# **Idoneità di Informatica**
# Lezione 3

Mario Verdicchio
Università degli Studi di Bergamo
A.A. 2026-2027

# Hardware



# Software



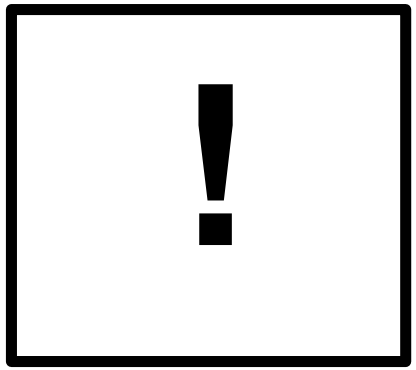







**Problema**

**Un ostacolo, un impedimento, una mancanza che vogliamo eliminare.**



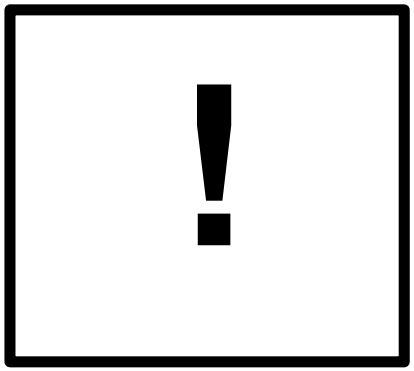
**Soluzione**

**Eliminazione del problema.**



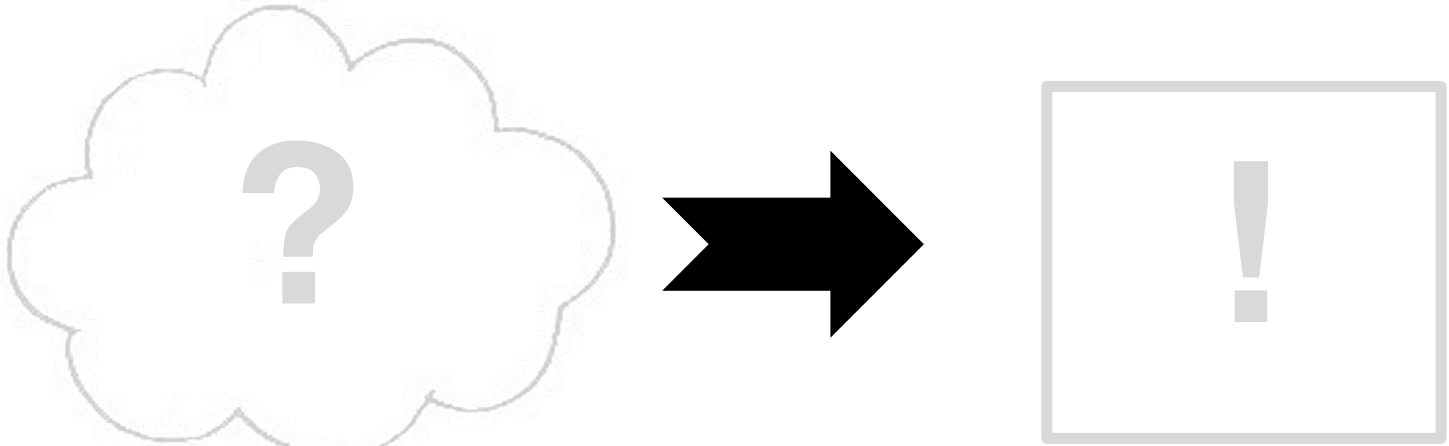
**Computer**






**Un problema si dice risolvibile quando esiste una soluzione per esso.**
**Se vogliamo che sia il computer a trovare la soluzione, il problema deve essere espresso sotto forma di dati che il computer possa elaborare.**

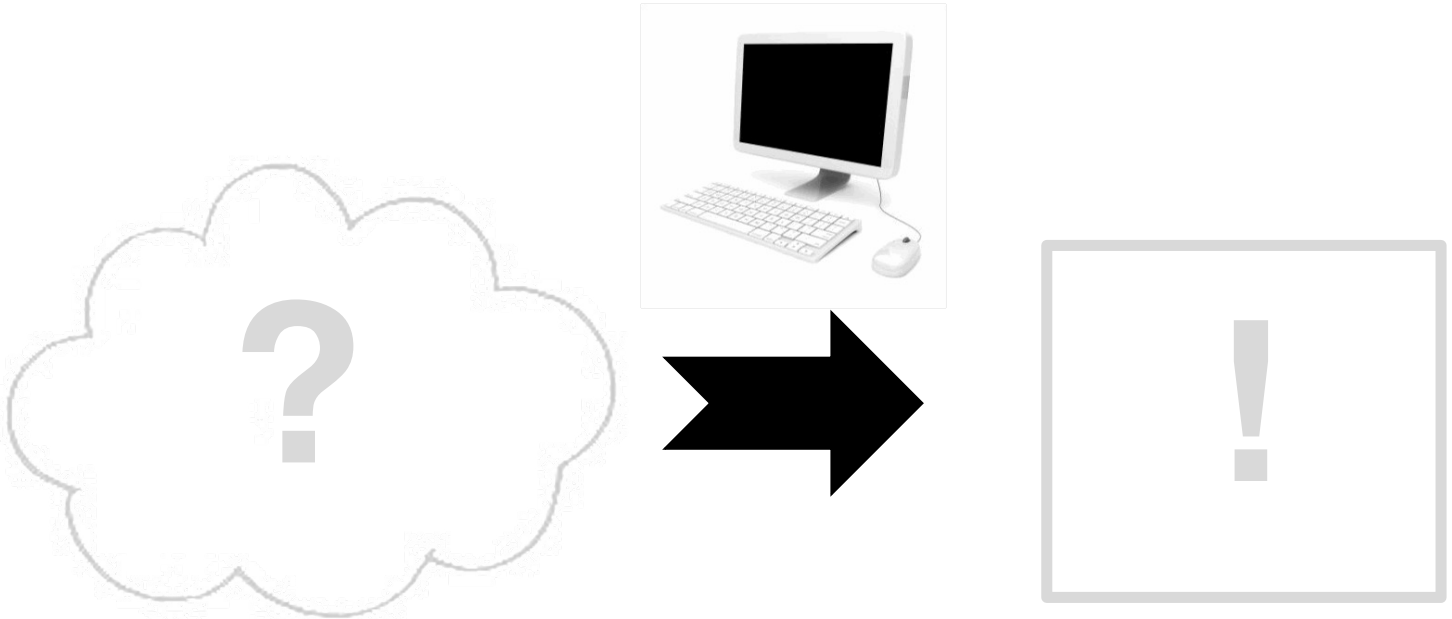



**Algoritmo**

**Un metodo per giungere alla soluzione che presenta le seguenti caratteristiche:**
**1) è una sequenza finita di operazioni ben definite;**
**2) dopo l'esecuzione di ciascuna operazione, è chiaro quale sia la successiva.**



