

INFORMATICA DI BASE

- 6 crediti -

Docente: Michele Piana

Email: michele.piana@univr.it

URL: <http://www.di.univr.it/~piana>

Ricevimento: Lunedì ore 14:00 – 15:00

Studio: 2.05 (Ufficio del Preside)

TANTO PER COMINCIARE

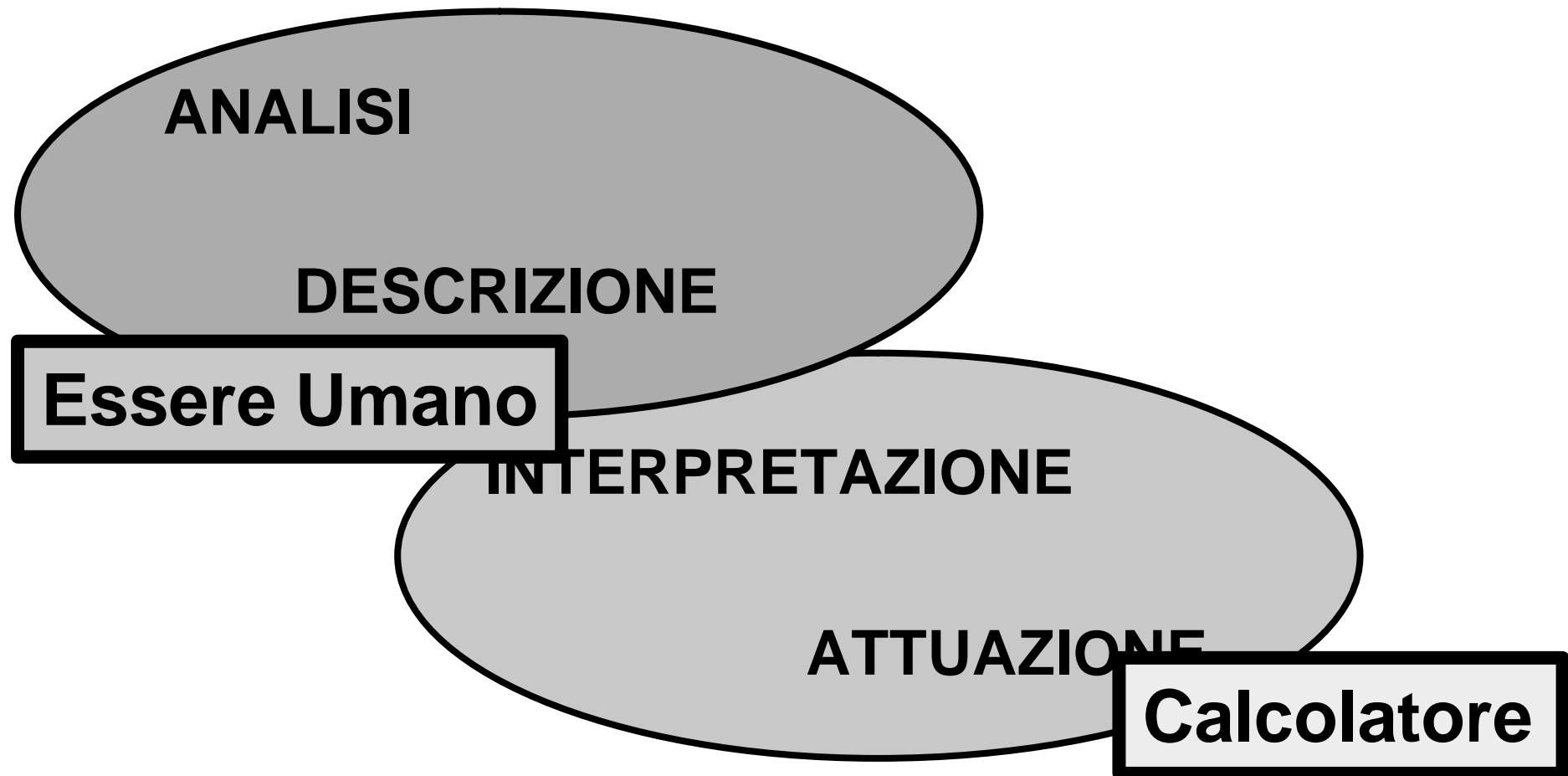
 LUNEDI 27 MARZO NON C'E' LEZIONE

 LUNEDI 3 APRILE NON C'E' LEZIONE

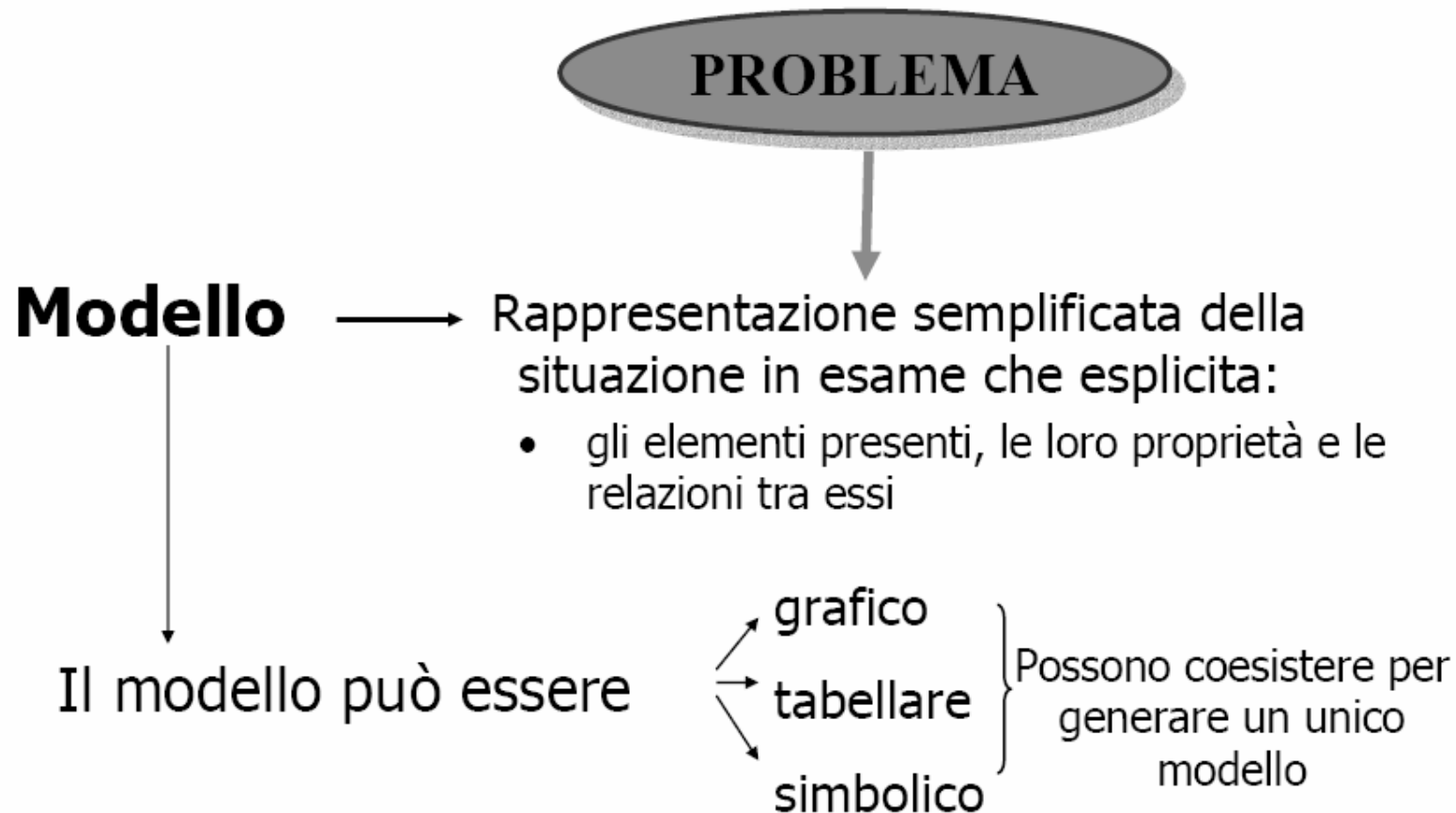
 LUNEDI 24 APRILE NON C'E' PONTE

RIASSUNTO DELLA LEZIONE PRECEDENTE

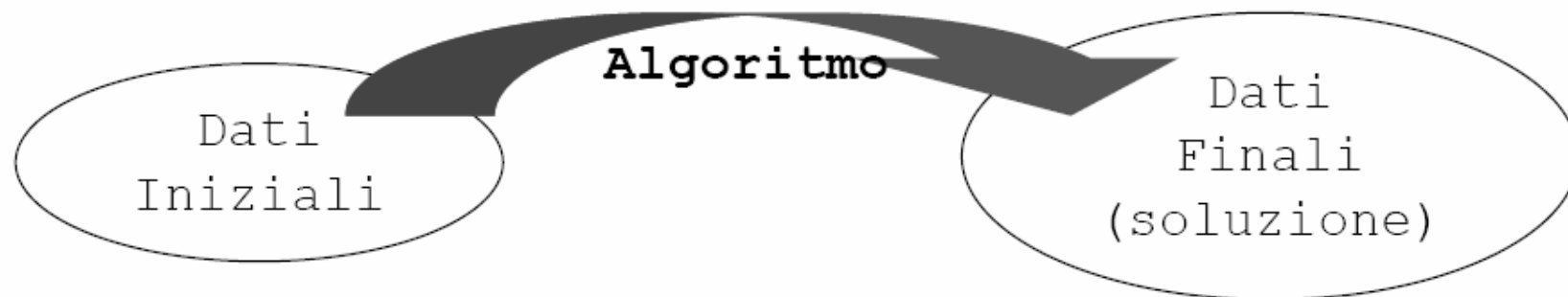
FASI DELLA SOLUZIONE DEL PROBLEMA



Modellazione del problema



Una definizione più precisa...



Si definisce *algoritmo* una *sequenza di azioni* che trasformi i dati iniziali in un numero finito di passi, elementari e non ambigui, per giungere al risultato finale.

Questa sequenza di azioni è valida per un insieme di dati iniziali ben definito e può essere eseguita da un opportuno esecutore.

FASI DELLA SOLUZIONE DEL PROBLEMA

- ✍ I problemi vengono scomposti in sottoproblemi

Problemi elementari



