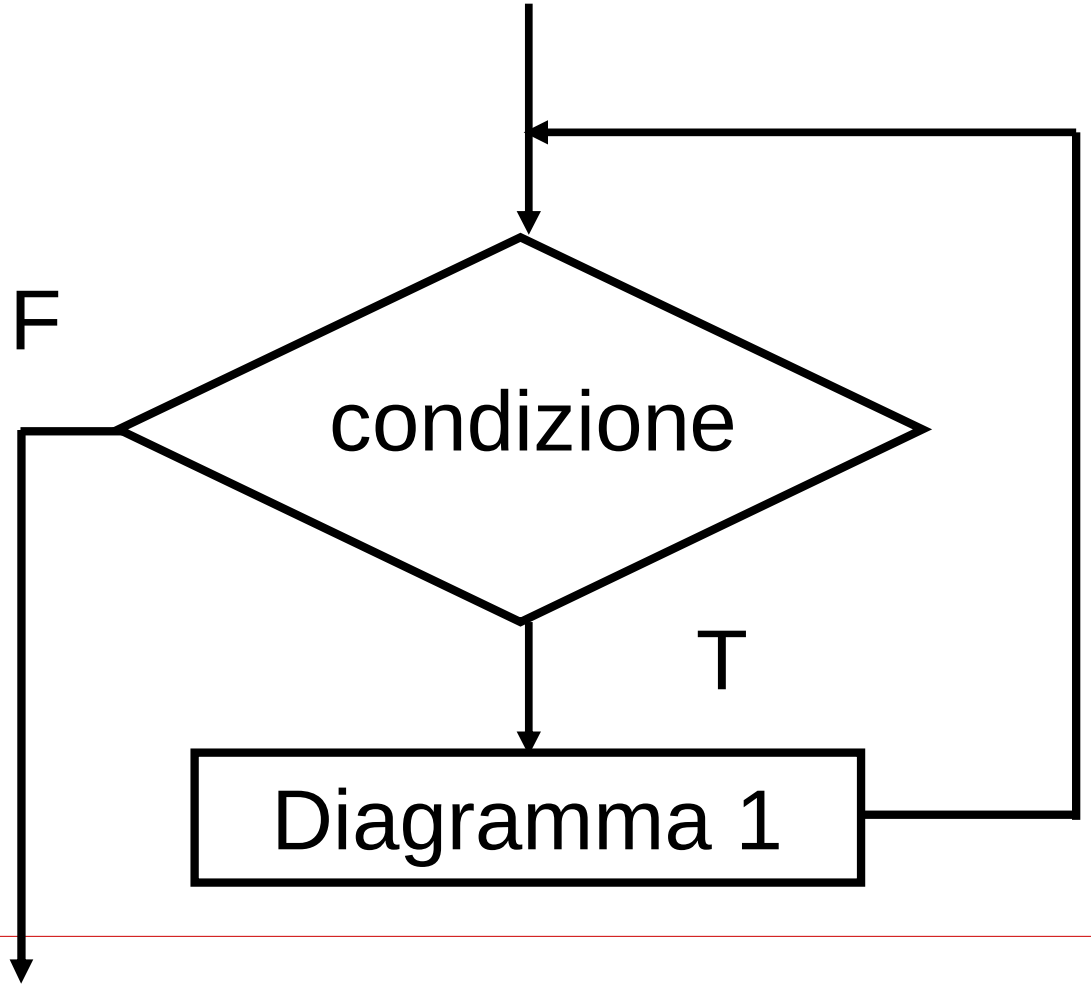
Blocchi di flusso: una o più azioni elementari



