Definizione di Informatica

Definizione IEEE

"L'informatica è la scienza della rappresentazione e dell'elaborazione dell'informazione"

Scienza = approccio rigoroso e sistematico

Informazione = è parte di qualsiasi attività umana. È il contenuto di una comunicazione o di una trasmissione

Rappresentazione = il problema di astrarre i concetti importanti per modellare la realtà di interesse

Elaborazione = uso e trasformazione dell'informazione in maniera funzionale al raggiungimento di obiettivi.

Al centro dell'informatica vi sono i concetti di:

-informazione : i dati del problema e come vengono rappresentati

-elaborazione: la trasformazione per il raggiungimento di un obiettivo

master COPY

Definizione ACM

"È lo studio sistematico degli algoritmi che descrivono e trasformano l'informazione"

Gli aspetti degli algoritmi di cui tratta sono:

-Teoria

-Analisi

-Progetto

-Efficienza

-Realizzazione

-Applicazione

master COPY COPISTERIA

Un **algoritmo** è una sequenza finita di passi/azioni, definiti con precisione, che portano alla realizzazione di un compito (soluzione di un problema)

Un algoritmo è la descrizione della soluzione di un problema in termini di un insieme di regole (passi) che, operando sui dati iniziali, consentono di ottenere dei risultati che costituiscono la soluzione del problema.

Se l'insieme dei dati di ingresso è finito, l'algoritmo deve terminare deve terminare dopo un numero finito di passi.

Osservazione→ descrivere un algoritmo non ha molto senso se non in relazione a qualcuno/qualcosa che eseguendo i passi dell'algoritmo risolverà il problema

Questo qualcuno/qualcosa prende il nome di **esecutore**

Un esecutore è definito in base a tre elementi:

→insieme delle operazioni che è capace di compiere:

-univocamente definito

-deterministico (porta allo stesso risultato)

→Insieme delle istruzioni che comprende/interpreta:

-definite in modo formale (sintassi)

-costituiscono un linguaggio

→Quali operazioni associare ad ogni istruzione:

master COPY

-l'insieme delle regole che ad ogni costrutto sintatticamente corretto associano le relative operazioni da compiere deve essere univocamente determinato

-caratterizzazione semantica

L'esecutore è sequenziale→ esegue un'istruzione per volta, una dopo l'altra.

Quando l'esecutore è una macchina automatica, il linguaggio non può essere ambiguo→ esistono delle regole sintattiche per descrivere le operazioni dell'esecutore

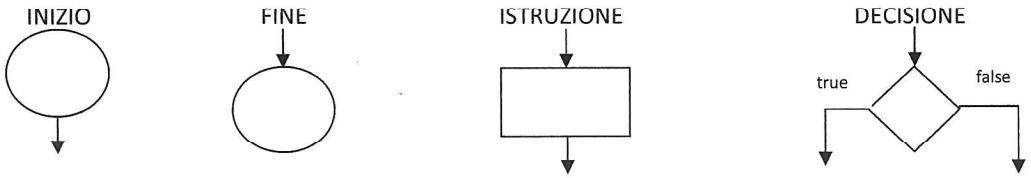
Errori:
-di sintassi→ errore nel descrivere ciò che l'esecutore deve fare; sono segnalati dal programma
-di semantica→ ad esempio invece di scrivere 2+3 scrivo 2*3. Non segnalati dal programma

Proprietà algoritmi:
-correttezza→ raggiungere lo scopo eseguendo l'algoritmo
-efficienza e complessità→ efficienza = numero di operazioni
-leggibilità→ come è stato scritto l'algoritmo
-genericità→ proprietà dell'algoritmo di poter essere utilizzato in diversi problemi

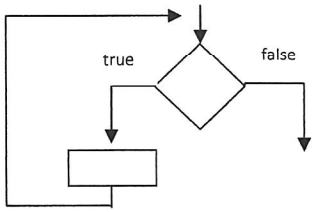
Il calcolatore è un particolare tipo di esecutore.
Il **linguaggio di programmazione** è il linguaggio che il calcolatore è in grado "di comprendere" ossia è in grado di tradurre in operazioni.
Se un algoritmo è descritto in modo da poter essere eseguito da un calcolatore, ossia tramite un linguaggio di programmazione, l'algoritmo prende il nome di programma.

Come scrivere algoritmi e programmi
Per la scrittura di algoritmo si può, in una fase iniziale, utilizzare il linguaggio naturale. Esistono anche dei linguaggi appositi per la loro descrizione. Si descrivono "semplicemente" la sequenza di operazioni (o step) da eseguire per realizzare un compito. Per scrivere il relativo programma, occorrerà tradurre gli "step" dal linguaggio naturale in statement del linguaggio di programmazione, tenendo conto sia della sintassi che della semantica del linguaggio.