azioni elementari

- ✍ **Azione elementare:** azione che può essere direttamente compiuta dall'*esecutore*

FASI DELLA SOLUZIONE DEL PROBLEMA

- ✍ Un **esecutore** è caratterizzato da:
 - Il **linguaggio** che è in grado di interpretare
 - L'insieme delle **azioni** che è in grado di compiere
 - L'insieme delle **regole** che a ogni costrutto linguistico sintatticamente corretto associano le relative azioni da compiere

LEZIONE 3 - ALGORITMI

CONTENUTO DELLA LEZIONE

 **Concetto di algoritmo**

 **Proprieta' degli algoritmi**

 **Variabili e tipi di dati**

 **Rappresentazione degli algoritmi**

 **Programmi**

Alcuni concetti

- **Algoritmo**
descrizione di come si risolve un problema
- **Programma**
algoritmo scritto in modo che possa essere eseguito da un calcolatore (**linguaggio di programmazione**)
- **Linguaggio macchina**
linguaggio **effettivamente** “compreso” da un calcolatore, caratterizzato da
 - istruzioni primitive semplici (e.g. max 2 operandi)
 - attenzione all'efficienza (costi, complessità, velocità)
 - difficile e noioso da utilizzare per un programmatore
- Compito dell'informatico è **produrre algoritmi** (cioè capire la sequenza di passi che portano alla soluzione di un problema) e **codificarli in programmi** (cioè renderli comprensibili al calcolatore)

Algoritmo (definizione informale)

- Sequenza **finita** di istruzioni,
 - **comprensibili** da un **esecutore** (si può trattare di uno strumento automatico),
 - che descrive come **realizzare un compito** (come risolvere un “problema”).
-
- Alcuni esempi
 - Istruzioni di montaggio di un elettrodomestico
 - Uso di un terminale Bancomat
 - Calcolo del massimo comune divisore di numeri naturali

Proprietà degli algoritmi

➤ **Correttezza**

- L'algoritmo perviene alla soluzione del compito cui è preposto, senza difettare di alcun passo fondamentale

➤ **Efficienza**

- L'algoritmo perviene alla soluzione del problema usando la minima quantità di risorse fisiche
 - tempo di esecuzione, memoria, ...