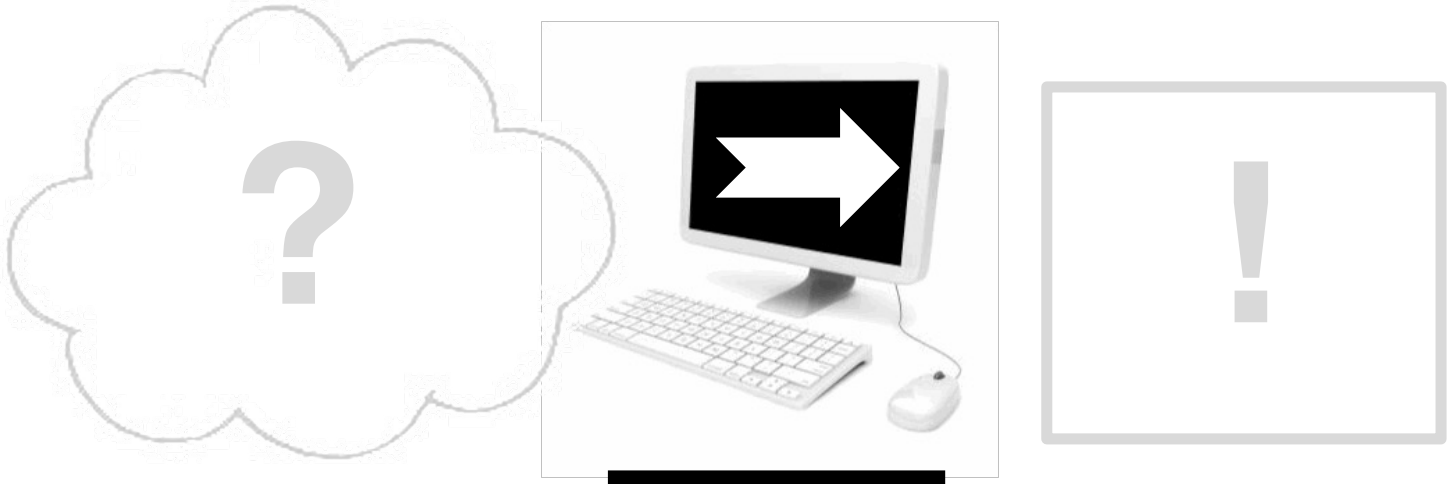

# Muḥammad ibn Mūsā al-Khwārizmī
# (780 – 850)




**Programma**

**Un algoritmo scritto in modo tale da poter essere eseguito da un computer.**

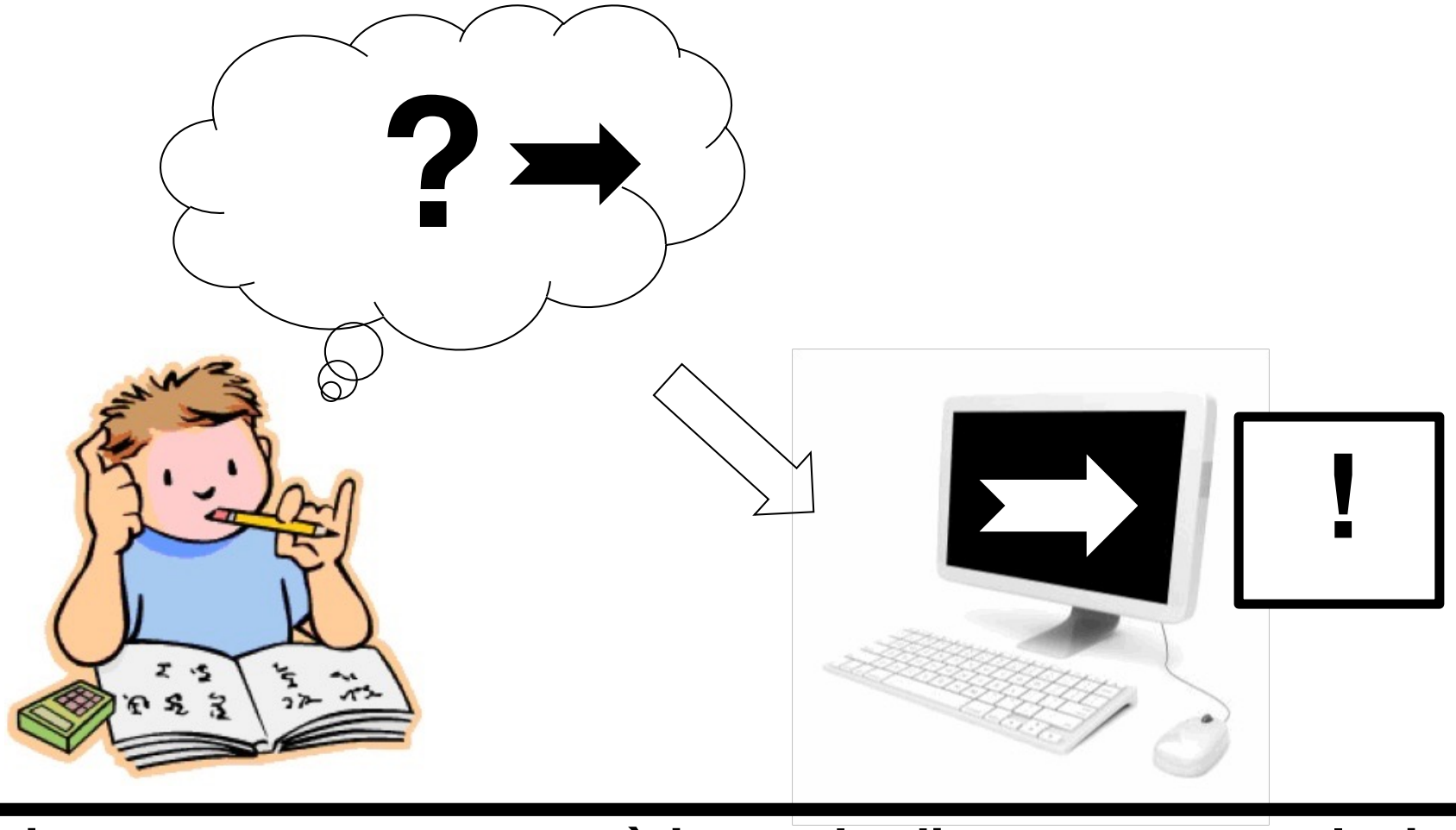
**Tutte le operazioni consistono nell'elaborazione di dati numerici.**




**Processo**

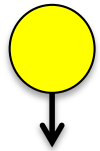
**Un programma in esecuzione su un computer.**

**In linea di massima, i programmi risiedono solitamente sul disco rigido del computer, mentre i processi risiedono nella RAM.**

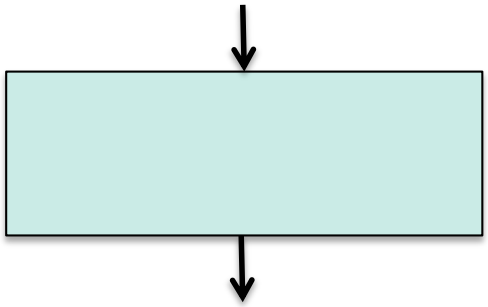



**Attenzione: un computer non è in grado di trovare una soluzione autonomamente. L'algoritmo deve prima essere ideato e successivamente trasferito nel computer sotto forma di programma, affinché venga eseguito e fornisca una soluzione.**