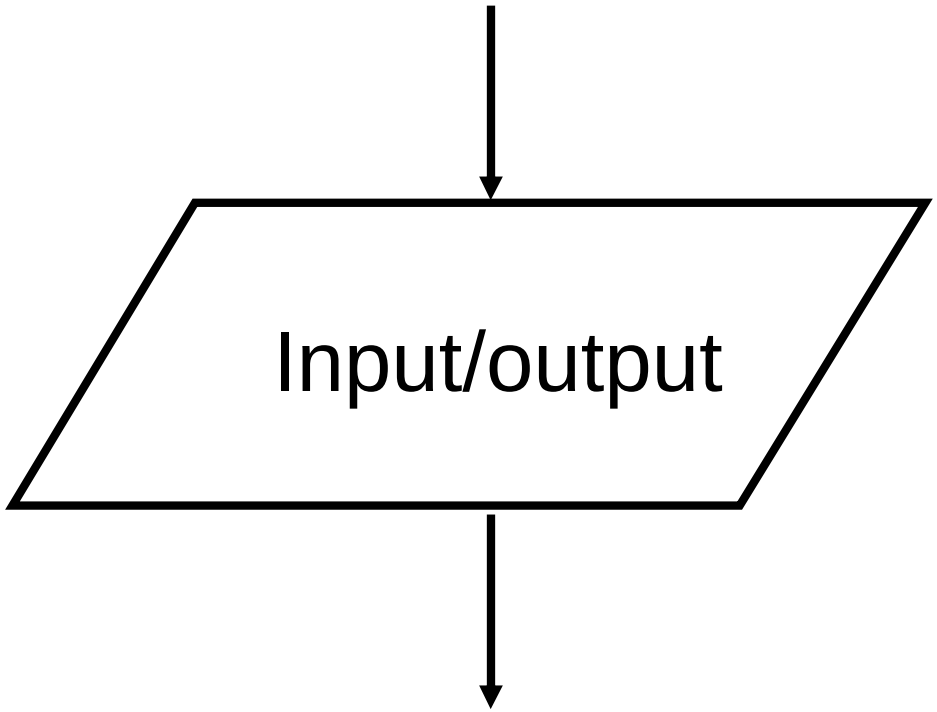
Blocchi di flusso: Blocco condizionale



Blocchi di flusso: Blocco di ripetizione



Blocchi di flusso: Input/Output



Il concetto di variabile

- Per eseguire una qualsiasi computazione, abbiamo bisogno di poter immagazzinare i risultati temporanei e finali della computazione stessa.
- Ogni linguaggio ad alto livello mette a disposizione le variabili: “contenitori” in cui immagazzinare i dati della computazione
- Concettualmente, le variabili sono come pezzi di carta su cui si possono annotare/modificare i valori di un calcolo che si sta facendo

Il concetto di variabile

- Ogni variabile ha un nome mnemonico, che si usa nel programma per riferirsi alla var. stessa.
- Una variabile contiene un valore che può essere modificato a piacimento
- Durante l'esecuzione di un programma, il sistema operativo mantiene una associazione tra il nome di ogni var. e l'indirizzo della cella di memoria in cui è memorizzato il suo valore
- Quindi una variabile è semplicemente una astrazione della cella di memoria fisica.

Il concetto di variabile

- Quando si scrive un programma è necessario dichiarare quali variabili vogliamo usare.
- Le variabili possono essere di tipo diverso, per indicare che le usiamo per memorizzare dati di tipo diverso:
  - Variabile **LETTERA**, tipo: carattere;
  - Variabile **SOMMA**, tipo: intero;

L’importanza delle variabili

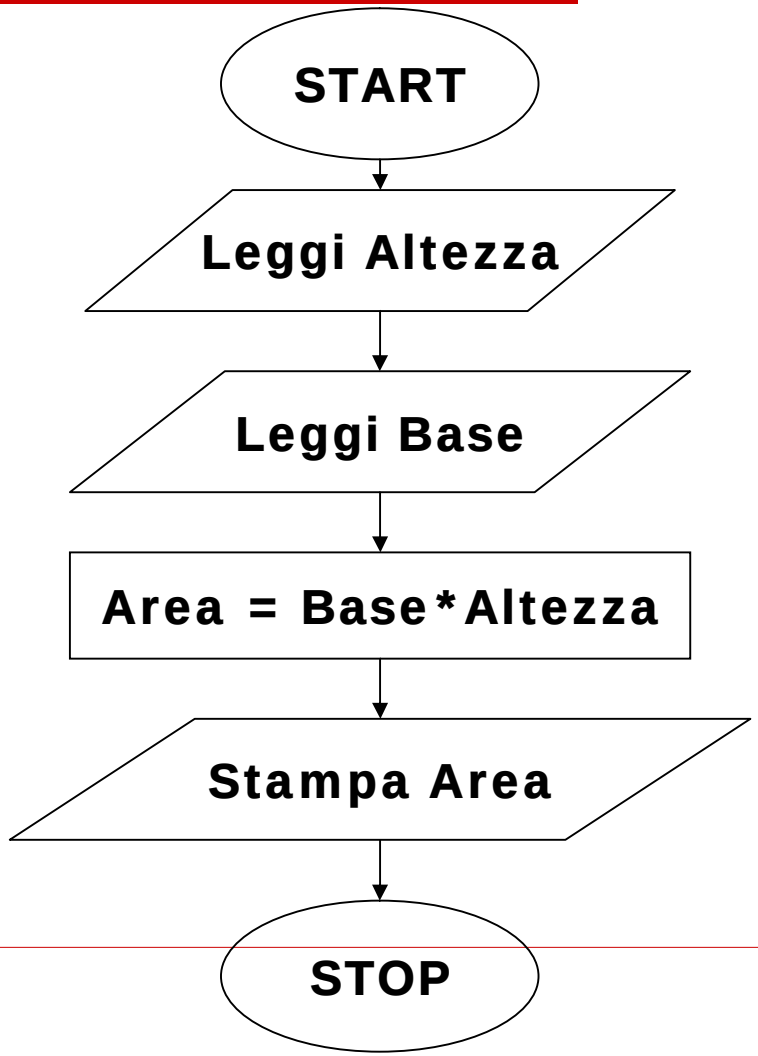
- Le variabili sono lo strumento fondamentale per assicurare la flessibilità dei programmi.
- Lo stesso programma, eseguito con variabili di valore diverso da risultati diversi. Lo stesso programma si adatta cioè alle esigenze del momento, senza dover essere riscritto