Proprietà di un algoritmo

➤ **Univocità**

- Non deve esistere alcun grado di libertà da parte del processore nell'esecuzione di ogni azione

➤ **Effettività**

- Le operazioni prescritte dall'algoritmo devono poter essere eseguite in *tempo finito*

➤ **Ingresso**

- Un algoritmo è corretto se porta ad un risultato coerente per ogni possibile scelta dei dati in ingresso

➤ **Uscita**

- L'algoritmo deve fornire uno o più dati in uscita

➤ **Terminazione**

- L'esecuzione di un algoritmo deve terminare in un *numero finito di passi*

Esempio



Data la seguente ricetta, composta da un insieme di azioni elementari, **si verifichi se rispetta le proprietà a cui un algoritmo deve sottostare:**

Biancomangiare

Dosi per 4 persone:

1 litro di latte; 200 gr. di zucchero; 200 gr. di amido; scorza di limone.

Sciogliete l'amido in un pochino di latte. Mescolate lo zucchero al latte rimanente ed unite un po' di scorza di limone grattugiata. Unite l'amido sciolto e mescolate il tutto. Fate cuocere a fuoco basso, rimescolando continuamente finché la crema non si addensa. Versate nelle coppette e lasciate raffreddare in frigorifero per tre ore.

Altri esempi di algoritmo

 In un editor di testi:

- Algoritmo per il controllo ortografico
- Algoritmo per il conteggio parole
- Algoritmo per sostituire una parola

Componenti dell'Algoritmo

✍ Algoritmo = istruzioni + strutture dati

- Variabili
- Tipi di istruzioni

VARIABILI

- ✍ In genere non è interessante risolvere la singola istanza di un problema ma occorre generalizzare
 - Es: $A=B+C$
- ✍ Le istruzioni fanno riferimento a variabili il cui valore può cambiare a seconda della situazione

VARIABILI

- ✍ Le variabili sono “contenitori” per i dati
- ✍ Ogni variabile ha un **nome** che la identifica univocamente
- ✍ La variabile assume un **valore** che in ogni istante corrisponde al dato contenuto nella variabile

VARIABILI

✍ In ogni algoritmo i nomi delle variabili sono usati in operazioni di assegnamento

Es: $r \leftarrow 5$

$pi \leftarrow 3.14$

e compaiono all'interno delle espressioni

Es: $c \leftarrow 2 \leftarrow pi \leftarrow r$

VARIABILI, 2

✍ Copiare il valore di x in y e il valore di y in x
algoritmo errato:

$X \leftarrow 2$ $y \leftarrow 3$

$x \leftarrow y$

$X \leftarrow 3$ $y \leftarrow 3$

$y \leftarrow x$

$X \leftarrow 3$ $y \leftarrow 3$

algoritmo corretto:

$temp \leftarrow y$

$Temp \leftarrow 3$

$y \leftarrow x$

$Y \leftarrow 2$

$x \leftarrow temp$

I dati

- Ogni variabile è caratterizzata dal suo *tipo*.
 - Tipi predefiniti: numeri, caratteri, booleani, ...
 - Altri tipi: stringhe, date, ...
- Variabili strutturate:
 - Vettori (o array)
 - Record

ARRAY (o VETTORI) – Tipi strutturati

✍ Es: la temperatura della settimana

– Double T[7]

✍ T[0]=18; T[1]=20; ...T[6]=22

e' importante definire la dimensione

MATRICI – Tipi strutturati

- ✍ Es: Numero alunni classe
 - Integer Alunni[5][3] (ho 2 dimensioni)
 - ✍ La prima A ha 25 alunni
 - Alunni[1][1]=25

