

Fondamenti di Informatica

Ing. Biomedica

Esercitazione n.1

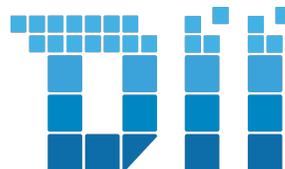
a.a. 2020/21

Antonio Arena

antonio.arena@ing.unipi.it



UNIVERSITÀ DI PISA



DIPARTIMENTO DI
INGEGNERIA
DELL'INFORMAZIONE



Definizione di Informatica

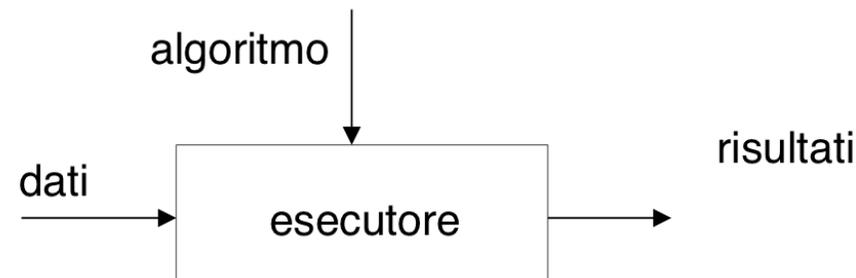
- **Informatica (definizione informale):** è la scienza della rappresentazione e dell'elaborazione dell'informazione.
- **Informatica (definizione formale dell'Association for Computing Machinery - ACM):** è lo studio sistematico degli algoritmi che descrivono e trasformano l'informazione, la loro teoria e analisi, il loro progetto, e la loro efficienza, realizzazione e applicazione.
- **Algoritmo:** sequenza precisa e finita di operazioni, comprensibili e perciò eseguibili da uno strumento informatico, che portano alla realizzazione di un compito. Esempi di algoritmi:
 - Istruzioni di montaggio di un elettrodomestico
 - Somma in colonna di due numeri
 - Bancomat



Algoritmi (1) - Definizione

- **Algoritmo:** sequenza *precisa* (non ambigua) e finita di operazioni, che portano alla realizzazione di un compito.
- Le operazioni utilizzate appartengono ad una delle seguenti categorie:
 - 1. Operazioni sequenziali**
Realizzano una singola azione. Quando l'azione è terminata passano all'operazione successiva.
 - 2. Operazioni condizionali**
Controllano una condizione. In base al valore della condizione, selezionano l'operazione successiva da eseguire.
 - 3. Operazioni iterative**
Ripetono l'esecuzione di un blocco di operazioni, finché non è verificata una determinata condizione.

- L'esecuzione delle azioni nell'ordine specificato dall'algoritmo consente di risolvere il problema.
- Risolvere il problema significa produrre risultati a partire da dati in ingresso.



L'algoritmo deve essere applicabile ad un qualsiasi insieme di dati in ingresso appartenenti al dominio di definizione dell'algoritmo (se l'algoritmo si applica ai numeri interi deve essere corretto sia per gli interi positivi che per gli interi negativi).



Algoritmi (3) - Esempi

1. **Calcolo soluzione dell'equazione di primo grado $ax + b = 0$**
 - leggi i valori di a e di b
 - calcola $-b$
 - dividi quello che hai ottenuto per a e chiama x il risultato
 - stampa x

2. **Calcolo del massimo fra due numeri a e b**
 - leggi i valori di a e di b
 - **se $a > b$ stampa a altrimenti stampa b**



Algoritmi (4) - Proprietà

- **Eseguibilità:** ogni azione deve essere eseguibile dall'esecutore in un tempo finito
- **Non-ambiguità:** ogni azione deve essere univocamente interpretabile dall'esecutore
- **Finitezza:** il numero totale di azioni da eseguire, per ogni insieme di dati in ingresso, deve essere finito

- **Algoritmi equivalenti**
 - hanno lo stesso dominio di ingresso
 - hanno lo stesso dominio di uscita
 - in corrispondenza degli stessi valori del dominio di ingresso producono gli stessi valori del dominio di uscita

Due algoritmi equivalenti forniscono lo stesso risultato, ma possono avere diversa efficienza e possono essere profondamente diversi.



