

Informatica

Esistono varie definizioni:

- “Scienza dei calcolatori elettronici (*Computer Science*)”
- “Scienza dell’informazione”

Definizione proposta:

“Scienza della rappresentazione e dell’elaborazione dell’informazione.”

Calcolatore Elettronico:

è uno **strumento** per la rappresentazione e l’elaborazione delle informazioni.

Fanno parte dell’Informatica:

- metodi per la rappresentazione delle informazioni
- metodi per la rappresentazione di procedimenti risolutivi
- linguaggi di programmazione
- architettura dei calcolatori
- sistemi operativi
- ecc.

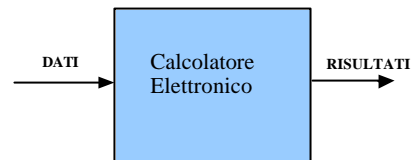
Risoluzione di Problemi mediante l’uso del Calcolatore Elettronico

Calcolatore elettronico:

- è uno strumento in grado di eseguire insiemi di *azioni* elementari.
- le azioni vengono eseguite su oggetti (*dati*) per produrre altri oggetti (*risultati*).
- L’esecuzione di azioni viene richiesta all’elaboratore attraverso opportune direttive, dette **istruzioni**.

Programmazione

È l’attività con cui si predispone l’elaboratore ad eseguire un particolare insieme di azioni su particolari dati, allo scopo di risolvere un certo problema.



I Problemi da risolvere

I problemi che siamo interessati a risolvere con l’elaboratore sono di natura molto varia.

Esempi:

- 1) Dati due numeri trovare il maggiore
- 2) Dato un elenco di nomi e relativi numeri di telefono trovare il numero di telefono di una determinata persona
- 3) Problema del Lupo, della Capra e del Cavolo
- 4) Dati a e b , risolvere l’equazione $ax + b = 0$
- 5) Stabilire se una parola viene alfabeticamente prima di un’altra
- 6) Somma di due numeri interi
- 7) Scrivere tutti gli n per cui l’equazione:
$$X^n + Y^n = Z^n$$
ha soluzioni intere (problema di Fermat)
- 8) Ordinare una lista di elementi
- 9) Calcolare il massimo comun divisore fra due numeri dati
- 10) Calcolare il massimo in un insieme

☞ La descrizione del problema non fornisce (in genere) un metodo per calcolare il risultato

Risoluzione di Problemi

- Affinché un problema sia risolvibile, in generale è necessario che la sua definizione sia chiara e completa.
- Non tutti i problemi sono risolvibili attraverso l’uso del calcolatore. In particolare esistono classi di problemi per le quali la soluzione automatica non è proponibile. Ad esempio:
 - se il problema presenta infinite soluzioni.
 - per alcuni dei problemi **non è stato trovato** un metodo risolutivo.
 - per alcuni problemi è stato dimostrato che **non esiste** un metodo risolutivo automatizzabile
- Noi ci concentreremo sui problemi che, ragionevolmente, ammettono un metodo risolutivo (**funzioni calcolabili**).
- Uno degli obiettivi del Corso è quello di presentare le **tecnologie** e le **metodologie** di programmazione:
Tecnologie: strumenti per la realizzazione di programmi
Metodologie: metodi per l’utilizzo in modo corretto ed efficace delle tecnologie di programmazione

Risoluzione di un problema

Con questo termine si indica il processo che:

- dato un **problema**, e
- individuato un **metodo risolutivo**

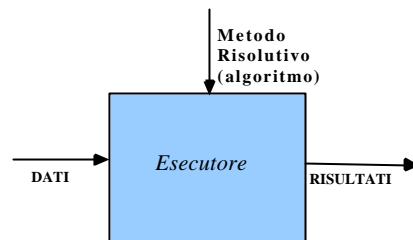
trasforma i dati iniziali nei corrispondenti risultati finali.

Affinché la risoluzione di un problema possa essere realizzata attraverso l'uso del calcolatore, tale processo deve poter essere definito come **sequenza di azioni elementari**.