RECORD – Tipi strutturati

✍ Es: *struct* PERSONA

- *String* Nome
- *String* Cognome
- *String* Data di nascita
- Es: PERSONA Perso

✍ Perso.Nome="Mario"

✍ Perso.Cognome="Rossi"

✍ Perso.Data di nascita= 10/10/1970

DATI e VARIABILI

- ✍ Nei programmi i dati sono modellati attraverso le VARIABILI
- ✍ Dichiarazione
- ✍ Assegnamento
 - Inizializzazione
 - aggiornamento

Esempio di algoritmo:

- ✍ Determinare il maggiore di due numeri interi x e y

ISTRUZIONI

✍ **Strutture condizionali:** quando è presente una condizione logica che può essere **vera** o **falsa** (condizioni a *valori booleani*)

– Es:

```
if (a>b) then max ? a  
else max ? b
```

ISTRUZIONI

✍ **Sottoprogrammi:** quando la scomposizione in sottoproblemi genera un numero di passi elevato la soluzione viene rimandata ad un altro problema *non elementare terminale*. Nei linguaggi di programmazione si parla di sottoprogrammi

✍ Es:

- *leggi A*
- *leggi B*
- *calcola MAX(A,B)*
- *Stampa MAX(A,B)*

ISTRUZIONI

- ✍ **Strutture iterative o cicli (*loop*):** quando un blocco di istruzioni (o sottoprogramma) viene ripetuto più volte.
- ✍ Sono istruzioni del tipo: “*finché* condizione *ripeti* azione”

Es:

P1: leggi A

P2: I=0

P3: finché I<10 cicla fino a P6

P4: stampa A

P5: I=I+1

P6: fine ciclo

P7: stop

ISTRUZIONI

Ingresso/Uscita

- Stampa A

Aritmetico/logiche

- Area ? $\pi \cdot r^2$

Istruzioni di controllo

- Se Val < Min allora Min ? Val

Strutture Iterative o cicli

- Fintantochè $I < N$

Rappresentazione degli Algoritmi

- 1. Linguaggio naturale
- 2. Diagramma a blocchi
- 3. Pseudo codice
- 4. Linguaggio di programmazione

Rappresentare gli algoritmi

Linguaggio naturale

- Sollevare il ricevitore
- Attendere il segnale di linea libera
- Comporre il numero
- ...

Pseudo codice

Input A,B

Tot \leftarrow 0

While A!=0 **Do**

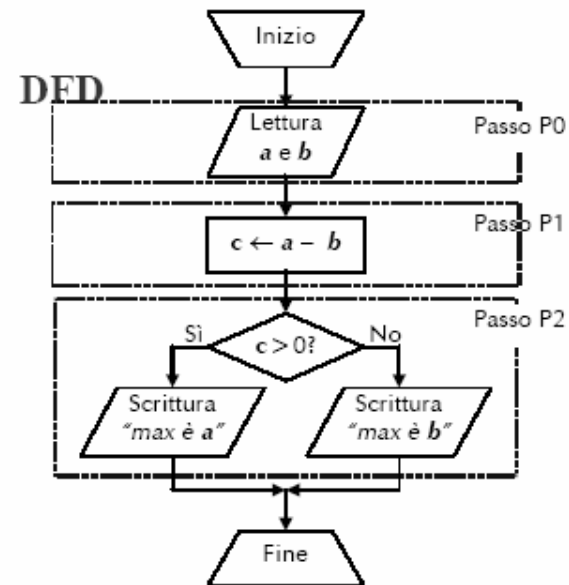
 Tot \leftarrow Tot + B

 A \leftarrow A - 1

Output Tot

04/04/02

DFD



Ling programmazione

```
#include <stdio.h>
```

```
Int main (void){
```

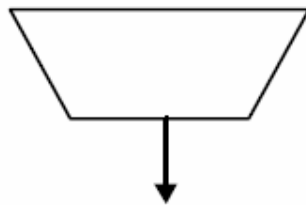
```
    puts("ciao mondo!");
```

```
    return Exit_success;
```