Esempi di algoritmi

Calcolo dell'area di un rettangolo

- Leggi da input l'altezza (H)
- Leggi da input la base (B)
- Calcola l'area
- Dai in output il risultato

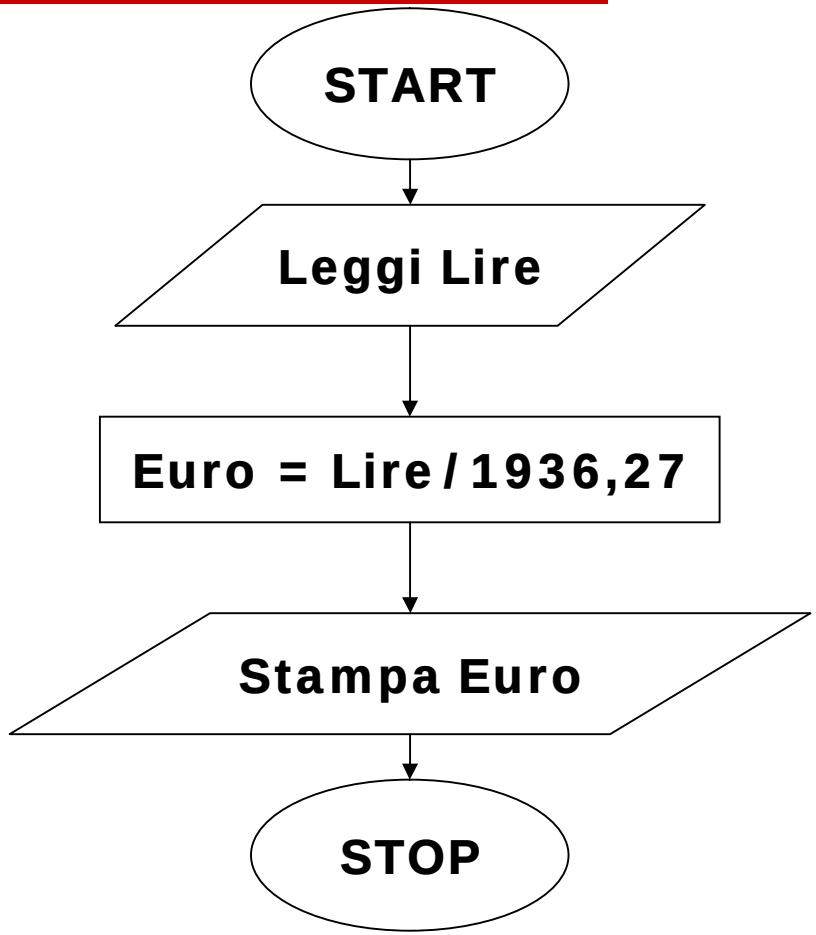
Calcola l’area di un rettangolo



Conversione lire in euro

- Leggi da input l'importo in lire
- Calcola il corrispettivo in Euro
- Dai in output il risultato

