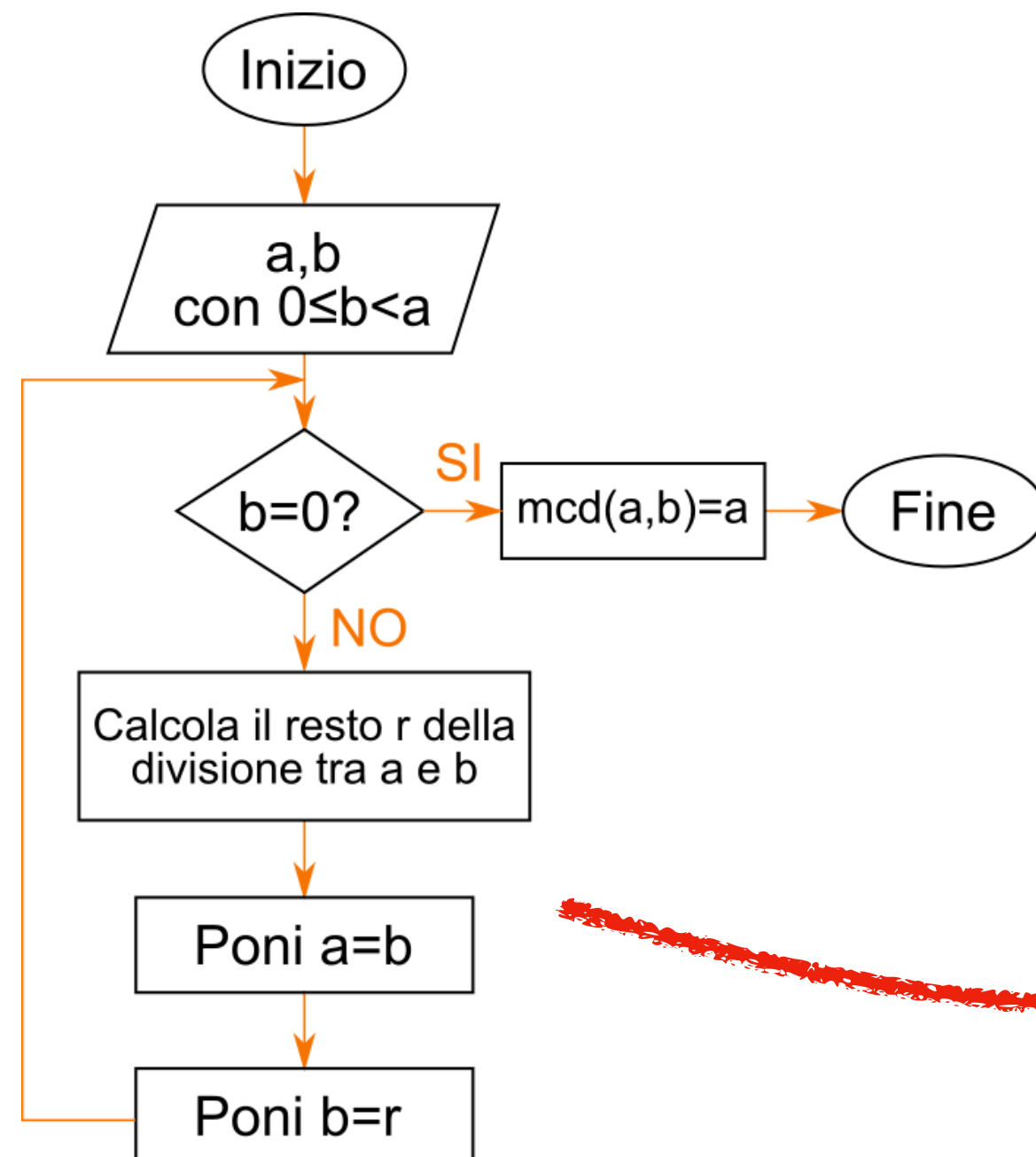


# Lab 2

# Dagli algoritmi ai programmi

## ALGORITMO

Linguaggio alto livello (per Umani)



## LINGUAGGIO di PROGRAMMAZIONE

- Consente la descrizione di algoritmi
- Usando costrutti comprensibili all'essere umano (q.m.)
- Traducibili automaticamente in linguaggio macchina