ALGORITMO

L'algoritmo è l'insieme ordinato delle azioni che risolve un dato problema.

- L'algoritmo descrive il metodo risolutivo attraverso un insieme ordinato di azioni.
- L'esecuzione delle azioni secondo l'ordine specificato dall'algoritmo, a partire dai dati di ingresso, consente di ottenere i risultati relativi alla soluzione del problema.



☞ Si fa riferimento ad un “**esecutore**”: con questo termine si intende una macchina astratta (non necessariamente un calcolatore) in grado di eseguire le azioni specificate dal metodo risolutivo.

Proprietà fondamentali dell'Algoritmo

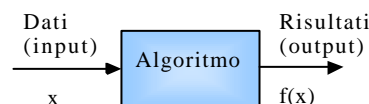
- 1• **Eseguibilità**: ogni “istruzione” deve essere eseguibile da parte dell'*esecutore* dell'algoritmo;
- 2• **Non Ambiguità**: ogni azione deve essere univocamente interpretabile dall'*esecutore*;
- 3• **Finitezza**: il numero totale di azioni da eseguire, per ogni insieme di dati di ingresso, è finito.

♦ L'algoritmo deve:

- essere applicabile a qualsiasi insieme dei dati di ingresso appartenenti al **dominio di definizione dell'algoritmo**;
- essere costituito da operazioni appartenenti ad un determinato **insieme di operazioni fondamentali** (sistema formale);
- le regole che costituiscono l'algoritmo non devono essere ambigue, cioè devono essere interpretabili in modo **univoco** qualunque sia la persona o la “macchina” che le legge.

Algoritmo

- Altre proprietà *desiderabili*:
 - generalità
 - terminazione
 - determinismo
 - efficienza
- Un algoritmo può essere visto come il procedimento di calcolo di una funzione che mappa uno o più valori di un **dominio** A (input dell'algoritmo) in un valore del **codominio** B (output dell'algoritmo).



Algoritmo: un po' di storia

- Il termine "algoritmo" deriva dal nome del matematico arabo *Al-Khowarizmi* del IX secolo d.c. che per primo suggerì un metodo per sommare due numeri rappresentati nel sistema numerico Hindu e contribuì alla fondazione dell'algebra.
- Nel medioevo il termine *algorismus* servì ad indicare il complesso di operazioni nel calcolo numerico con numeri arabi.
- Attualmente, con il termine **algoritmo** si indica la sequenza finita di passi effettuabili da una macchina per risolvere una classe di problemi in tempo finito.

Algoritmi e Programmi

- Ogni elaboratore è una macchina in grado di eseguire azioni elementari su oggetti detti **dati**.
- L'esecuzione delle azioni è richiesta all'elaboratore tramite comandi elementari chiamati **istruzioni**.
- Le istruzioni sono espresse attraverso un opportuno formalismo:

il linguaggio di programmazione.

Es:

```
SUM A,B
read(x)
scanf("%d",&Y)
```

- La formulazione testuale di un algoritmo in un linguaggio comprensibile ad un elaboratore è detta **programma**.

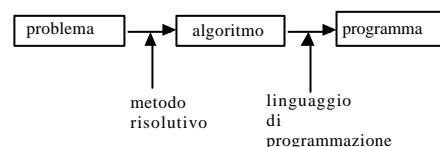
```
main()
{int A, B;

printf("Immettere i dati (A,B): ");
scanf("%d %d", &A, &B);
printf("Risultato: A+B=%d\n", A+B);
}
```

Algoritmi e Programmi

Dato un problema, la sua soluzione può essere ottenuta mediante l'uso del calcolatore, compiendo i seguenti passi:

- 1) individuazione di un procedimento risolutivo
- 2) scomposizione del procedimento in insieme ordinato di azioni → **ALGORITMO**
- 3) rappresentazione dei dati e dell'algoritmo attraverso un formalismo comprensibile per l'elaboratore (linguaggio di programmazione → **PROGRAMMA**)