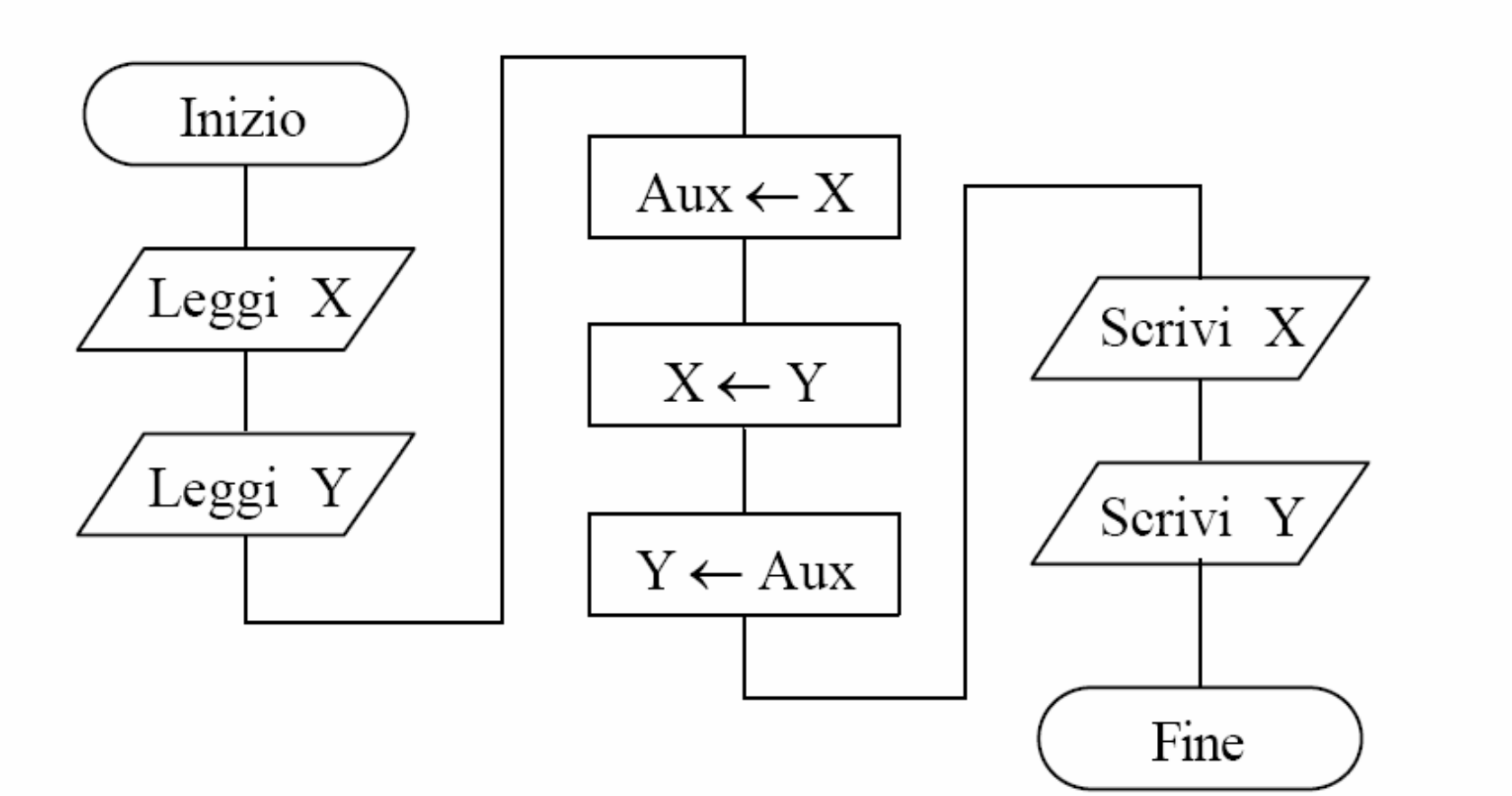
Conversione lire in Euro



Scambio dei valori di due variabili

- Leggi valore prima variabile X
- Leggi valore seconda variabile Y
- Conserva X in una variabile temporanea Aux
- Assegna il valore di Y ad X
- Assegna il valore di Aux a Y
- Scrivi X
- Scrivi Y