# Diagrammi di flusso

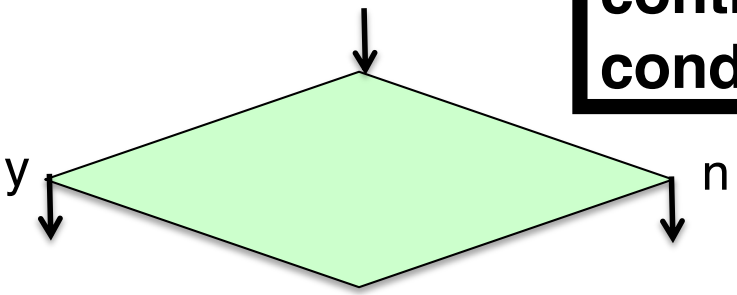


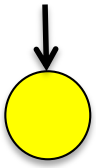
inizio



operazione

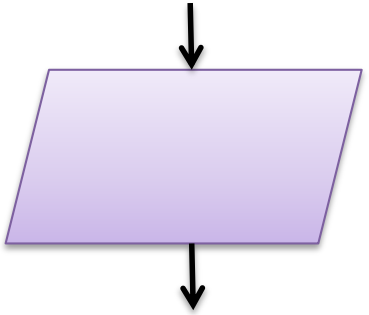
controllo di condizione

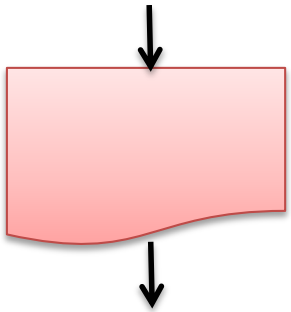


fine

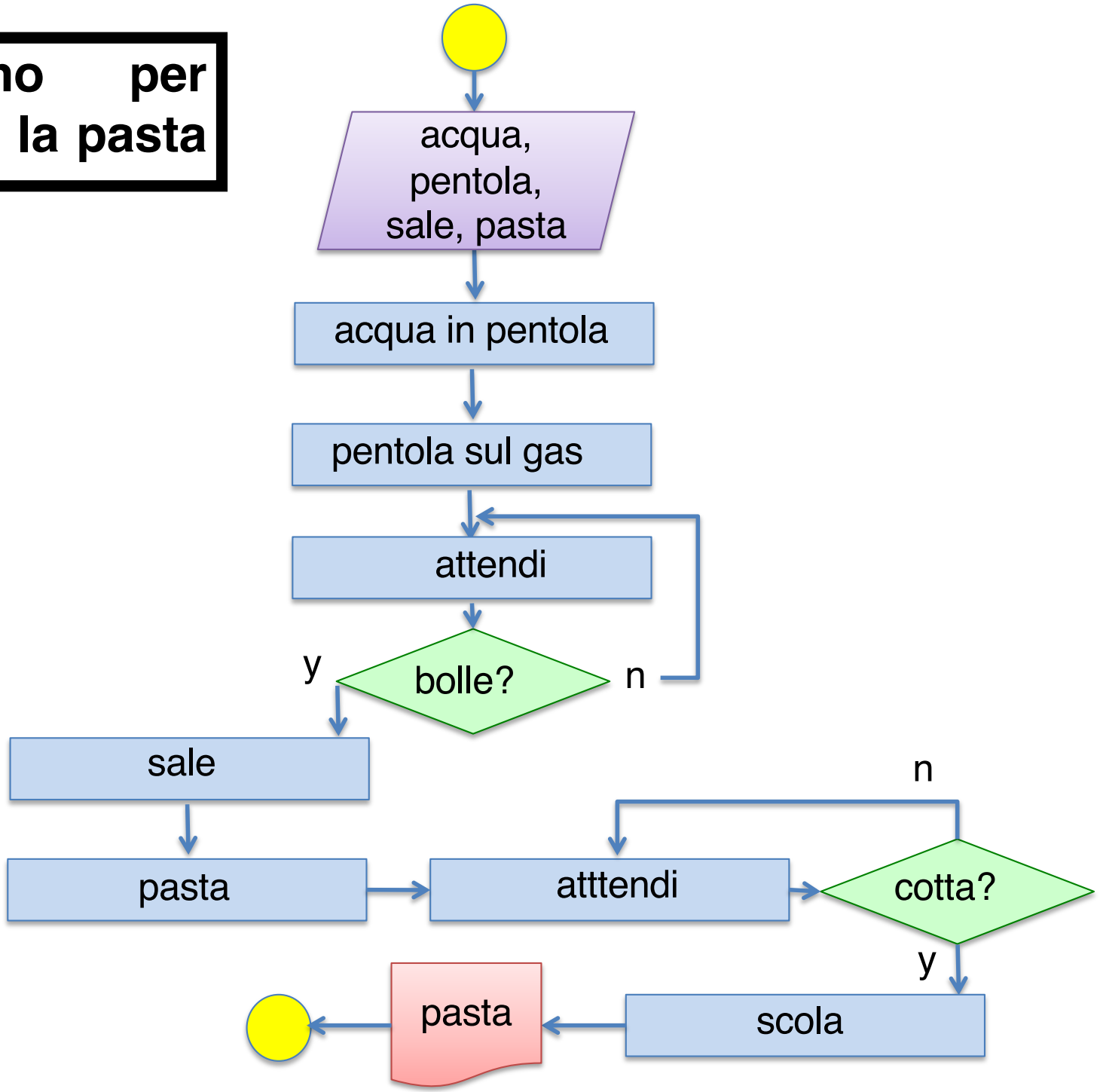
input





output

**Algoritmo per cuocere la pasta**



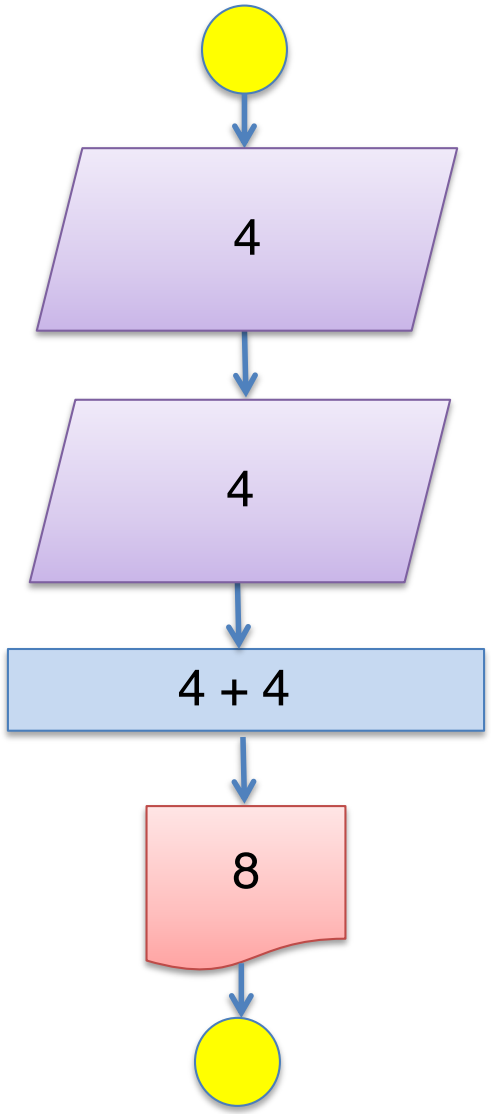
**Algoritmo per esseri umani, non per computer, per vari motivi.**