Algoritmi (5) – Esempio algoritmi equivalenti

L'algoritmo di moltiplicazione tra due numeri

Algoritmo 1

Somme successive:

$$12 \times 12 = 12 + 12 + \dots + 12 = 144$$

Algoritmo 2

“somma e shift”:

$$\begin{array}{r} 12 \times \\ 12 = \\ \hline 24 + \\ 12 = \\ \hline 144 \end{array}$$



Algoritmi (6) – Proprietà

- **Proprietà essenziali degli algoritmi:**
- **Correttezza:**
un algoritmo è corretto se esso perviene alla soluzione del compito cui è preposto, senza difettare in alcun passo fondamentale.
- **Efficienza:**
un algoritmo è efficiente se perviene alla soluzione del compito cui è preposto nel modo più veloce possibile, compatibilmente con la sua correttezza.



Algoritmi (7) – Ricapitolando

- La formulazione testuale di un algoritmo in un linguaggio comprensibile ad un calcolatore è detta **PROGRAMMA**.

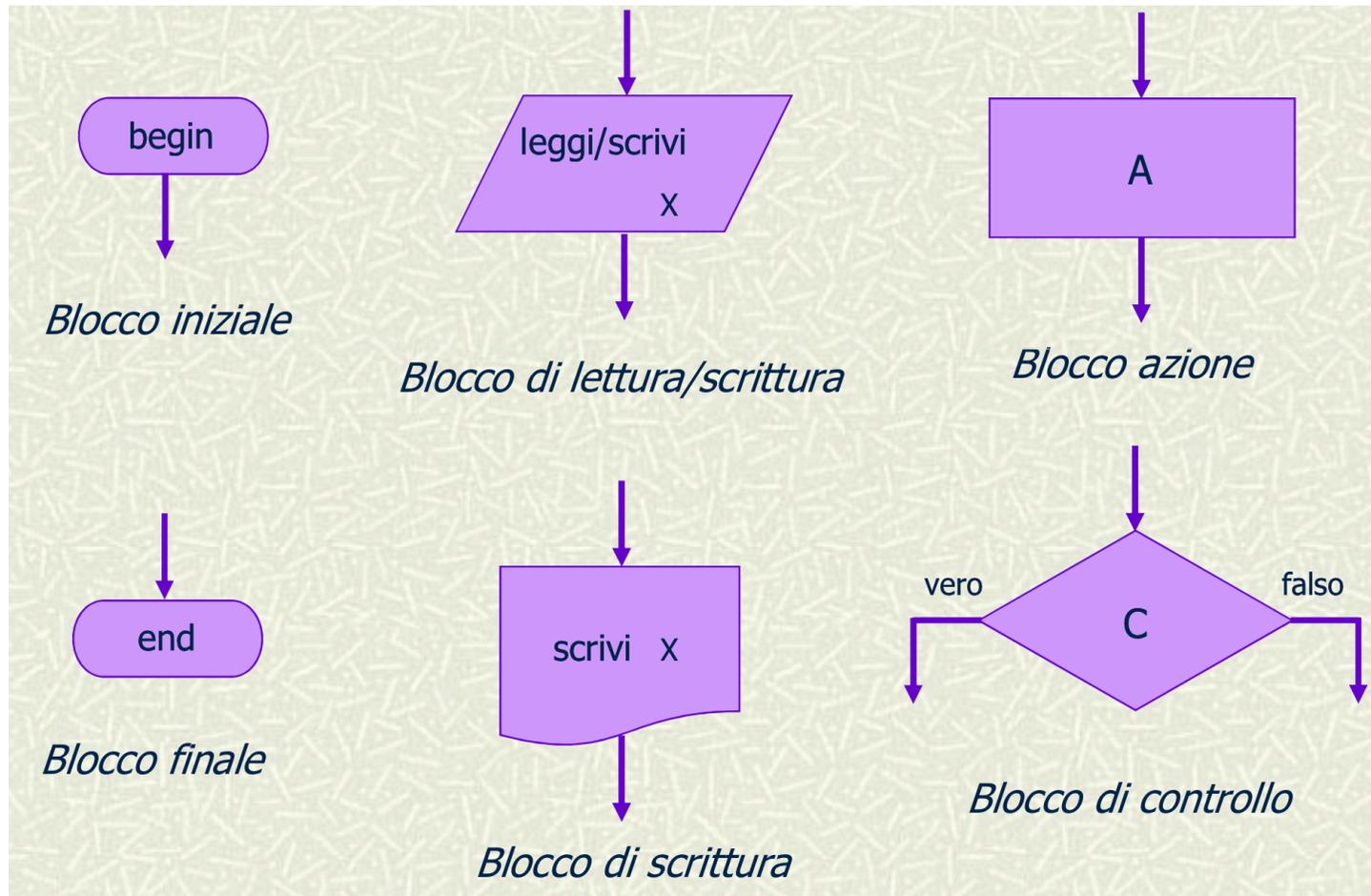
- Ricapitolando, per risolvere un problema:
 1. Individuazione di un procedimento risolutivo
 2. Scomposizione del procedimento in un insieme ordinato di azioni → *Algoritmo*
 3. Rappresentazione dei dati e dell'algoritmo attraverso un formalismo comprensibile al calcolatore: *Linguaggio di programmazione* → *Programma*