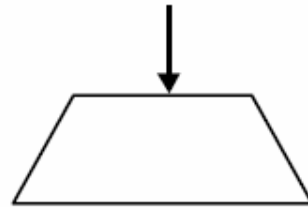
```
}
```

Introduzione ai sistemi informatici

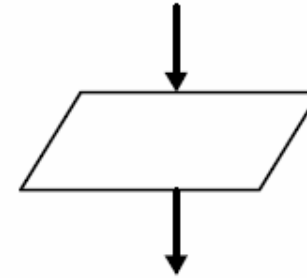
Diagrammi di flusso



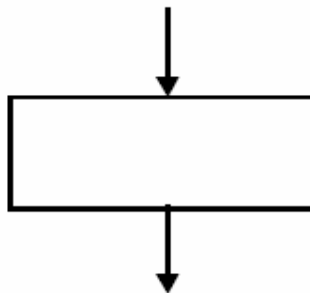
Inizio



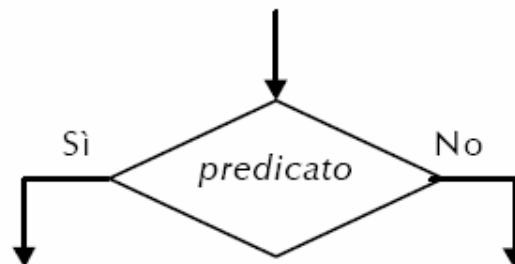
Fine



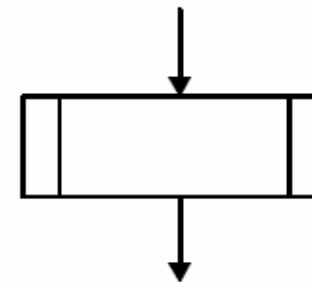