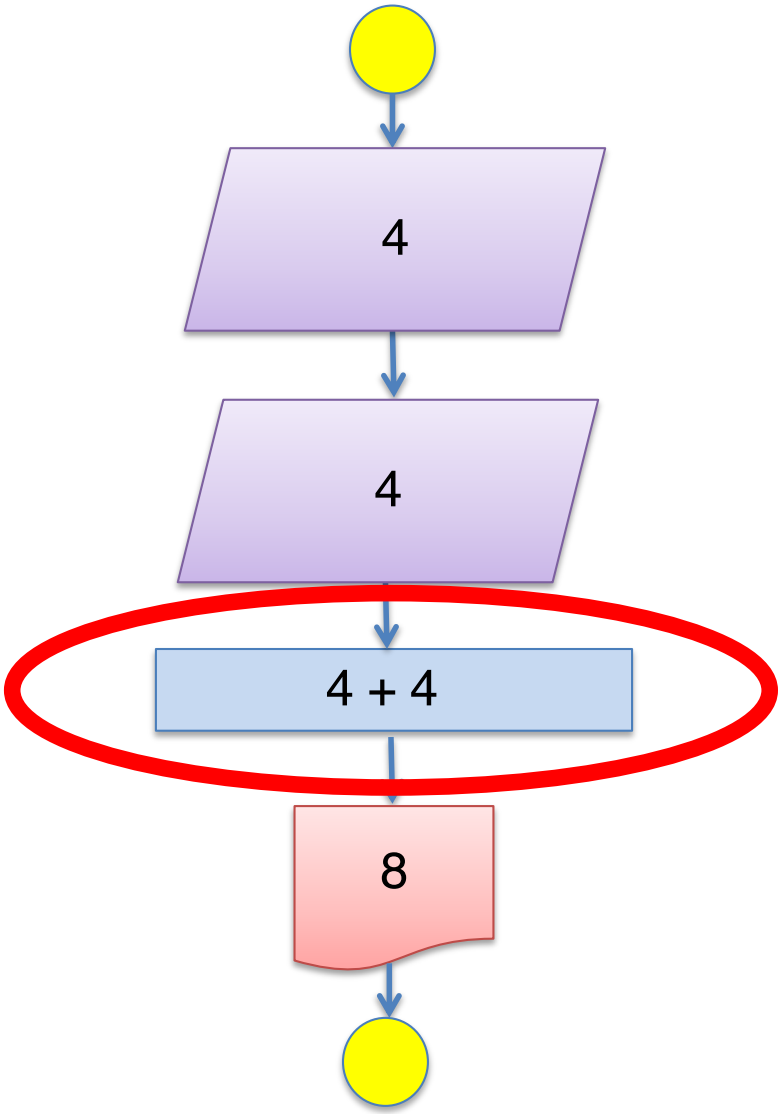



**Un computer è un calcolatore: può solo ricevere dati numerici, elaborare dati numerici, e restituire dati numerici.**

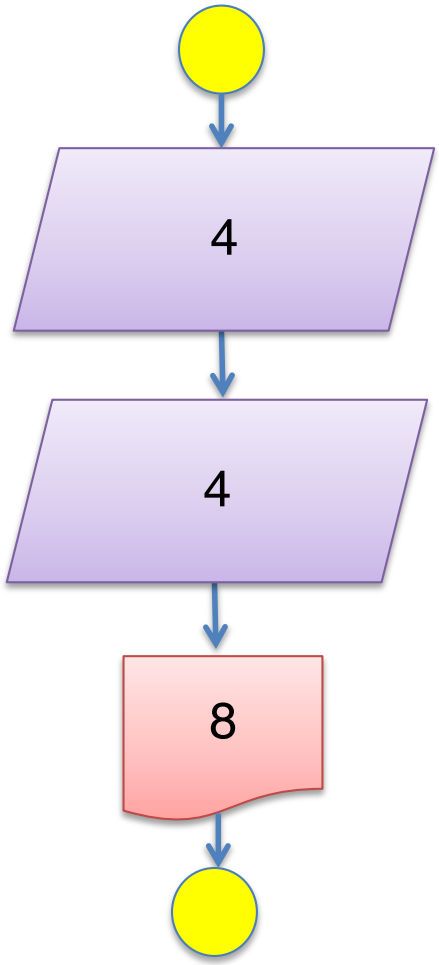
**Algoritmo per fare 4 + 4.**


4
4
4 + 4
8



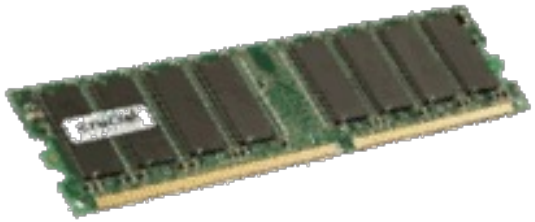

4
4
4 + 4
8




4
4
8



# Variabili

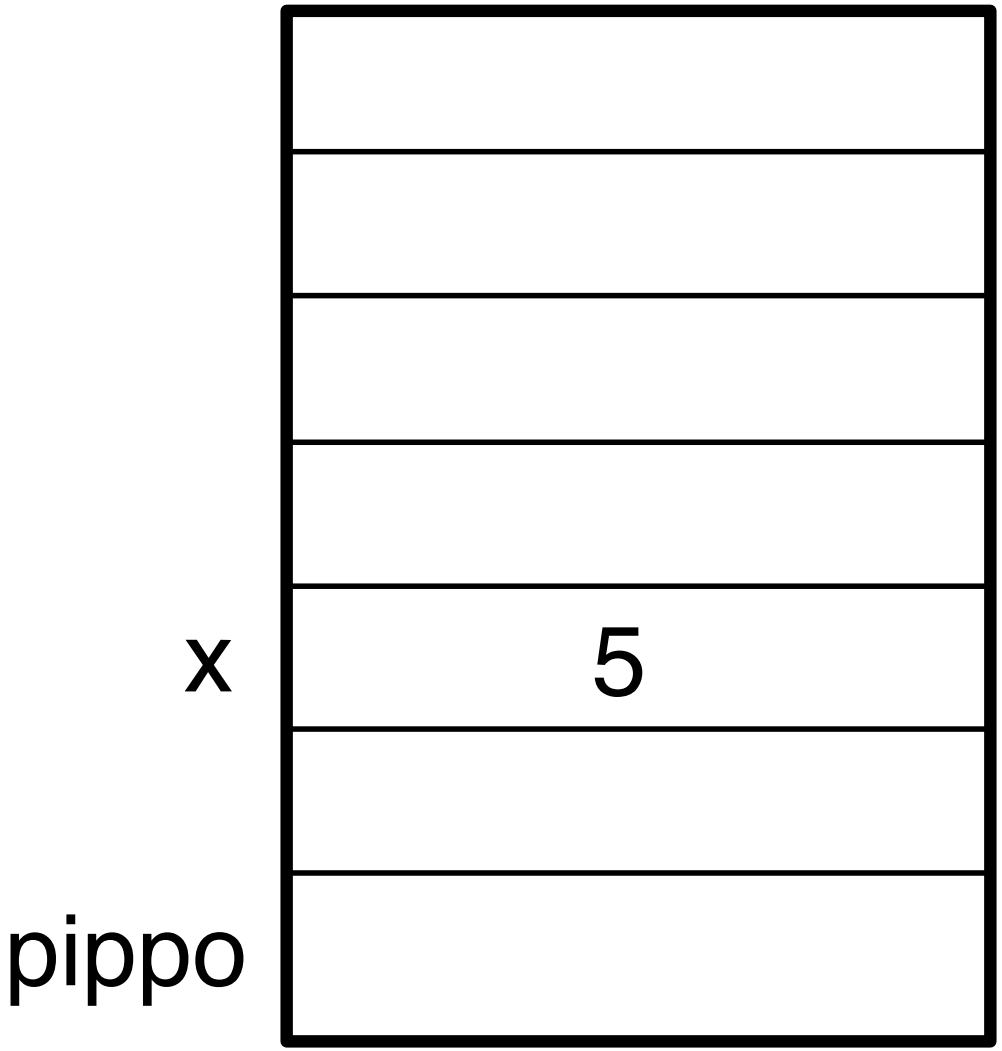


x

pippo

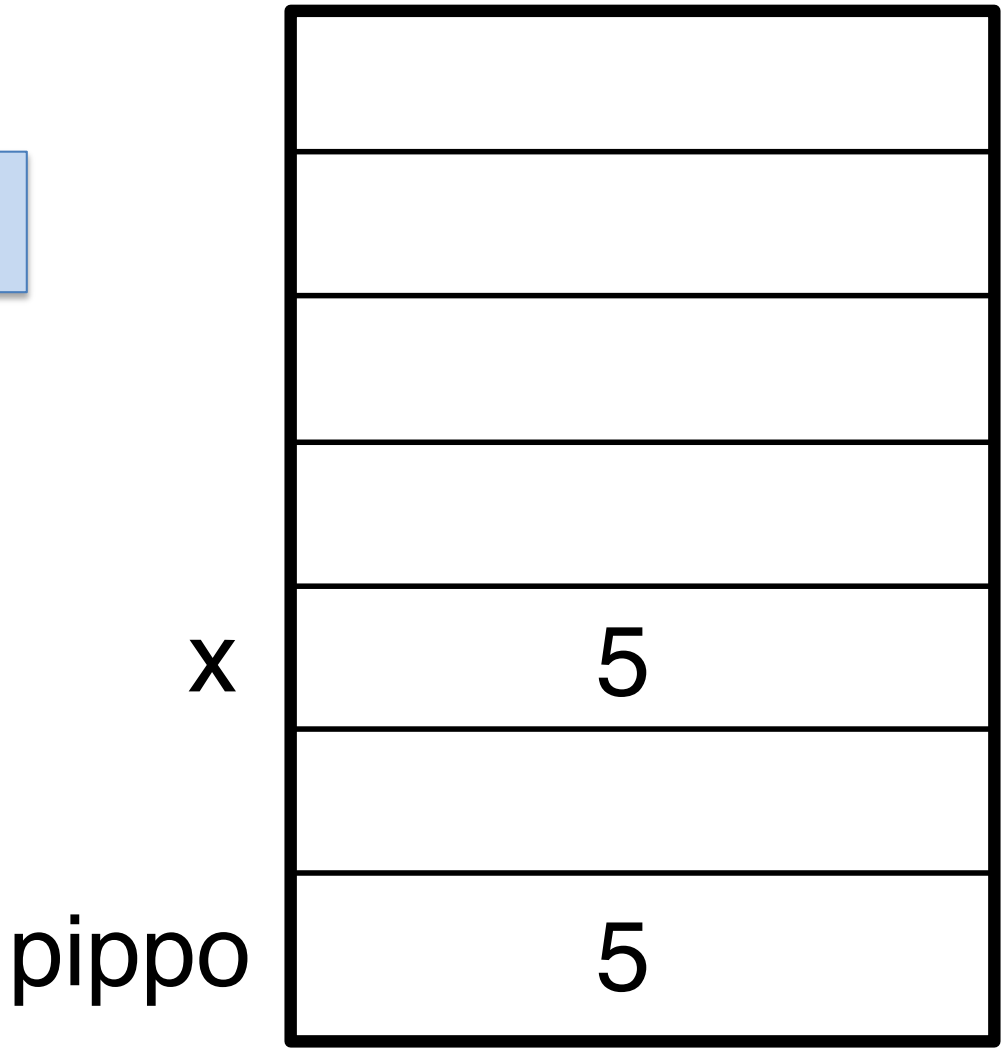
**Assegnamenti**

$x \leftarrow 5$