Diagrammi di flusso (1)

- Il linguaggio dei diagrammi a blocchi è un possibile formalismo per la descrizione di algoritmi
- Il diagramma a blocchi, o flowchart (diagramma di flusso), è una rappresentazione grafica dell'algoritmo
- Un diagramma a blocchi descrive il flusso delle operazioni da eseguire per realizzare la trasformazione, definita nell'algoritmo, dai dati iniziali ai risultati
- Ogni istruzione dell'algoritmo viene rappresentata all'interno di un blocco elementare, la cui forma grafica è determinata dal tipo di istruzione
- I blocchi sono collegati tra loro da linee di flusso, munite di frecce, che indicano il susseguirsi di azioni elementari

Diagrammi di flusso (2)





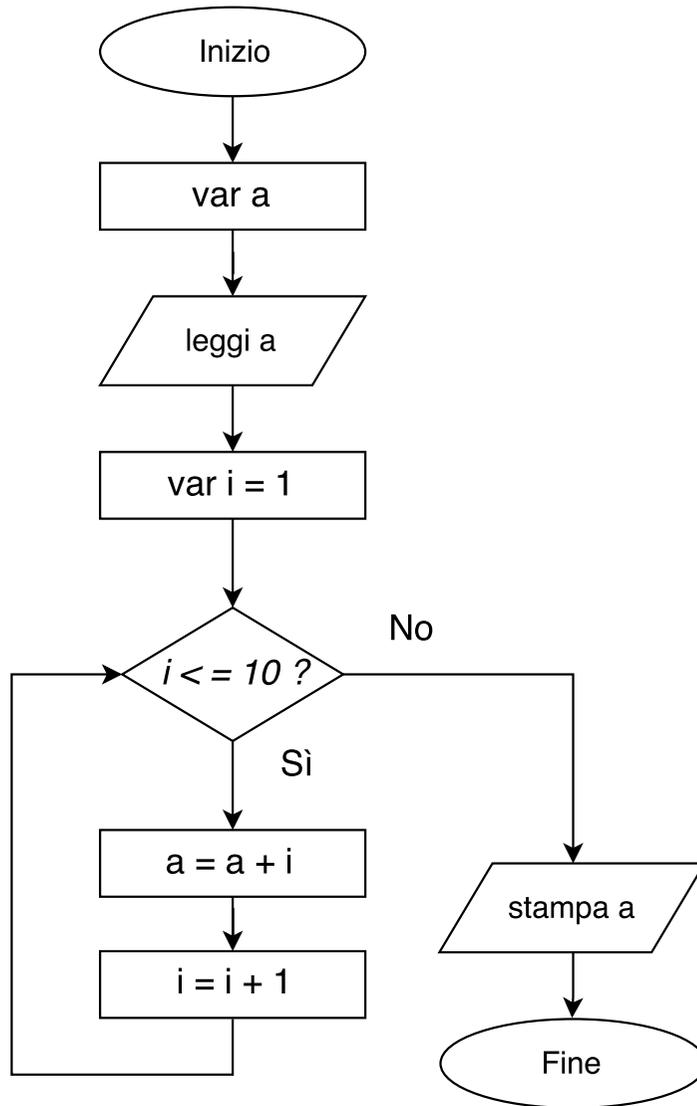
Diagrammi di flusso (3)