Scambio dei valori di due variabili

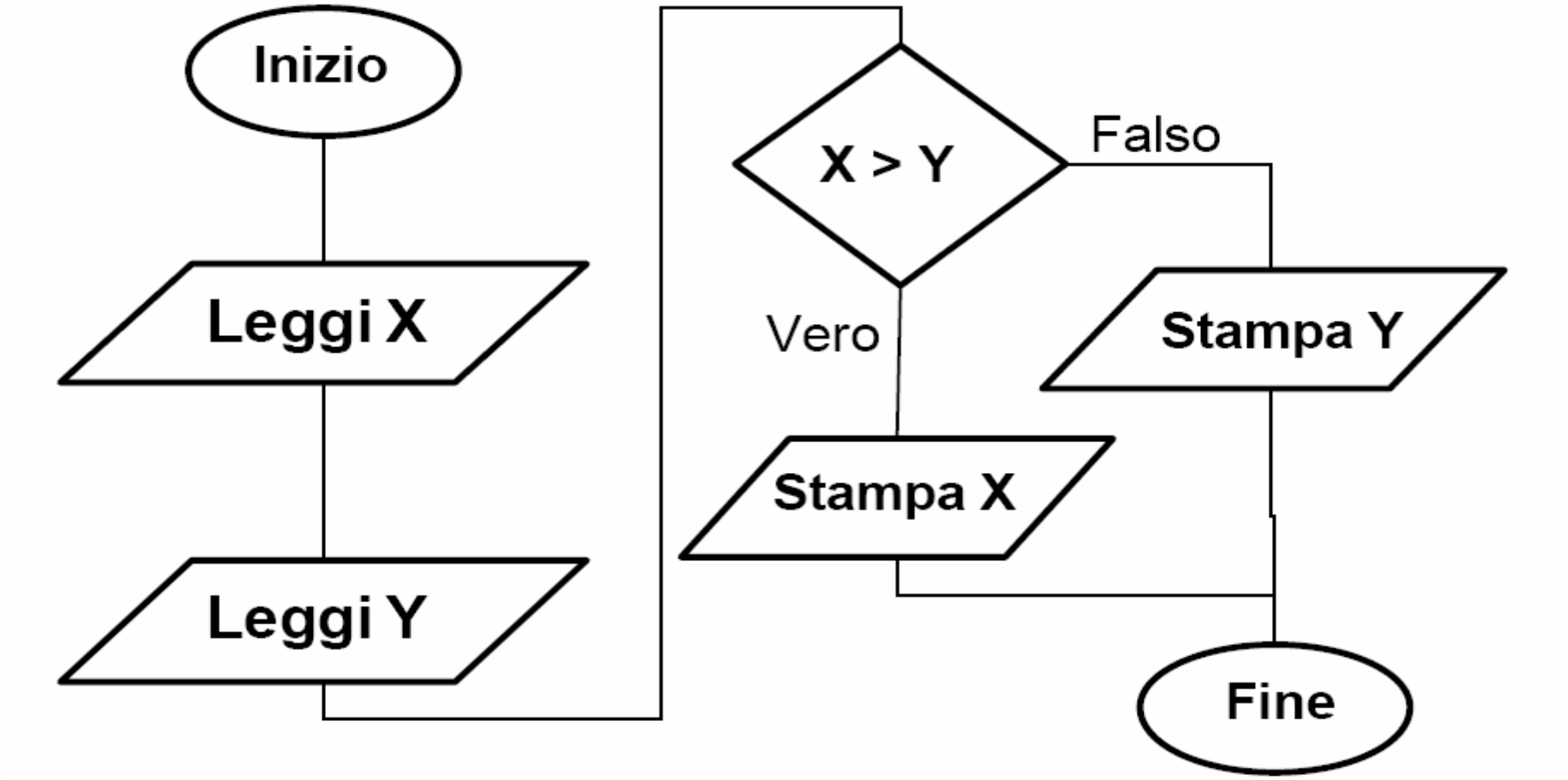


Massimo tra due numeri

- Leggi X
- Leggi Y
- Se X > Y
  - Stampa X

  Altrimenti
  - Stampa Y

Massimo tra due numeri



Pari o dispari

- Leggi N
- Dividi N per 2
- Se Resto = 0
  - Scrivi “N è pari”

  Altrimenti
  - Scrivi “N è dispari”

Pari o dispari