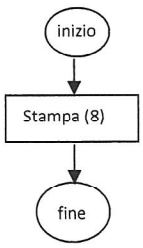
Flow chart (diagrammi di flusso)
Rappresentano il flusso dell'esecuzione cioè come lavora l'esecutore.



CICLI

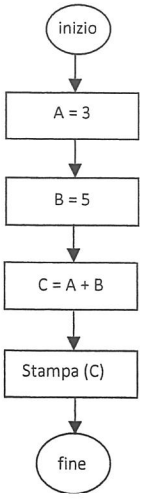


Esempio
Sommare 3 e 5



Stampa = comando che stampa su schermo ciò che c'è tra parentesi

Con questo algoritmo ho sommato 3 e 5 ma il conto l'ho fatto io→ l'algoritmo non ha fatto niente sul conto

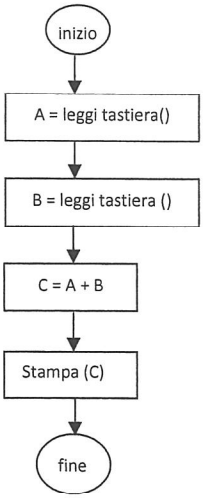


A = 3 operazione di ASSEGNAMENTO →ciò che sta a dx lo assegna a ciò che sta a sx

L'assegnamento non è un'operazione simmetrica! → se mettessi 3 = A il calcolatore vorrebbe prendere il valore di A e assegnarlo a 3, ma A non ha alcun valore: è un errore.

Questo algoritmo ha fatto il suo dovere.
Questo algoritmo è generico perché qualunque siano A e B è in grado di calcolarne la somma. MA ogni volta qual volta voglia fare la somma devo modificare il programma, cambiando gli addendi da sommare.

Voglio scrivere un programma in grado di sommare qualsiasi numero senza doverlo modificare ogni volta.
Ovvero un programma indipendente dai numeri da sommare.

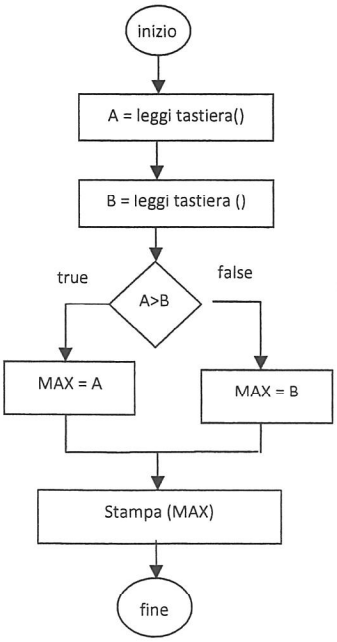


In questo modo quando cambio i numeri non devo modificare il programma

Il flusso dell'esecuzione è sequenziale, lineare: una istruzione dopo l'altra.

master COPY COPISTERIA

<u>Esempio</u>
Calcolare il massimo tra A e B
Non posso scriverlo in maniera lineare poiché ci sono due soluzioni alternative. Uso comando DECISIONE.

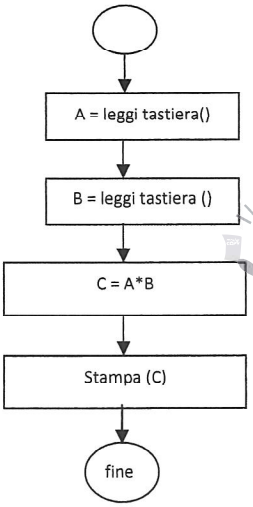


master COPY

IMPORTANTE→ devo verificare che l'algoritmo funzioni anche per A = B.
è corretto perché ad esempio 2 > 2 è falsa, segue il ramo false, e stampa B cioè il massimo.
Se A = B cambia il ramo che segue, ma da il risultato giusto