- **Un diagramma a blocchi è un insieme di blocchi elementari composto da:**
 - 1. un blocco iniziale (e solo uno!)**
 - 2. un blocco finale (e solo uno!)**
 - 3. un numero finito n ($n \geq 1$) di blocchi di azione e/o di blocchi di lettura/scrittura**
 - 4. un numero finito m ($m \geq 0$) di blocchi di controllo**



Diagrammi di flusso (4)

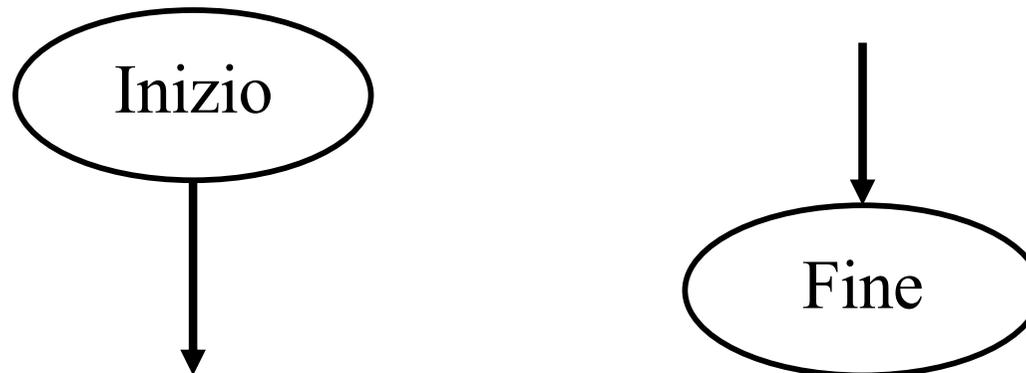
- **L'insieme dei blocchi elementari che descrivono un algoritmo deve soddisfare le seguenti condizioni:**
 - **ciascun blocco di azione o di lettura/scrittura ha una sola freccia entrante ed una sola freccia uscente**
 - **ciascun blocco di controllo ha una sola freccia entrante e due frecce uscenti**
 - **ciascuna freccia entra in un blocco oppure si innesta in un'altra freccia**
 - **ciascun blocco è raggiungibile dal blocco iniziale**
il blocco finale è raggiungibile da qualsiasi altro blocco



- L'insieme dei dati di ingresso e dei risultati vengono rappresentati attraverso dei nomi simbolici, detti **variabili** (ad esempio, «a»), che possono essere visti come dei contenitori in cui mantenere dei risultati.
- Può essere inoltre necessario introdurre delle variabili "temporanee" (ad esempio, i), necessarie alla risoluzione del problema: tali variabili vengono anch'esse rappresentate da nomi simbolici.

Blocchi (1) - Inizio e fine esecuzione

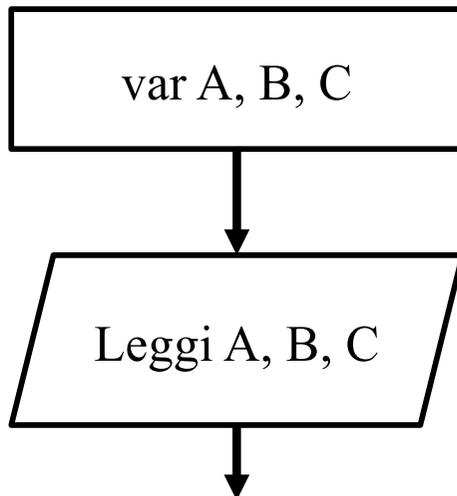
- **Inizio** è il blocco da cui deve iniziare l'esecuzione dell'algoritmo (uno solo). La freccia è sempre uscente dal blocco inizio.
- **Fine** è il blocco che fa terminare l'esecuzione dell'algoritmo (uno solo). La freccia è sempre entrante nel blocco fine.



- A questi blocchi non corrisponde alcuna azione!