Il procedimento è complesso e non è univoco

Algoritmi: Alcuni Esempi

Il Problema della capra, del lupo e del cavolo:

- porta la capra sull'altra sponda
- torna indietro
- porta il cavolo sull'altra sponda
- porta indietro la capra
- porta il lupo sull'altra sponda
- torna indietro
- porta la capra sull'altra sponda

Soluzione dell'equazione $ax + b = 0$

- leggi i valori di a e b
- calcola $-b$
- dividi quello che hai ottenuto per a e chiama x il risultato
- stampa x

Stabilire se una parola P viene alfabeticamente prima di una parola Q

- leggi P, Q
- **ripeti quanto segue:**
 - se prima lettera di $P <$ prima lettera Q
allora scrivi vero
 - altrimenti** se prima lettera $P > Q$
allora scrivi falso
 - altrimenti** (le lettere sono $=$)
togli da P e Q la prima lettera
 - fino a quando hai trovato le prime lettere diverse.**

Calcolo del massimo di un insieme:

- scegli un elemento come massimo provvisorio max
- per ogni elemento i dell'insieme: se $i > max$ eleggi i come nuovo massimo provvisorio max
- il risultato è max

NOTA: si utilizzano **variabili** cioè nomi simbolici usati nell'algoritmo per denotare dati.

Algoritmi equivalenti

Due algoritmi si dicono **equivalenti** quando:

- hanno lo stesso dominio di ingresso;
- hanno lo **stesso dominio di uscita**;
- in corrispondenza degli stessi valori nel dominio di ingresso **producono gli stessi valori nel dominio di uscita**

Due algoritmi equivalenti:

- forniscono lo stesso risultato
- possono avere differente efficienza
- possono essere profondamente diversi

Algoritmi Equivalenti: Esempio

Calcolo del massimo comun divisore fra m ed n

- **Algoritmo a :**
 - calcola l'insieme I dei divisori di m
 - calcola l'insieme J dei divisori di n
 - calcola l'insieme K dei divisori comuni:
 $K = I \cap J$
 - calcola il massimo in K : questo è il risultato
- **Algoritmo b :** si basa sul **metodo di Euclide**:

$$\begin{aligned} \text{mcd}(m,n) &= m \quad (\text{oppure } n) && \text{se } m=n \\ \text{mcd}(m,n) &= \text{mcd}(m-n, n) && \text{se } m>n \\ \text{mcd}(m,n) &= \text{mcd}(m, n-m) && \text{se } m<n \end{aligned}$$

Algoritmo:

- Finché m è diverso da n esegui le seguenti azioni:
 - se $m>n$ sostituisci a m il valore $(m-n)$
 - altrimenti sostituisci a n il valore $(n-m)$
- Il massimo comun divisore è n

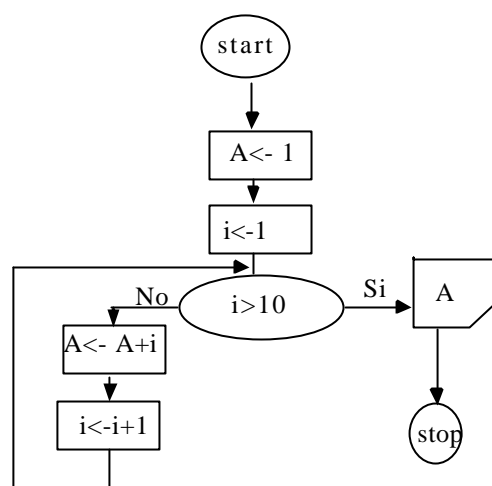
Gli algoritmi a e b sono equivalenti

Rappresentazione di Algoritmi: Diagrammi di flusso

È un formalismo che consente di rappresentare graficamente gli algoritmi.