Operazioni
di ingresso/uscita



Elaborazione



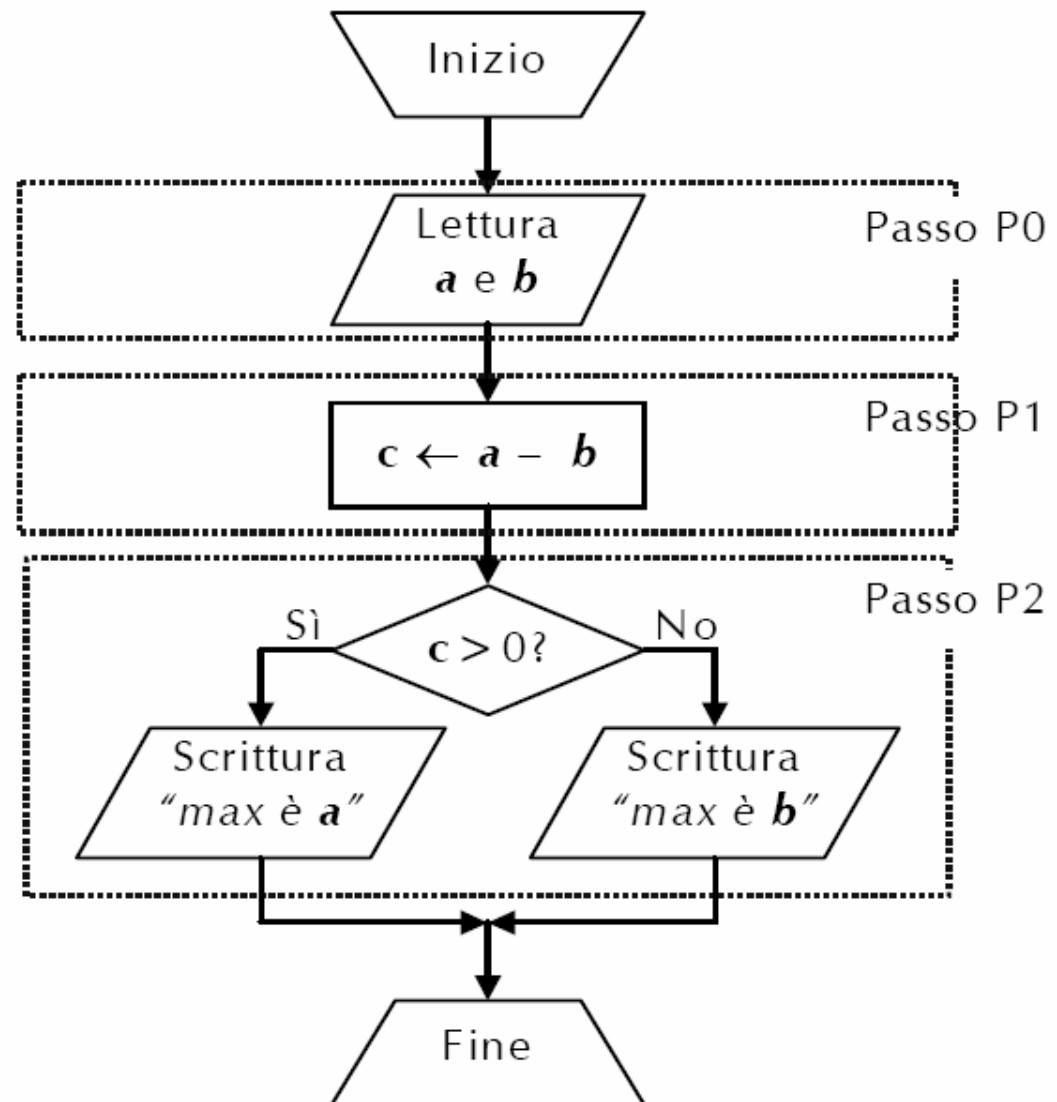
Selezione a due vie



Sottoprogramma

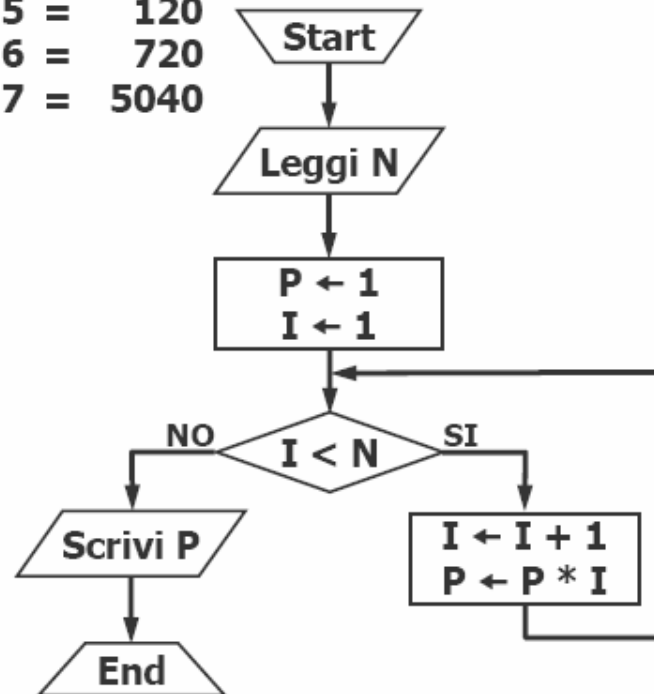
Esempio

- **Esempio:** dati in ingresso due numeri A e B, si calcoli e stampi il maggiore.