- **Esecuzione dell'istruzione:**

Si ricevono dall'unità di ingresso (per esempio, la tastiera) tanti valori quante sono le variabili specificate all'interno del blocco

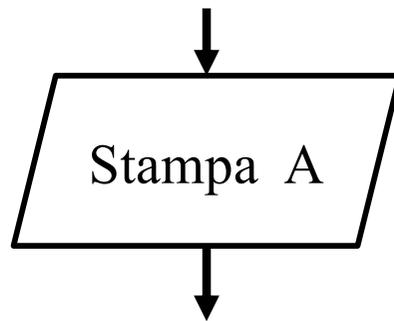


Significato blocchi:

«Crea le variabili (contenitori) di nome A,B,C. Leggi i tre valori da tastiera, e salvali rispettivamente nelle variabili (contenitori) di nome A, B e C»

- **Esecuzione dell'istruzione:**

Si calcolano i valori delle espressioni e si trasmettono all'unità di uscita (ad esempio, il video)



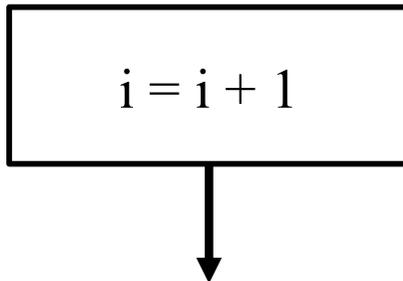
Significato blocco:

«calcola il valore dell' espressione A (che potrebbe essere anche il valore salvato in una variabile e stampala a video)»

N.B. nel caso in cui A fosse una variabile, il suo contenuto non viene alterato dall'esecuzione del blocco di uscita

- **Esecuzione dell'istruzione:**

Si calcola il valore dell'espressione a destra del simbolo «=» e lo si salva (*assegna*) nella variabile indicata a sinistra del simbolo «=»

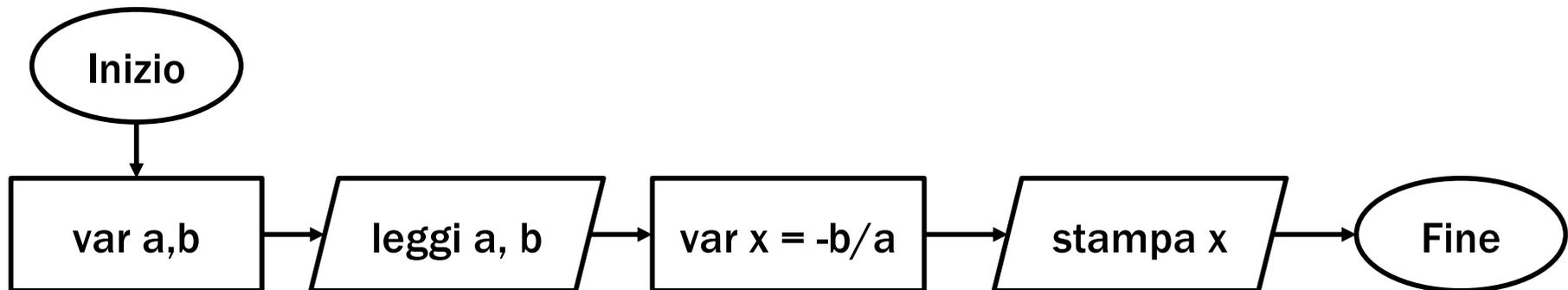


Significato blocco:

«Calcola il valore dell'espressione $i+1$ ed assegnalo alla (salvalo nella) variabile i »

- Esempio: se i vale 6 prima di entrare nel blocco, calcolo $6+1$ (7) e lo salvo nella variabile i .
- Quando si esce dal blocco la variabile i vale 7.

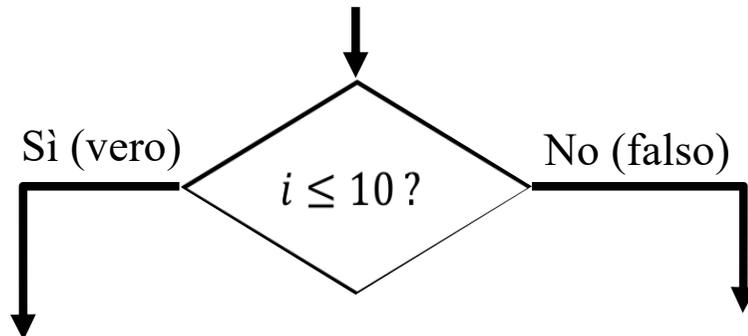
- Con questi blocchi visti fino ad ora è possibile già rappresentare un insieme di algoritmi, ad esempio il calcolo della soluzione dell'equazione di primo grado $ax + b = 0$



- Questi tipi di diagrammi di flusso sono detti *sequenziali*
- E' corretto questo algoritmo? Funzione per ogni coppia di ingresso (a,b)?