## ESECUTORE

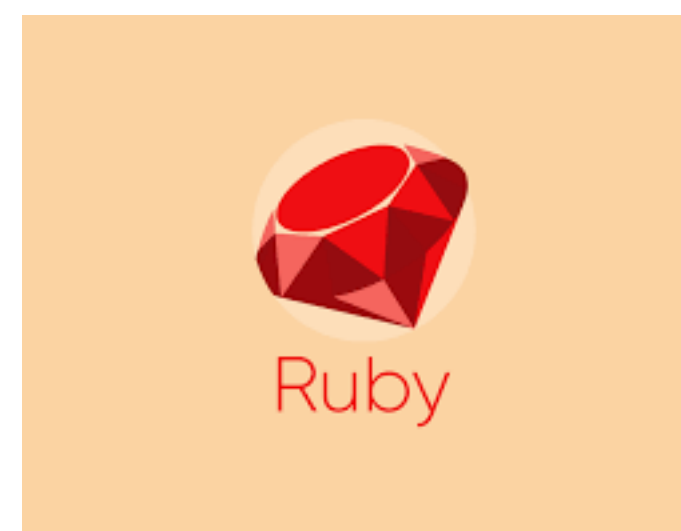
Linguaggio basso livello (per Macchine)



# Linguaggi di Programmazione



...about  
700  
programming languages  
available!!!!



# C++

## Pros e Cons

Come fare a recensire una bicicletta se non si sa che cosa è un cambio?

Per noi un linguaggio varrebbe un altro ma:

- Lo userete in altri corsi (TNDS)
- È abbastanza a basso livello per consentire di vedere che cosa succede in memoria (capire che cosa sta facendo la macchina)
- È abbastanza a basso livello per non fare schifezze di nascosto (ogni riferimento a Python è totalmente voluto)
- (Può essere usato come linguaggio imperativo, evitando gli oggetti)

# Strumenti di Lavoro

**Ambiente:** Linux, Windows, Mac, Android, non importa fintanto che potrete installare ed usare in modo comodo i seguenti strumenti:...

**Editor di Testi:** serve a scrivere il codice sorgente, ovvero un file di testo semplice (no immagini, formattazione strane, Emoji) contenente le istruzioni in linguaggio C++:

- Base (GEdit, VIM, EMACS) NO WORD!!!!
- Integrated Development Environment (IDE): Visual Studio Code, Eclipse, XCode (MAC)

**Compilatore C++:** un programma che prende in ingresso il file di testo prodotto con l'editor e lo traduce in linguaggio macchina, salvando la traduzione in un altro file. Ci sono diversi compilatori a disposizione:

- GNU Compiler Collection (gcc): contiene anche il compilatore C++
- clang (Apple)
- MinGW (porting (adattamento) Windows del gcc)
- Altro....



# Strumenti di Lavoro (continua)

**Ambiente:** non dovete fare i salti mortali. Qualsiasi ambiente abbiate a disposizione, tenetelo! Quello che ci serve si può installare ovunque. Vedi Ariel.

**Editor di Testi:** qui servirebbe un corso. Tenete conto che gli IDE sono molto comodi: highlight delle parole chiave, indentazione automatica...

**Compilatore C++:** si installa ovunque (vedi sopra). Attenzione che noi useremo la compilazione "da riga di comando", quindi cercate di avere i comandi di compilazione "da riga di comando" disponibili (non vale schiacciare un pulsante!)

# Ambiente di Lavoro: LabCalcolo

**Ambiente:** Linux (distribuzione....???)

**Editor di Testi:** installato GEdit, VIM, e l'IDE Visual Studio Code

**Compilatore C++:** gcc (versione 8.4.1)

**"Interfaccia":** useremo una cosa "nuova": la shell di sistema (ne parliamo in lab)

# Workflow

La tipica sessione di laboratorio consiste nello svolgimento di esercizi volti ad allenare la vostra capacità di :

- definire un algoritmo per la soluzione del problema
- verificare la correttezza dell'algoritmo
- determinare i costrutti del linguaggio necessari all'implementazione dell'algoritmo
- scrivere il programma che risolve il problema assegnato
- verificare la correttezza del codice scritto

La parte di scrittura del codice, in particolare, si può così schematizzare



Scritto codice nuovo, tutto ok,  
passo al prossimo pezzo di soluzione



Scritto codice nuovo, errore sintassi/funzionamento,  
provo a correggere