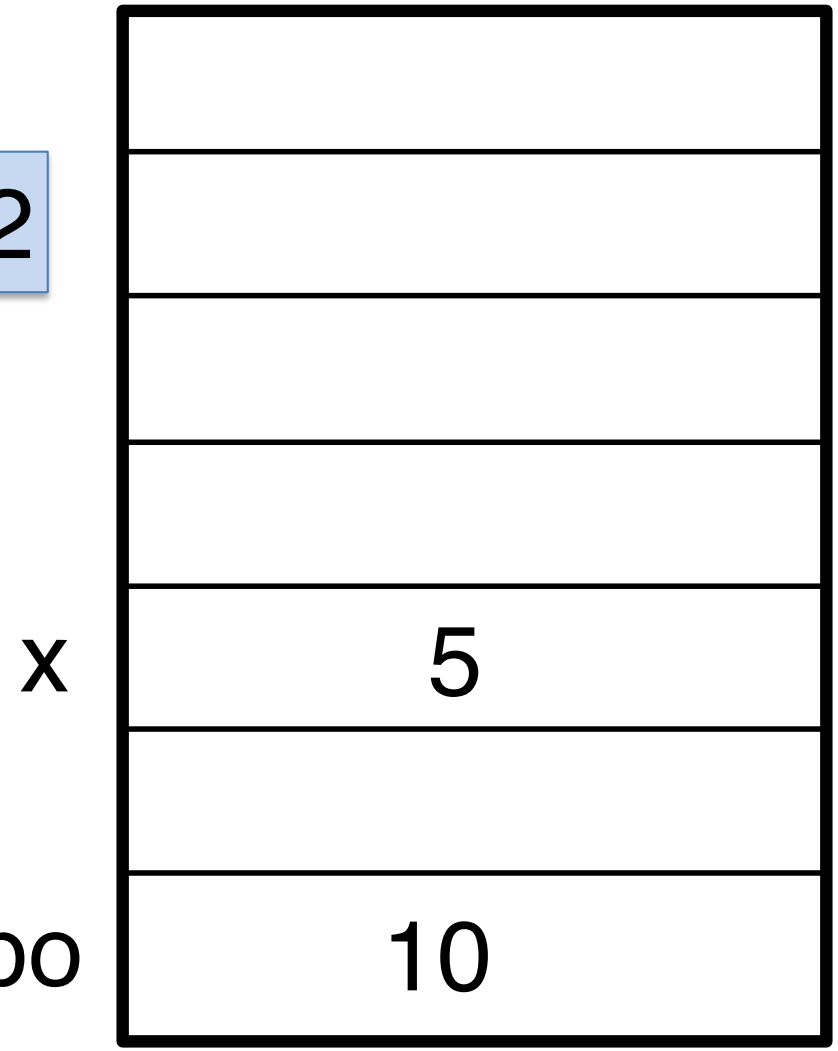
**Assegnamenti**

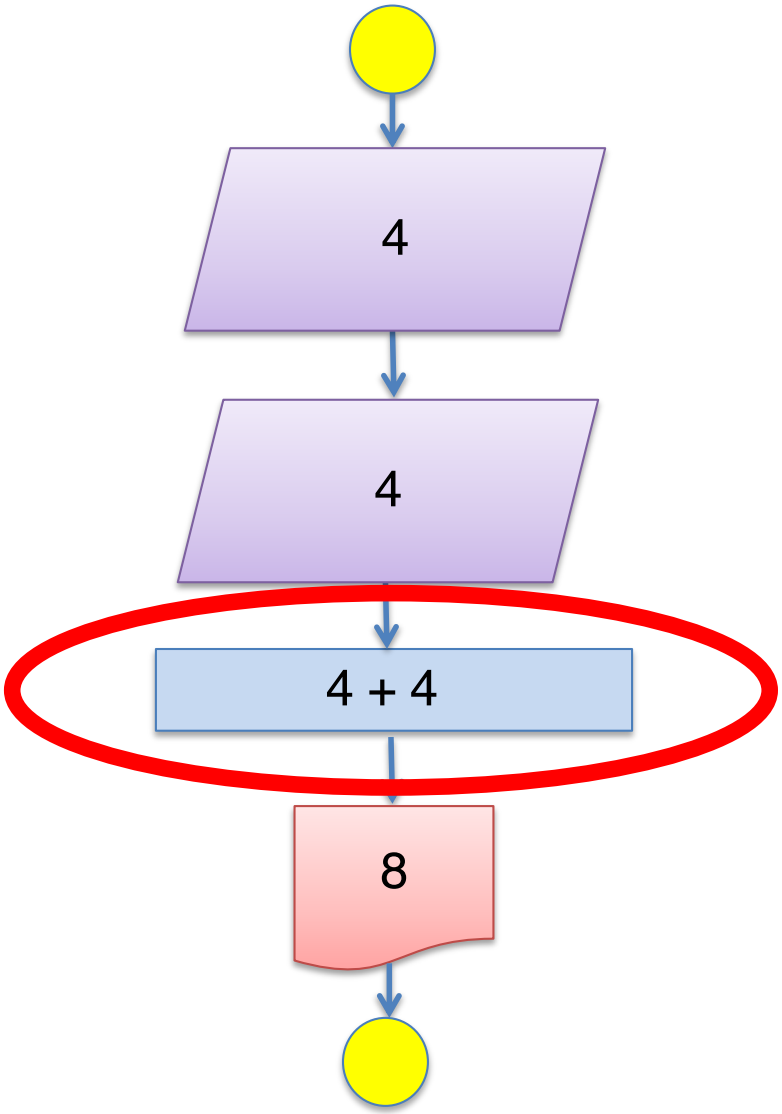
pippo ← x



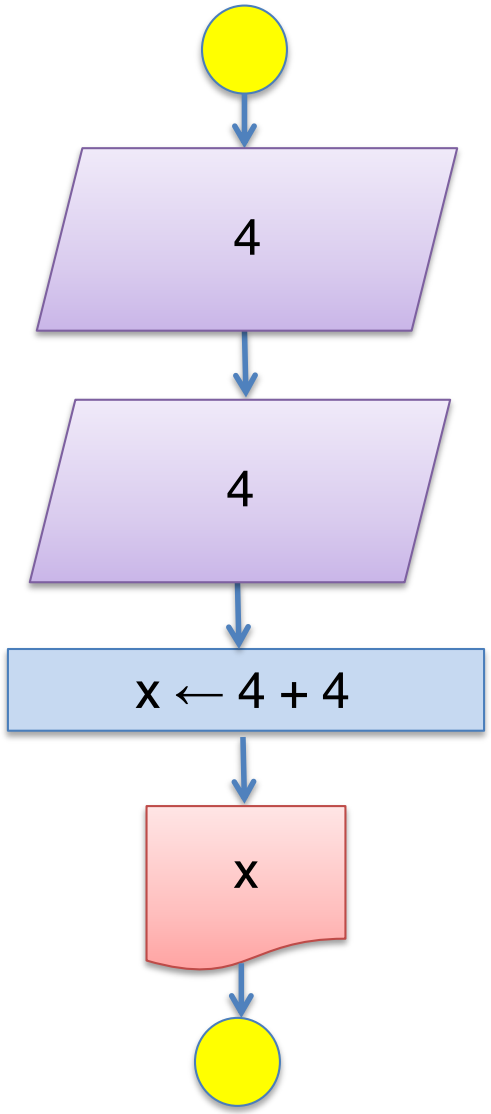
**Assegnamenti**

pippo ← pippo × 2

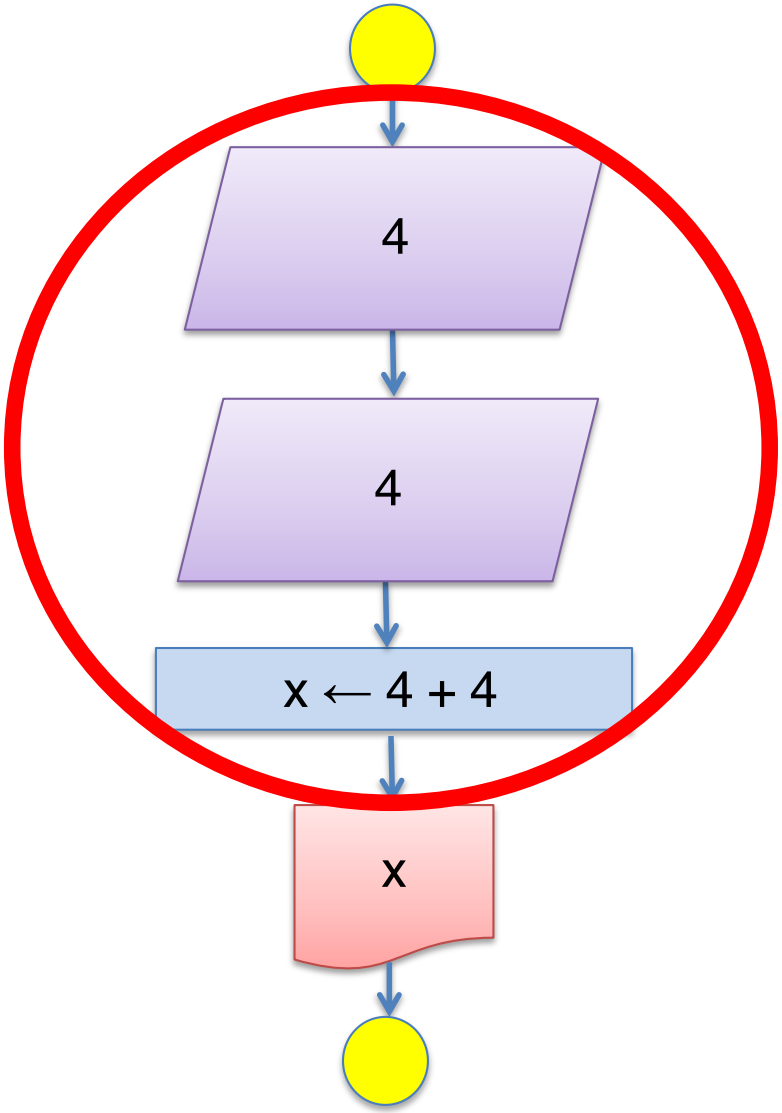



4
4
4 + 4
8

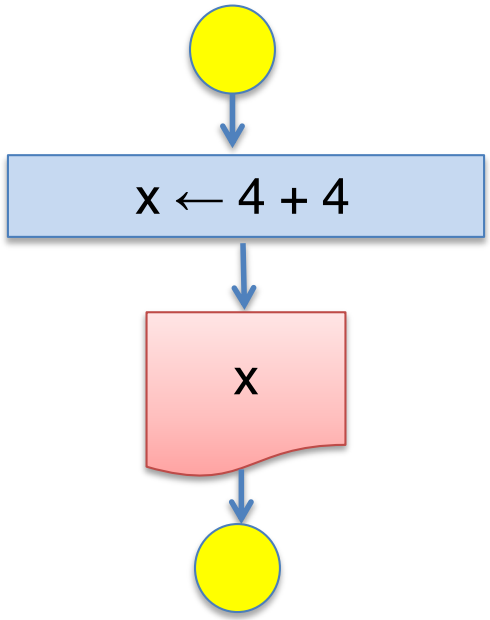



4
4
x ← 4 + 4
x

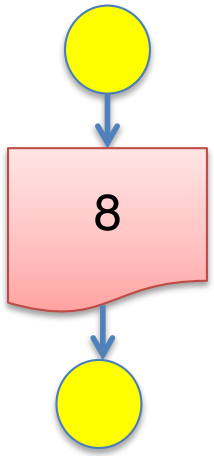



4
4
x ← 4 + 4
x



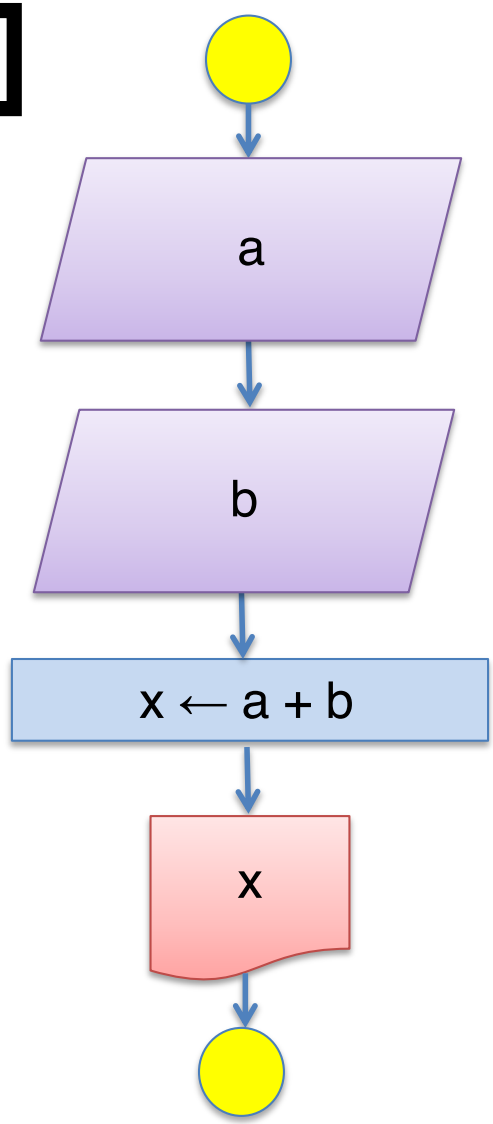

x ← 4 + 4
x