Esecuzione dell'istruzione:

Si valuta la condizione specificata all'interno del blocco: se è verificata, si prosegue con la line di flusso contrassegnata da «Sì», altrimenti (se non è verificata) si prosegue con il ramo etichettato con «No»



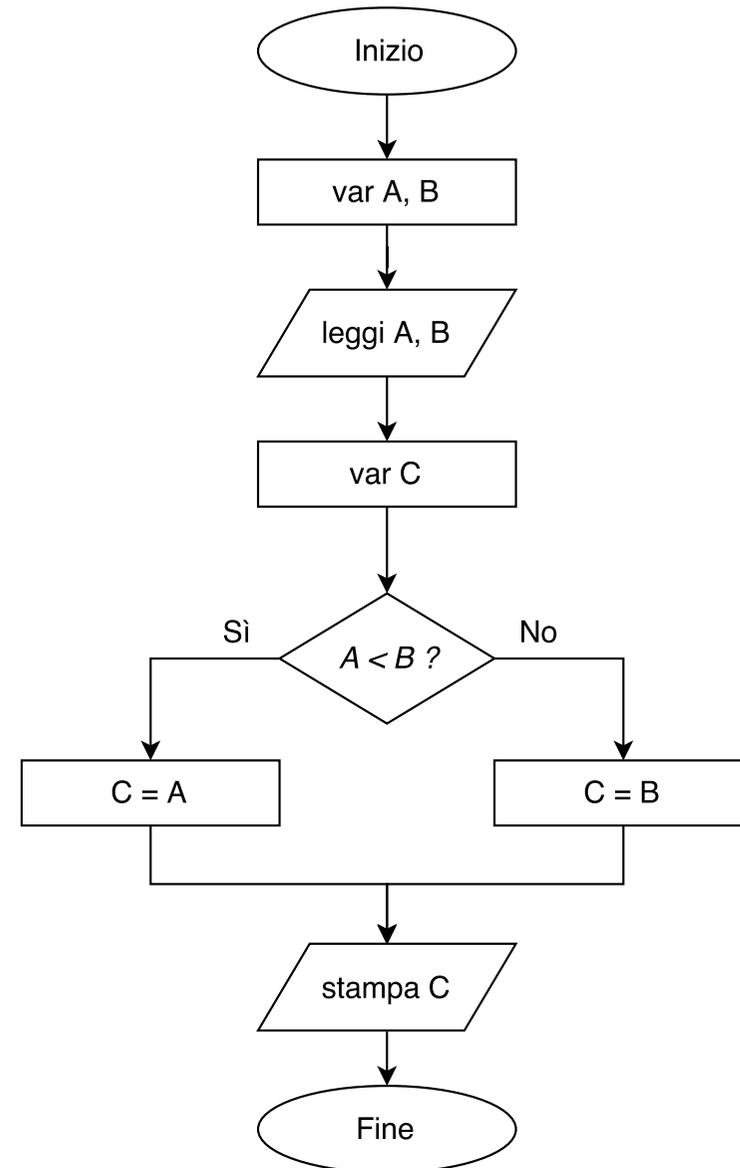
Significato blocco:

«Calcola il valore dell'espressione $i \leq 10$: se è vero, prosegui per il ramo Sì, altrimenti prosegui per il ramo No»