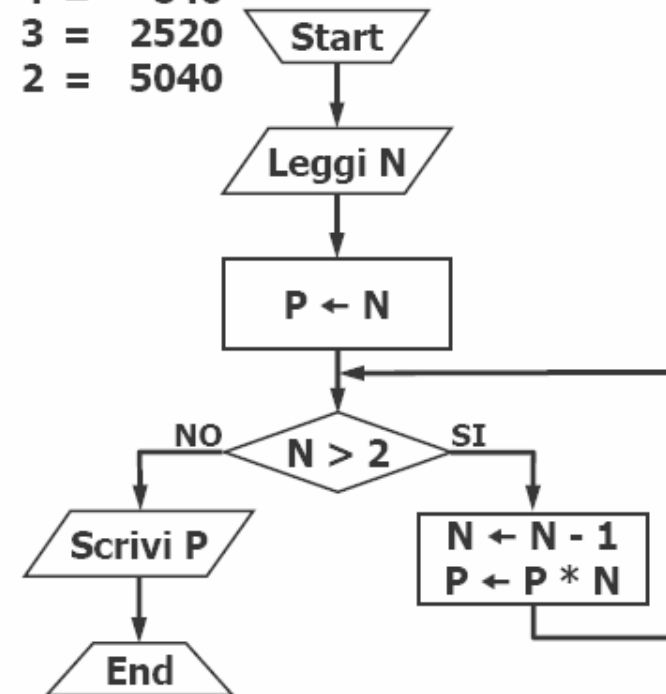


Diverse alternative (e.g. 7!)

1	*	2	=	2
2	*	3	=	6
6	*	4	=	24
24	*	5	=	120
120	*	6	=	720
720	*	7	=	5040

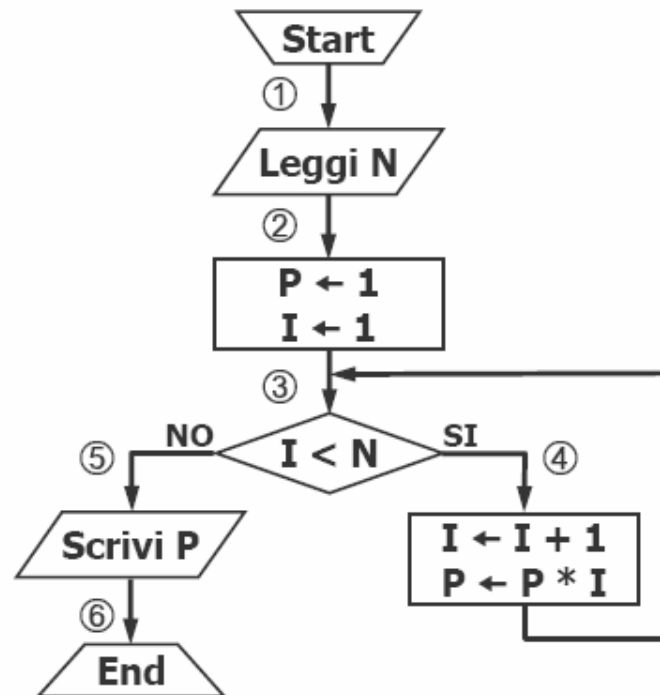


7	*	6	=	42
42	*	5	=	210
210	*	4	=	840
840	*	3	=	2520
2520	*	2	=	5040



"Tracciato" dell'esecuzione

Ipotizziamo di calcolare 4!





T	pos	N	I	P	note
t ₁	①	??	??	??	
t ₂	②	4	??	??	
t ₃	③	4	1	1	I < N
t ₄	④	4	1	1	
t ₅	③	4	2	2	I < N
t ₆	④	4	2	2	
t ₇	③	4	3	6	I < N
t ₈	④	4	3	6	
t ₉	③	4	4	24	I = N
t ₁₀	⑤	4	4	24	
t ₁₁	⑥	4	4	24	

**Attenzione: il valore iniziale non è 0!!
Può non esserci oppure esserci ma non essere noto.**

PROGRAMMI

- ✍ **Un algoritmo non è ancora utilizzabile dall'esecutore automatico**
- ✍ **Sono stati definiti linguaggi di programmazione di “*alto livello*”**

PROGRAMMA definizione

-  ***Un programma è una successione di istruzioni, che mediante un opportuno programma di traduzione possono essere trasformate in istruzioni in linguaggio macchina (“basso livello”) direttamente eseguibili dal calcolatore***
-  **Es di linguaggi di programmazione:**
 - C, C++, Java, Pascal, Basic, etc.**

RIFERIMENTI AL LIBRO

Lezione 3 (algoritmi)

- Paragrafi 3.1.1; 3.1.2
- Paragrafi 3.3; 3.3.1; 3.3.2