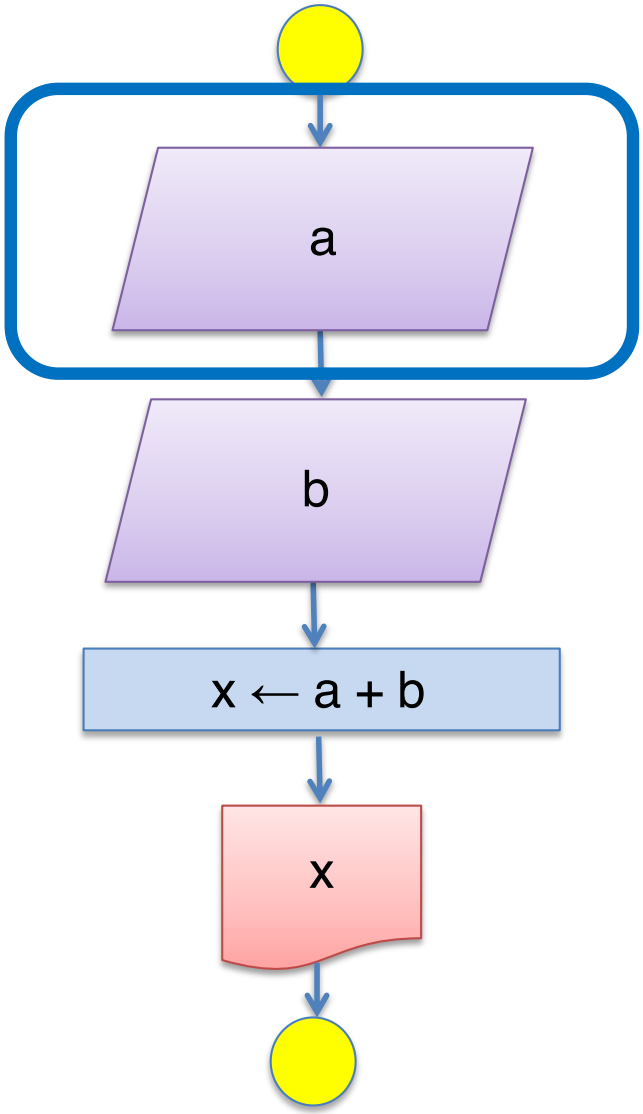

8

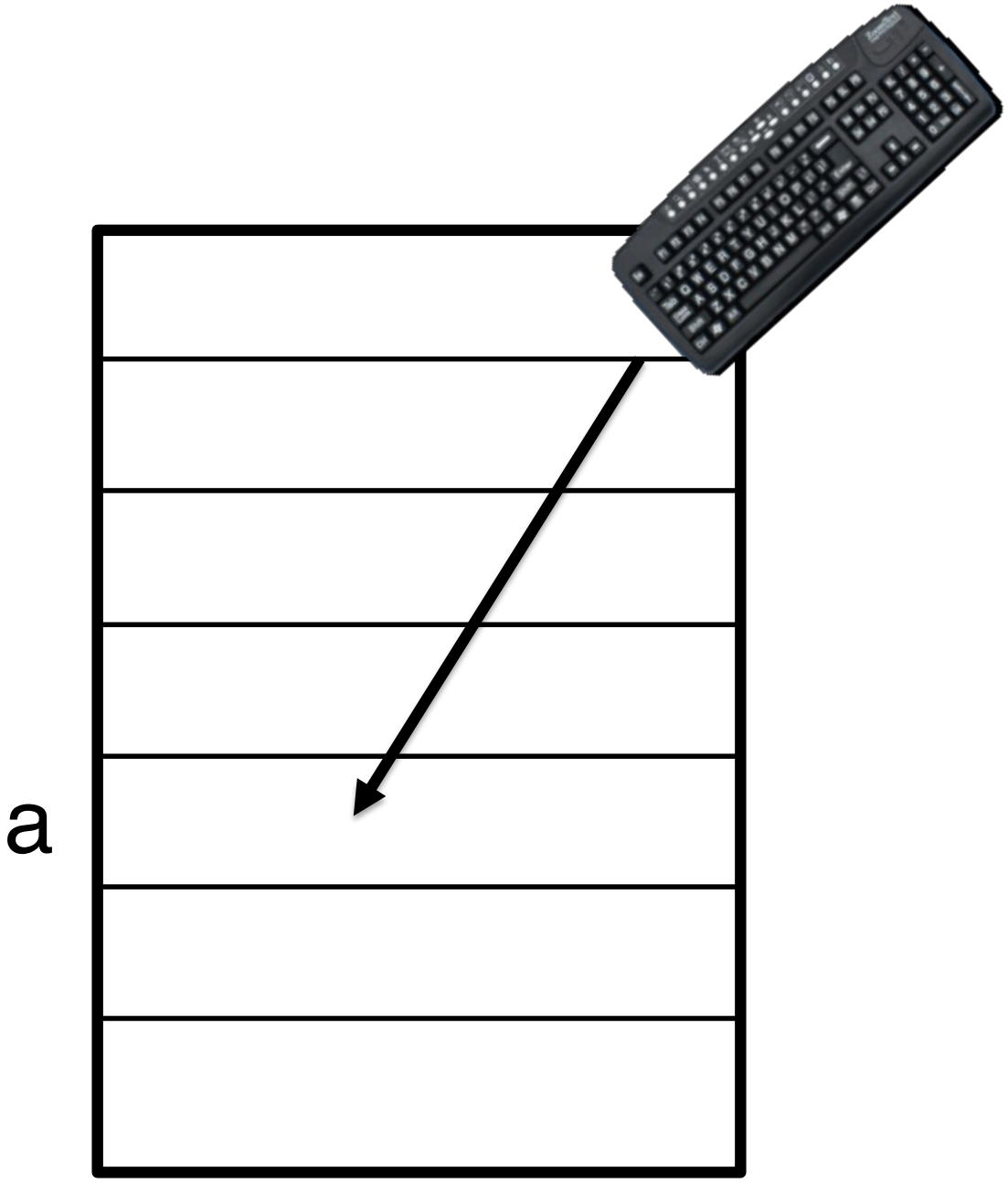


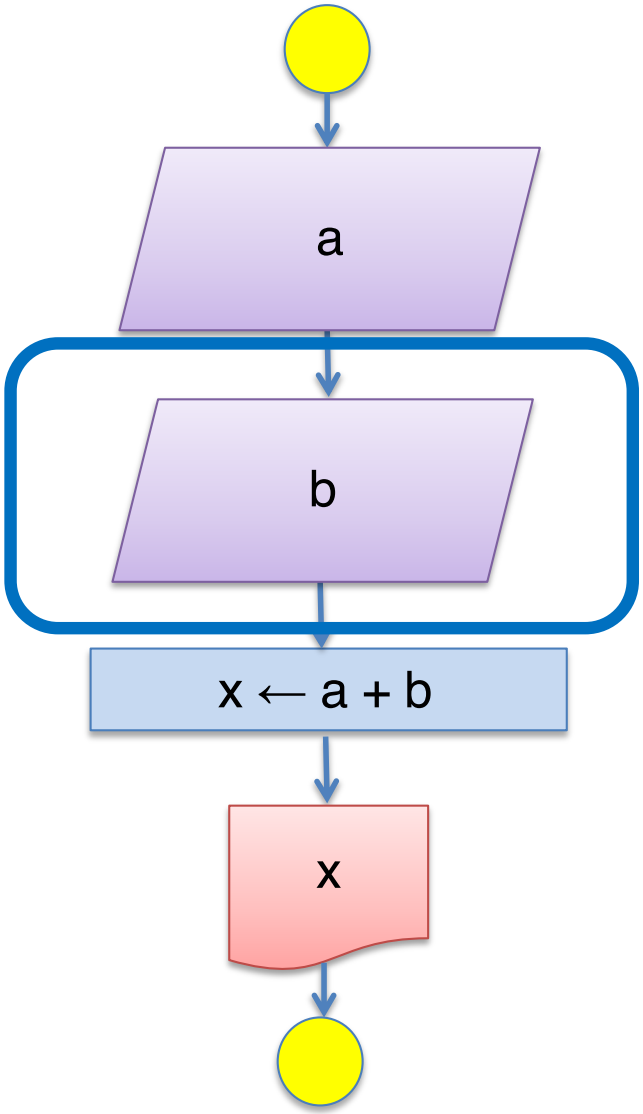
# Parametrizzazione

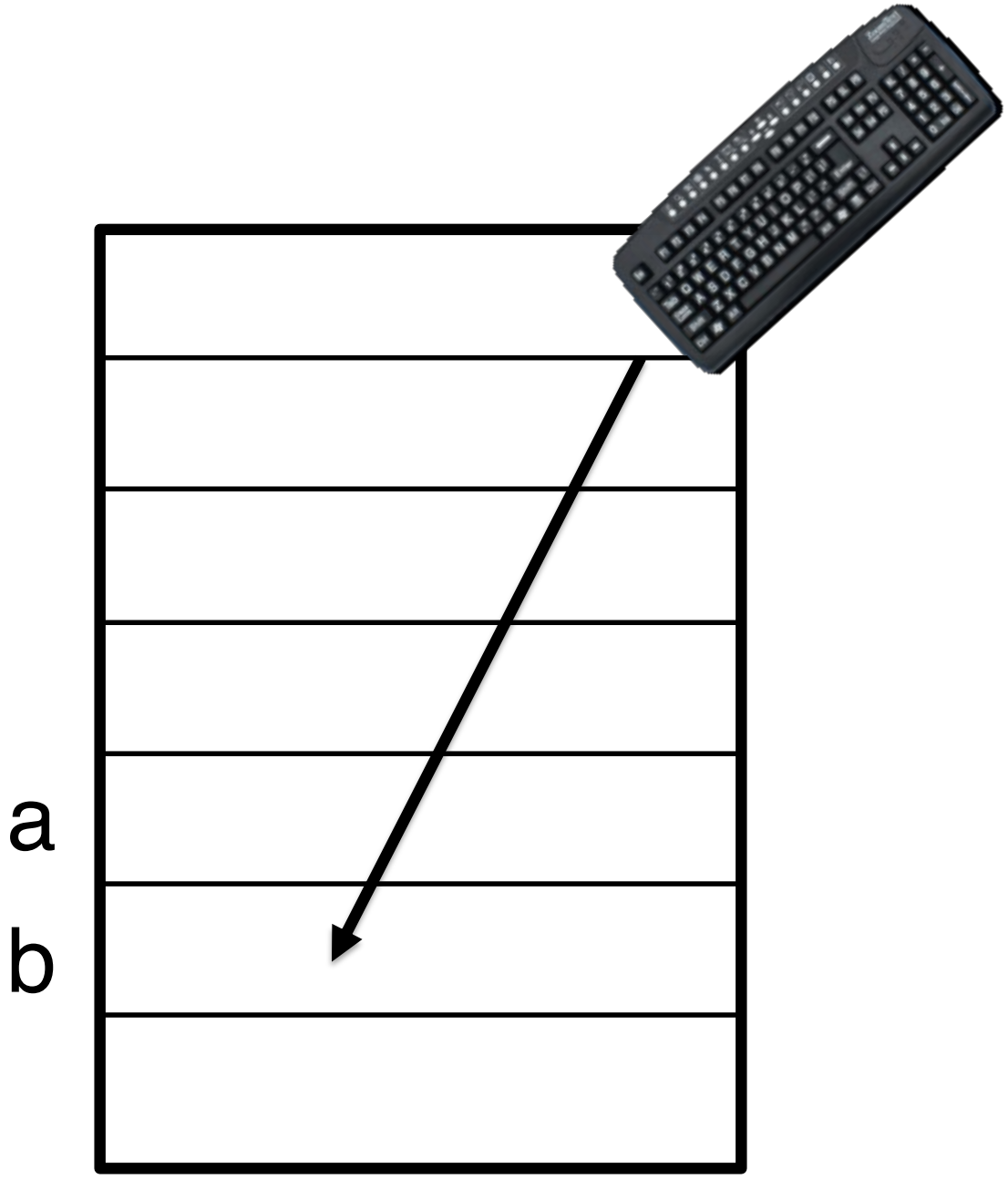


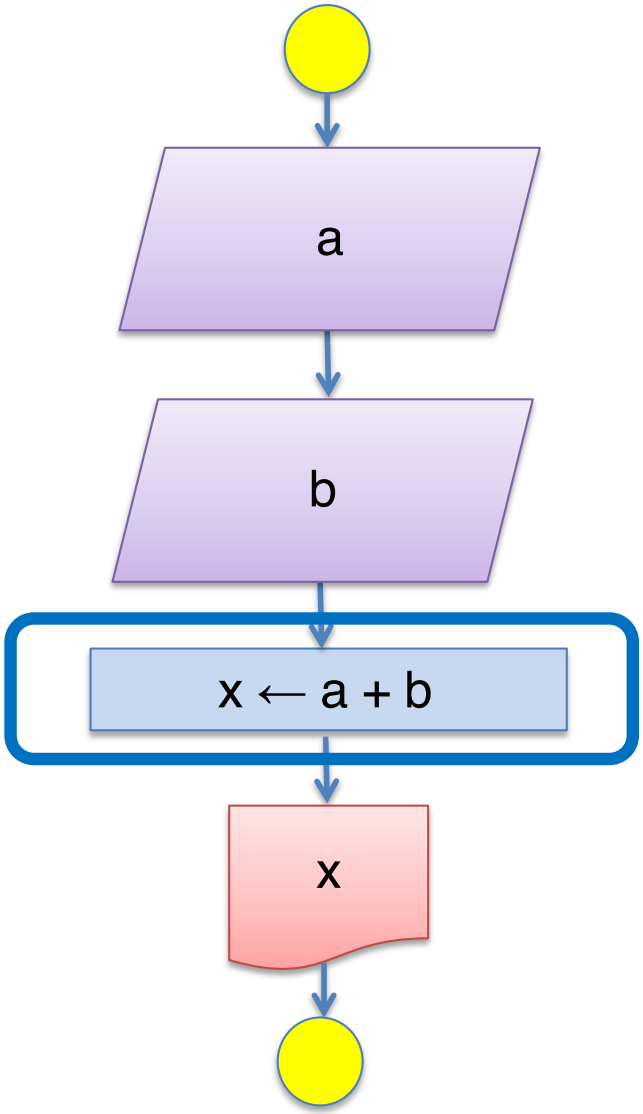


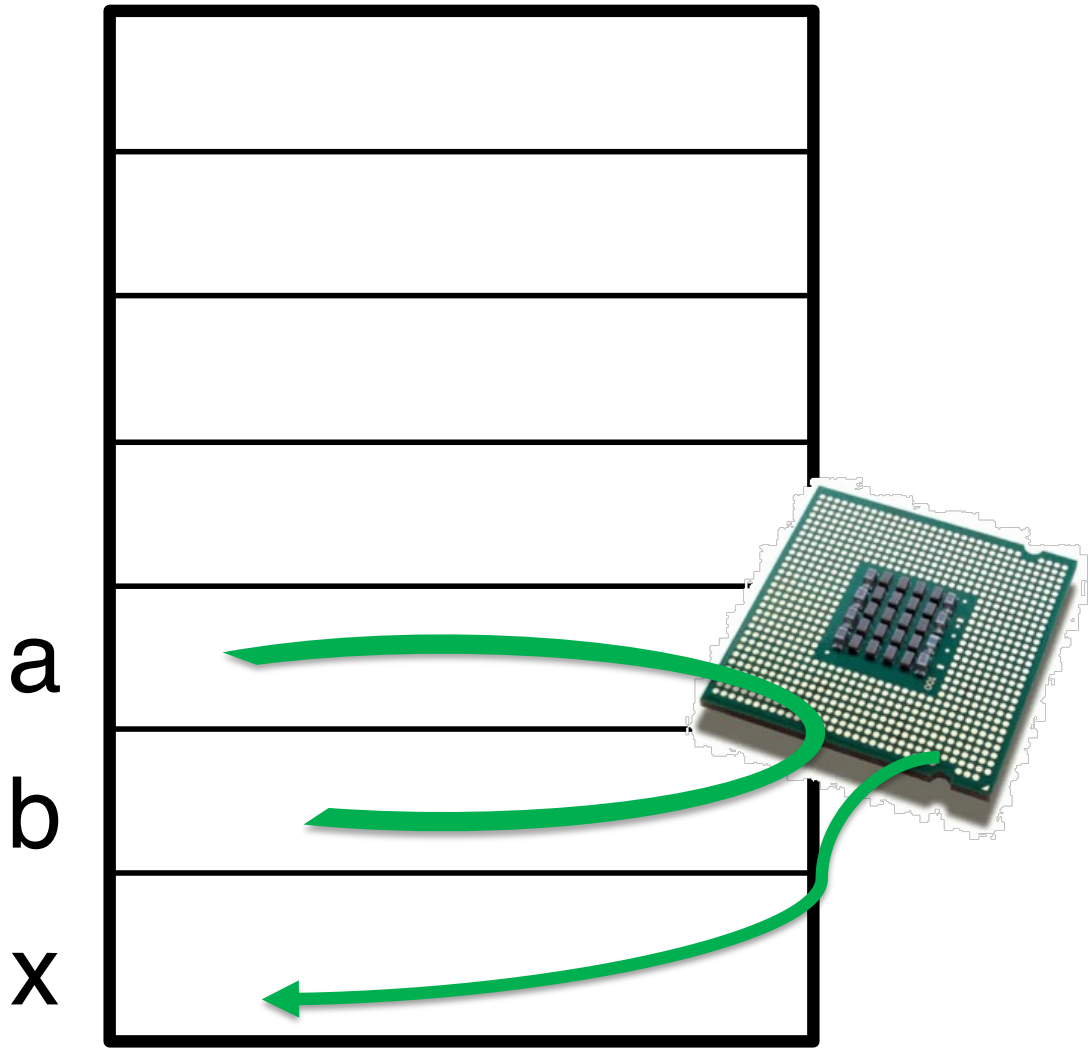


a
b
x ← a + b
x

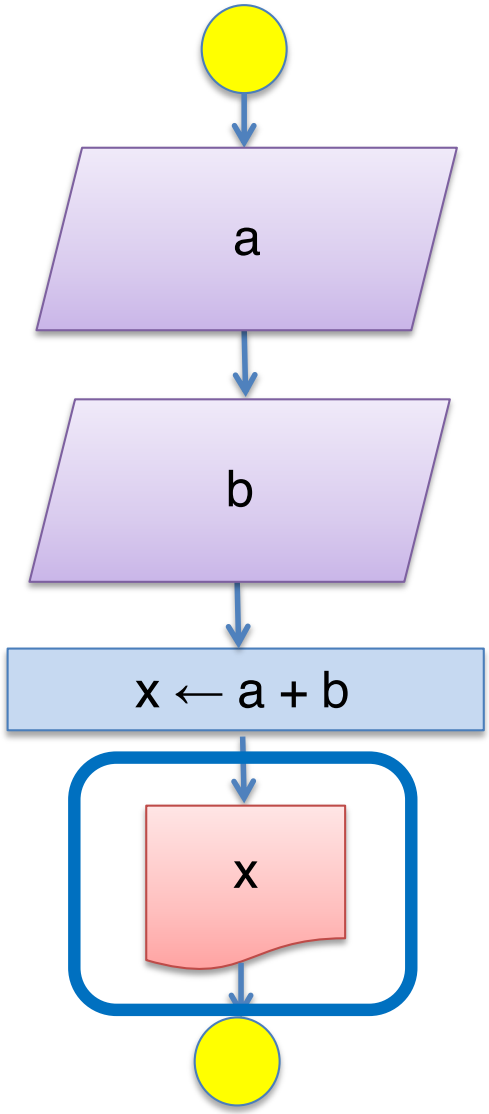

a

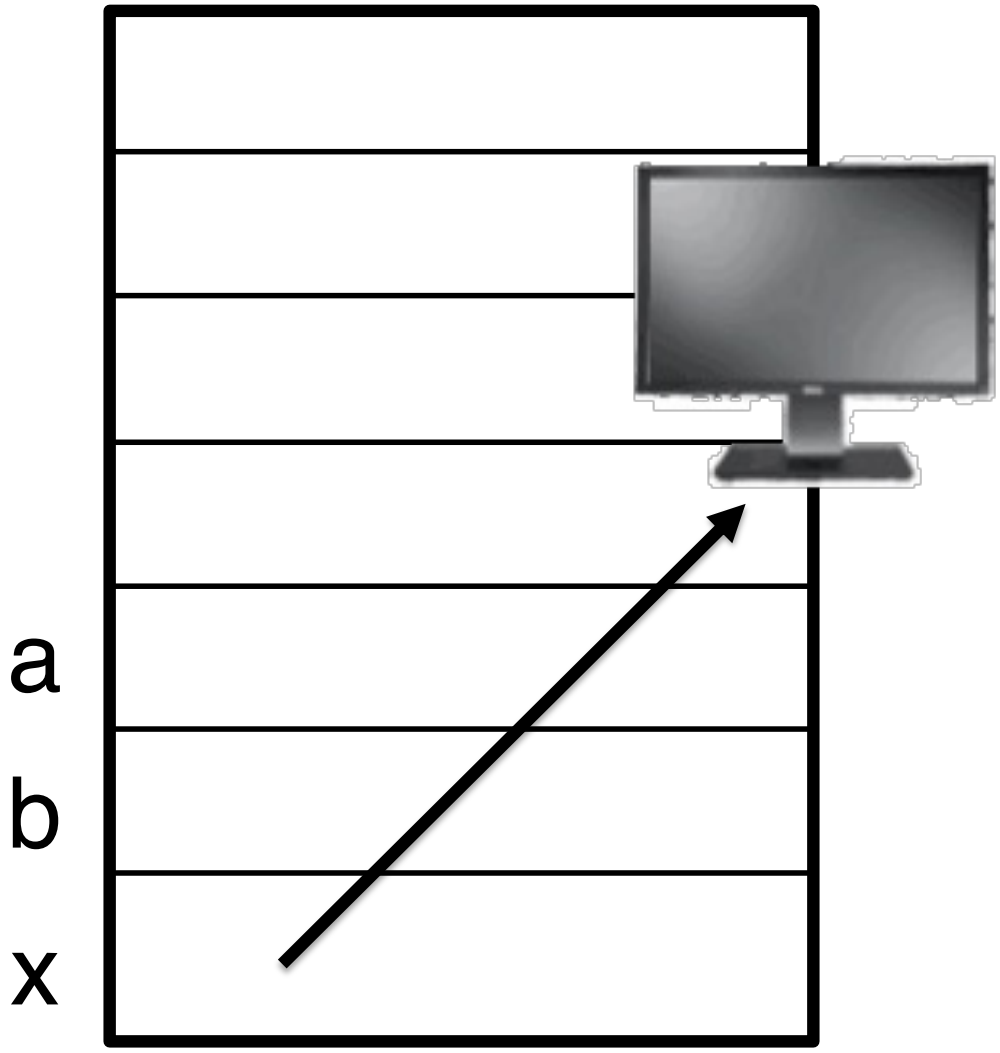

a
b
x ← a + b
x


a
b


a
b
x ← a + b
x


a
b
x

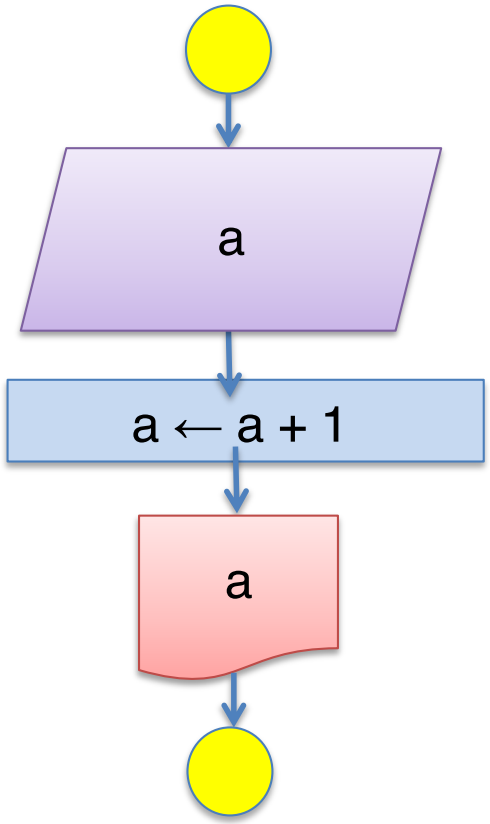

a