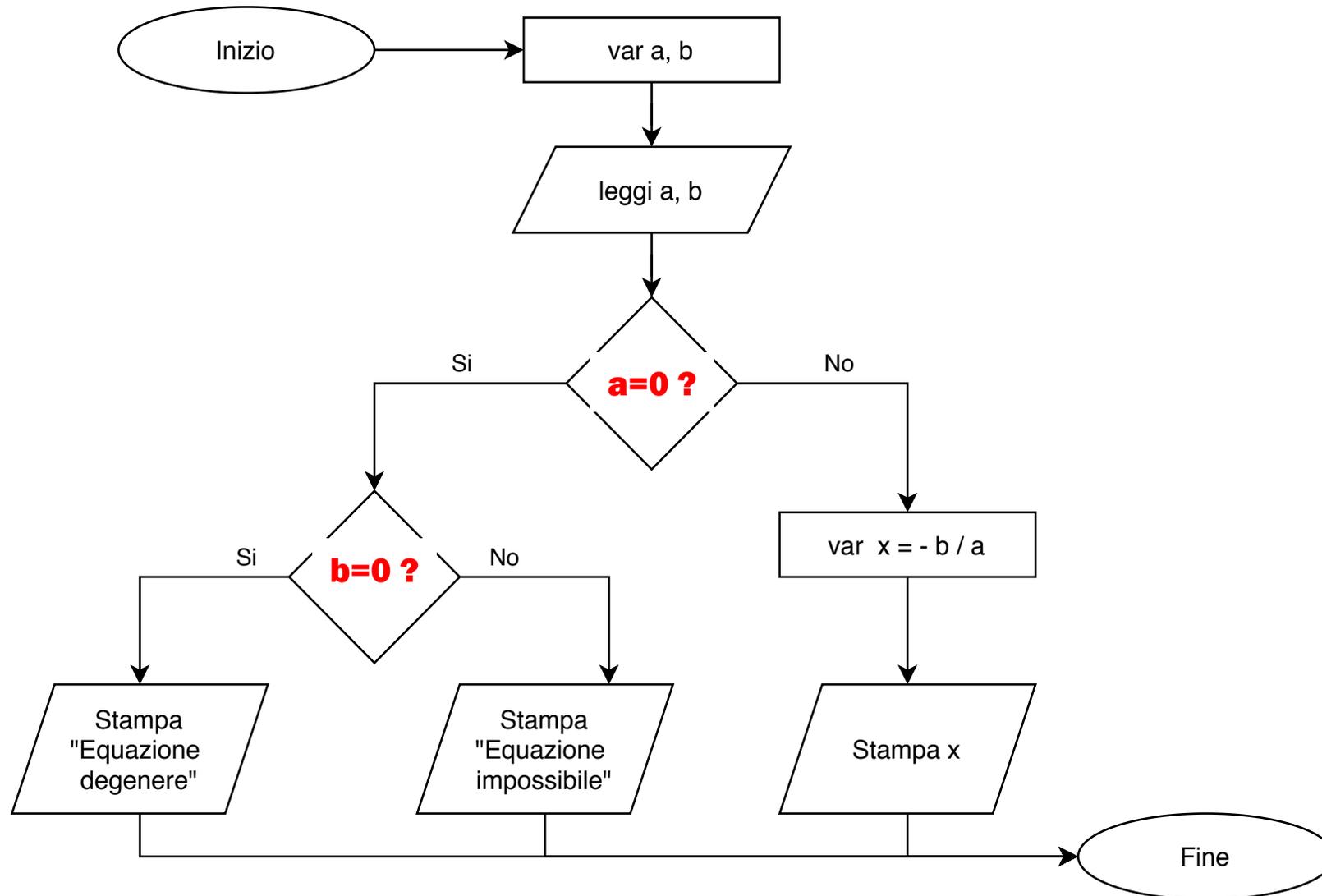
- Il blocco condizione è l'elemento base per realizzare **alternative** e **ripetizioni**, e in generale diagrammi di flusso *strutturati*.

Questo diagramma di flusso è un semplice diagramma *strutturato*, ovvero ha almeno due percorsi differenti che partono da inizio e arrivano a fine.

Quale problema risolve questo algoritmo?



- **Calcolo soluzione equazione di primo grado $ax + b = 0$**





Blocchi (6) – Iterazione o Ripetizione

Esprime la ripetizione di una sequenza di istruzioni.

Nel caso più generale (ripetizione enumerativa), è costituita da:

- **Inizializzazione:** assegnazione dei valori iniziali alle variabili caratteristiche del ciclo (viene eseguita una sola volta);
- **Controllo:** determina, in base al valore delle variabili che controllano l'esecuzione del ciclo se il ciclo deve essere ripetuto o meno;
- **Corpo:** esecuzione delle istruzioni fondamentali del ciclo che devono essere eseguite in modo ripetitivo;
- **Modifica:** modifica dei valori che controllano l'esecuzione del ciclo (eseguito ad ogni iterazione).