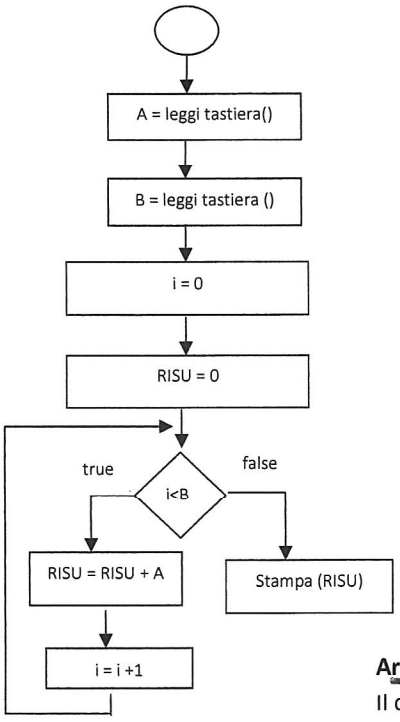
Esempio

Prodotto tra due numeri



Usando questo algoritmo do per scontato che il calcolatore sappia: leggere da tastiera, fare il prodotto e stampare sullo schermo

Supponiamo che il calcolatore non sappia fare il prodotto ma solo le somme



A = 3 B = 1 faccio un solo ciclo ; A =3 B =2 faccio due cicli.

A = 3 B=2 Verifico la condizione del ciclo:

i<B true→ RISU = RISU + A = 0 + 3 = 3

i = i +1 = 0 + 1 = 1

riverifico la condizione del ciclo: i <B ? 1<2 si

RISU = RISU + A = 3 + 3 = 6

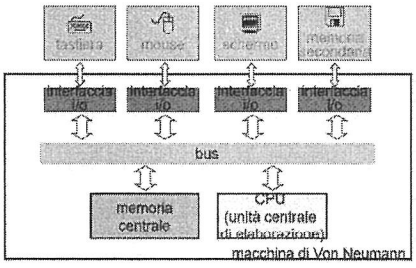
i = i +1 = 1 + 1 = 2

riverifico la condizione del ciclo: i <B ? 2<2 no → esco dal ciclo

stampo RISU = 6

Architettura del calcolatore

Il calcolatore adotta l'architettura di Von Neumann



Caratteristiche:
Venne sviluppata negli anni 50 ma è ancora attuale. è l'architettura del calcolo sequenziale. Dati e istruzioni sono memorizzati in una memoria unica che permette sia la lettura che la scrittura. I contenuti della memoria sono indirizzati in base alla loro posizione, indipendentemente dal tipo di dato o istruzione contenuto.
Le istruzioni vengono eseguite in modo sequenziale.
Elementi dell'architettura:
-La CPU:
è l'esecutore. Esegue gli step dell'algoritmo/programma coordinando la comunicazione fra i vari componenti.
La CPU deve essere veloce, ma dato che per ogni operazione deve accedere alla memoria principale anche la memoria deve essere veloce.
-La memoria principale:
Contiene le istruzioni e i dati che costituiscono un programma.
È volatile: quando si toglie l'alimentazione tutti i dati e i programmi vengono cancellati.
È veloce
-Il bus:
è un insieme di fili che permette lo scambio di dati tra i vari elementi del sistema. CPU,memoria principale e dispositivi I/O sono connessi da un unico bus, insieme di fili su cui viaggiano i segnali.
Caratteristica dell'architettura di Van Neumann è che si usa un unico bus per connettere tutti gli elementi.
-L'insieme dei dispositivi di I/O (input/output):
Sono indispensabili in quanto consentono alla CPU di dialogare con il mondo esterno. Consentono di acquisire dati dal mondo esterno e rendono visibili i risultati dell'elaborazione.
Tastiera, mouse dispositivi di input. Video dispositivo di output. Scheda di rete dispositivo di input e output.

Nell'insieme è un sistema: insieme di componenti interagenti fra di loro

La CPU utilizza la memoria principale per leggere dati e programmi. È veloce ma è volatile, ovvero se ne perde il contenuto quando non è alimentata. È necessario che nel calcolatore sia presente un supporto di memorizzazione non volatile (memoria di massa).
Memorie di massa
In essa sono contenuti i programmi e i dati. All'atto dell'avvio di una applicazione, il programma e i dati vengono trasferiti dal supporto di massa alla memoria principale e la CPU inizia ad eseguire le istruzioni. Il sistema operativo (un particolare programma) si occupa del trasferimento dalla memoria di massa alla memoria centrale.
Le memorie di massa sono più lente ma più economiche della memoria principale.
L'hard disk è un tipo di memoria di massa, la principale, sempre presente in un calcolatore. Esistono anche altri tipi di supporto: dischi ottici, nastri magnetici, floppy disk.
Il disco rigido (hard disk)
Sistema di memorizzazione: non volatile ma lento (non ha la stessa velocità della memoria principale).
Dato che l'HDD è lento il calcolatore legge i programmi dalla memoria centrale e non direttamente dall'hard disk altrimenti la CPU sarebbe lenta.
La memoria principale è costosa mentre l'HDD è economico.