b
x ← a + b
x


a
b
x

**Algoritmo per fare a + b.**

## Esercizio

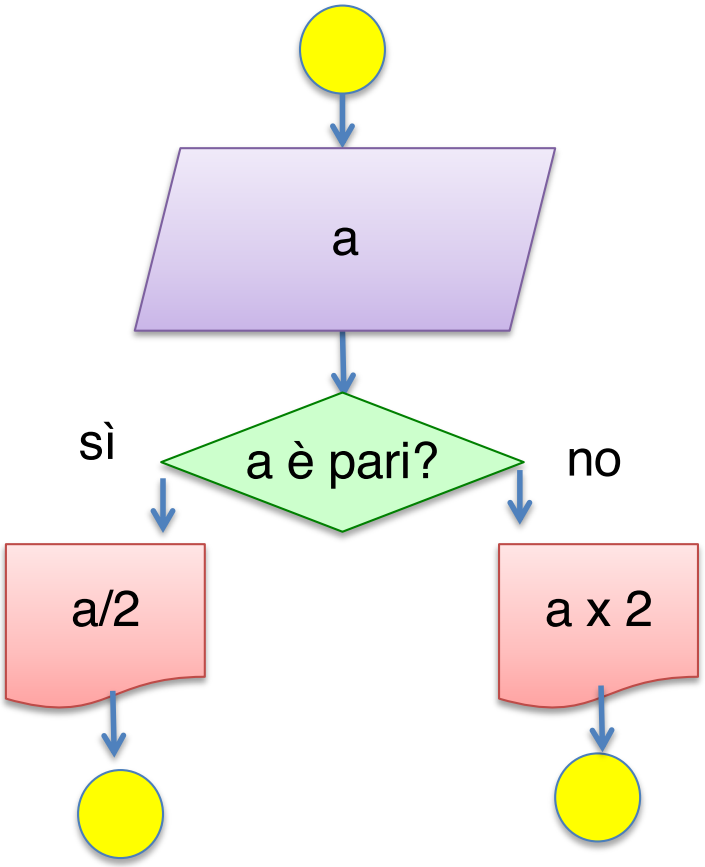
**Fare il diagramma di flusso di un algoritmo che riceve in input un numero e restituisce in output il numero aumentato di 1.**


a
a ← a + 1
a



## Esercizio

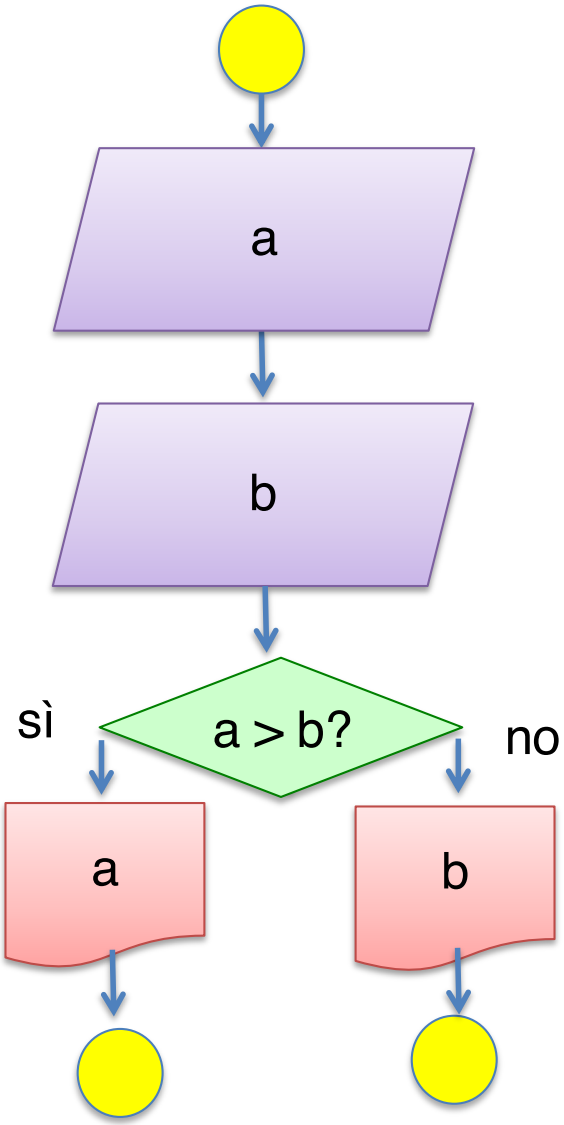
**Fare il diagramma di flusso di un algoritmo che riceve in input un numero e restituisce il suo doppio se il numero è dispari, la sua metà se il numero è pari.**


a