▪ Inizializzazione:

la variabile i sarà la variabile che controllerà l'esecuzione del corpo del ciclo, e il numero di volte che verrà eseguito

▪ Controllo:

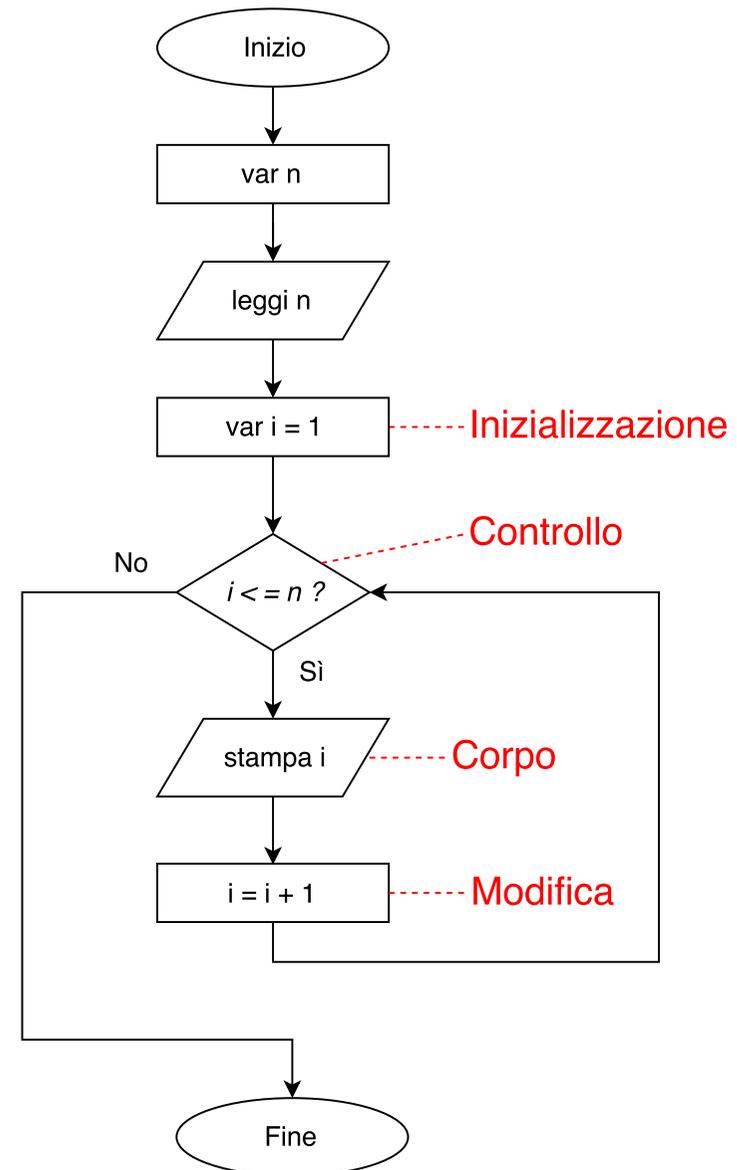
il corpo del ciclo verrà eseguito fintanto che la variabile i è minore o uguale alla variabile n (fornita tramite tastiera dall'utente)

▪ Corpo:

viene stampata i

▪ Modifica:

la variabile i viene incrementata di uno. Prima o poi arriverà al valore della variabile n , e il ciclo terminerà





Blocchi (8) – in conclusione...

In generale, qualsiasi algoritmo si compone attraverso le seguenti strutture:

- **Concatenazione:** istruzioni di lettura, scrittura, assegnamento, calcolo di espressioni eseguite in sequenza
- **Alternative:** blocchi condizionali
- **Iterazioni**

Chiaramente, la complessità di un algoritmo potrà essere alta o bassa, ma sarà **sempre e in ogni caso** esprimibile attraverso queste blocchi base appena vista (composte in modo vario, come componiamo pezzi di lego), che formano la cosiddetta base della programmazione “strutturata”.



- **Mediante diagrammi di flusso, descrivere le operazioni da eseguire per poter completare la richiesta:**
 - 1. Leggere un numero da tastiera e stampare a video se è positivo o negativo**
 - 2. Leggere da tastiera due numeri e stampare a video la loro somma**
 - 3. Leggere da tastiera due numeri e stampare a video il maggiore dei due**
 - 4. Leggere tre numeri da tastiera e stampare a video il massimo tra i tre numeri**
 - 5. Leggere da tastiera il valore dei lati di un triangolo e stampare a video perimetro e area del triangolo**
 - 6. Calcolare il risultato dell'equazione di secondo grado leggendo da tastiera i coefficienti dell'equazione $ax^2 + bx + c = 0$ e stampando il valore di x_1 e x_2**