- un **diagramma** di flusso descrive le azioni da eseguire ed il loro ordine di esecuzione.
- Ogni azione elementare corrisponde ad un simbolo grafico (blocco) diverso (sono convenzioni non universali).
- Ogni blocco ha un ramo in ingresso ed uno o più rami in uscita; collegando tra loro i vari blocchi attraverso i rami, si ottiene un diagramma di flusso.
- Un diagramma di flusso appare, quindi, come un insieme di blocchi di forme diverse che contengono le istruzioni da eseguire, collegati fra loro da linee orientate che specificano la sequenza in cui i blocchi devono essere eseguiti (flusso del controllo di esecuzione).

Diagrammi di Flusso



Diagrammi di Flusso

- L'insieme dei dati di ingresso e dei risultati vengono rappresentati attraverso dei nomi simbolici, detti **variabili** (ad esempio, A)
- 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.
- I diagrammi di flusso hanno notevoli limiti per la soluzione di problemi di una certa complessità.

Diagrammi di Flusso

Valori:

Numerici: interi e reali

Logici: Vero e Falso

Alfanumerici (stringhe): "AAAA", "C. Colombo"

Grandezze:

- **Costanti:** Quantità note a priori, il cui valore non cambia durante l'esecuzione
- **Variabili:** Rappresentate da un nome simbolico cui è assegnato un valore che può cambiare durante l'esecuzione dell'algoritmo.

Espressioni:

- Sequenze di variabili e costanti combinate fra loro mediante operatori (ad es., operatori **aritmetici**: +, -, *, /); ad esempio:

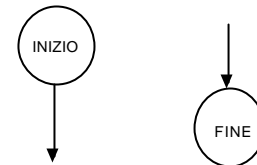
$s + r * 5$
A
100

- Nella valutazione di una espressione, si sostituisce ad ogni variabile il suo valore attuale e si eseguono le operazioni secondo un ordine prestabilito da regole di precedenza (possono comparire parentesi).
- A tutte le variabili che compaiono nell'espressione deve essere stato associato un valore prima della valutazione dell'espressione
- Espressioni relazionali e logiche: danno come risultato vero o falso (>, <, =, ≥, ≤, ≠)

Istruzioni (*blocchi*) fondamentali

Inizio e fine esecuzione (start e stop)

- Inizio** è il blocco da cui deve iniziare l'esecuzione (uno solo).
- Il blocco **fine** fa terminare l'esecuzione dell'algoritmo (almeno uno).

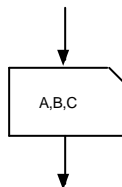


Ingresso (lettura, read, input)

Esecuzione dell'istruzione:

Si ricevono dall'unità di ingresso (per esempio, la tastiera) tanti valori quante sono le variabili specificate all'interno del blocco, e si assegnano nello stesso ordine alle variabili.

- A, B, C sono nomi di **variabili**.

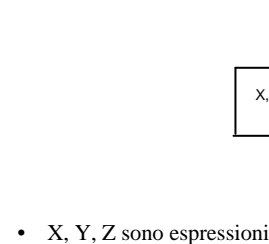


- ♦ *“Leggi i tre valori dati in ingresso, ed assegnali rispettivamente alle variabili A, B, e C.”*

Uscita (stampa, type, print, output)

Esecuzione:

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



- X, Y, Z sono espressioni

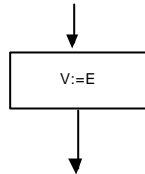
- ♦ *“Calcola i valori delle espressioni X, Y e Z, e trasmettili in uscita.”*

N.B.: I valori di X, Y, Z non vengono alterati dall'esecuzione del blocco.

Assegnamento

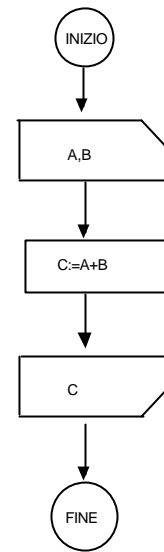
Esecuzione:

Si calcola il valore dell'espressione a destra del simbolo " $:=$ " e lo si assegna alla variabile indicata a sinistra del simbolo " $:=$ " (con eventuale perdita del valore precedente di V)



- V è il nome di una variabile, E è una espressione.
- ♦ “Calcola il valore dell’espressione E ed assegnalo alla variabile V”

Esempio (sequenza):

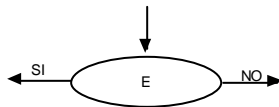


Condizioni o salti condizionati

Esecuzione:

Si valuta la condizione specificata all'interno del blocco: se è verificata, si prosegue con la linea di flusso contrassegnata da "SI", altrimenti (se non è verificata) si prosegue per il ramo etichettato con "NO".