a è pari?
sì
no
a/2
a x 2

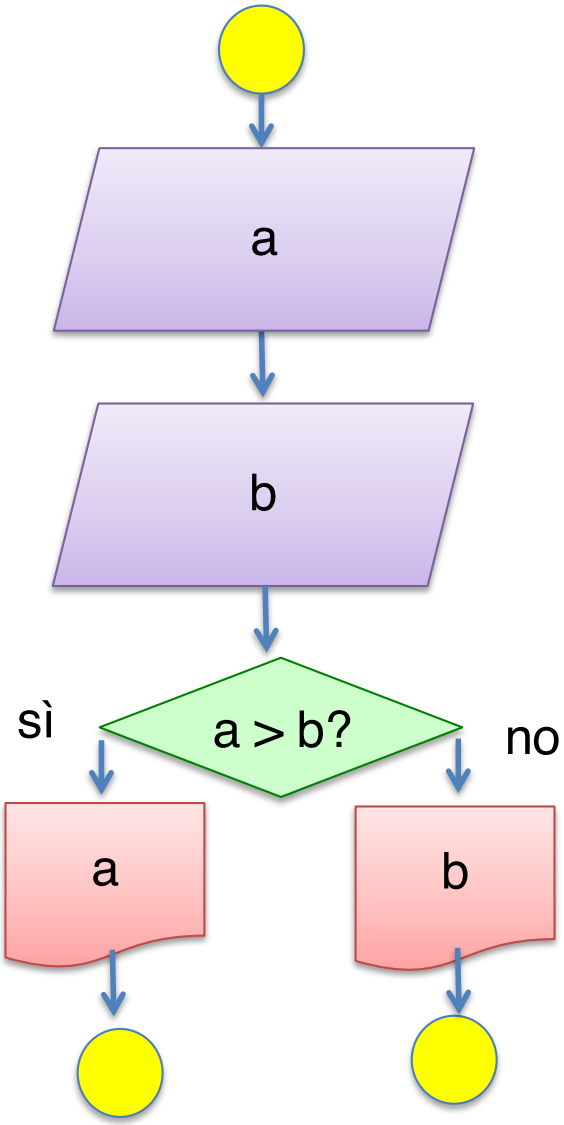


## Esercizio

**Fare il diagramma di flusso di un algoritmo che riceve in input due numeri e restituisce il più grande dei due.**


a
b
sì
a > b?
no
a
b






**e se a = b?**
**I casi limite vanno trattati con attenzione.**



## Esercizio

**Fare il diagramma di flusso di un algoritmo che riceve in input due numeri e restituisce il loro massimo comune divisore.**