- E è un'espressione *relazionale* (o *logica*): restituisce valore **vero**, oppure **falso**.

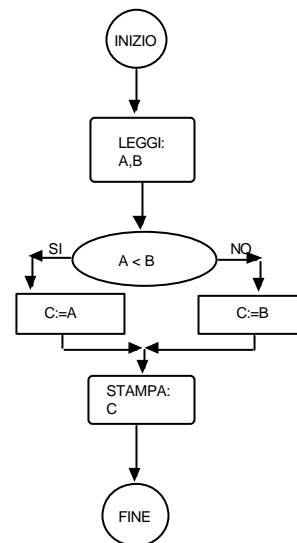


- ♦ “Calcola il valore dell’espressione E: se è vero, prosegui per il ramo SI, altrimenti prosegui per il ramo NO”

Il blocco condizione è l'elemento di base per realizzare **alternative** e **ripetizioni**.

Alternativa

Esprime la scelta tra due possibili azioni (o sequenze di azioni) mutuamente esclusive:



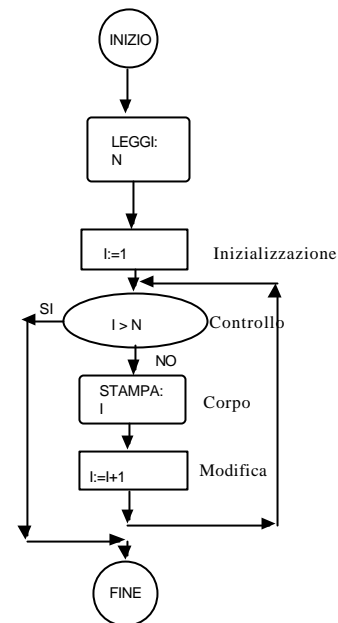
Iterazione o Ripetizione

Esprime la ripetizione di una sequenza di istruzioni.

Nel caso più generale, è costituita da:

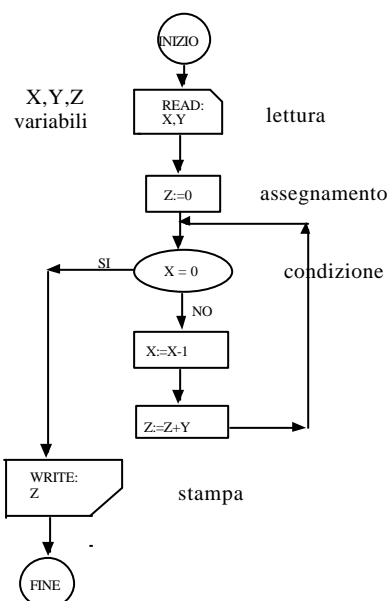
- **Inizializzazione:** assegnazione dei valori iniziali alle variabili caratteristiche del ciclo (viene eseguita una sola volta);
- **Corpo:** esecuzione delle istruzioni fondamentali del ciclo che devono essere eseguite in modo ripetitivo;
- **Modifica:** modifica dei valori delle variabili che controllano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

Iterazione



Esempio:

Algoritmo che calcola il prodotto come sequenza di somme (si suppone $Y > 0$, $X \geq 0$).



Esercizi Proposti:

- 1) Date le lunghezze dei tre lati di un triangolo, progettare l'algoritmo che determina il tipo di triangolo che essi rappresentano: equilatero, isoscele o scaleno.
- 2) Data in ingresso una sequenza di N interi, progettare l'algoritmo che determina il massimo tra i valori dati.