PROBLEMI
E
ALGORITMI

prof.ssa VESPIA CATERINA

LICEO CLASSICO AGLI ANGELI

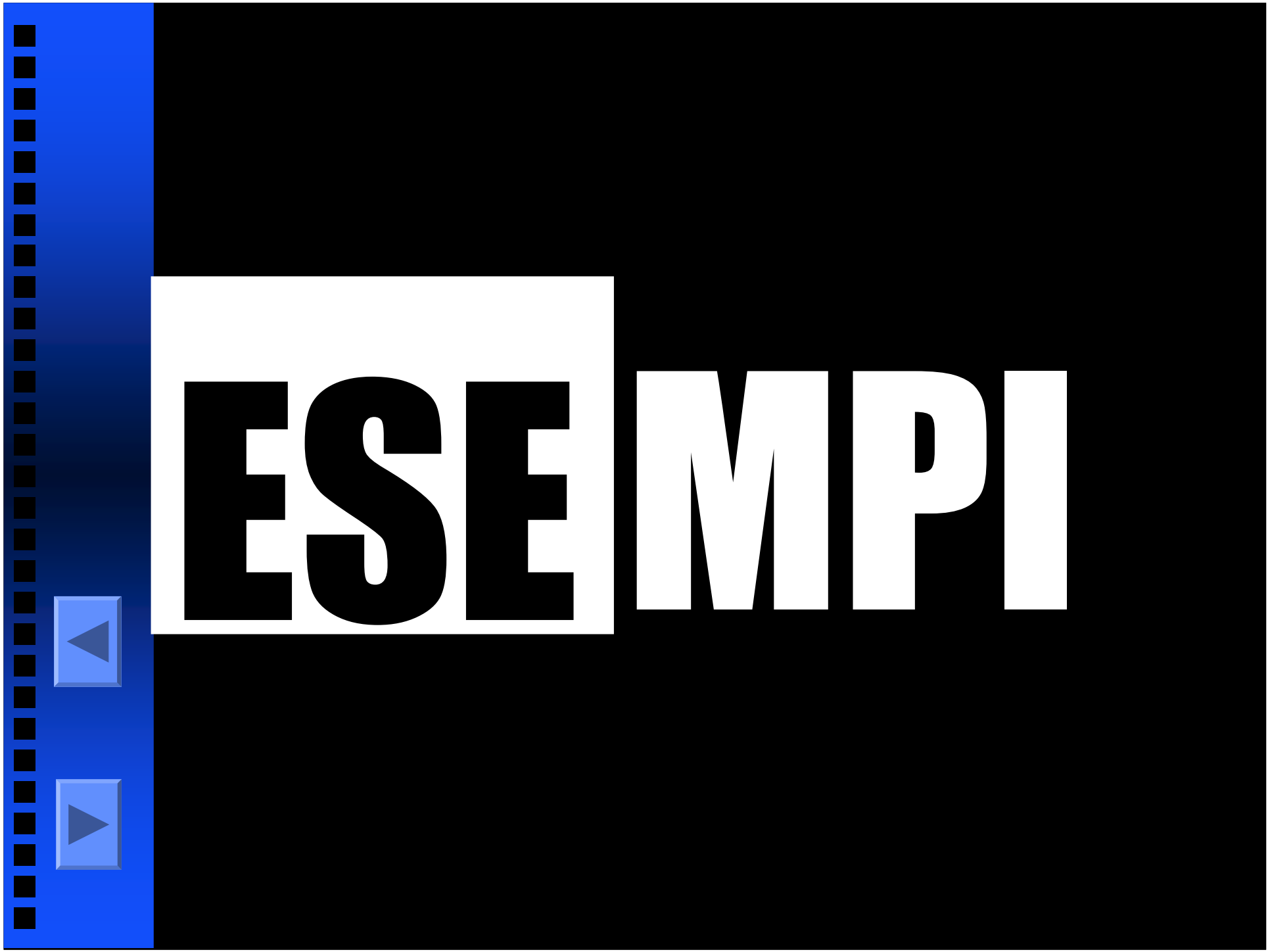
CONTENUTI

- Problemi.
- Concetto di algoritmo.
- Caratteristiche di un algoritmo.
- Descrizione di algoritmi - Diagrammi di flusso.
- Rappresentazione di algoritmi - Linguaggio di progetto.
- Costruzione strutturata di algoritmi : schemi fondamentali.

Problema

=

Situazione di parziale ignoranza


ESE
MPI

1° Problema(di tipo deterministico):

Il sig. Rossi vuole acquistare una casa.

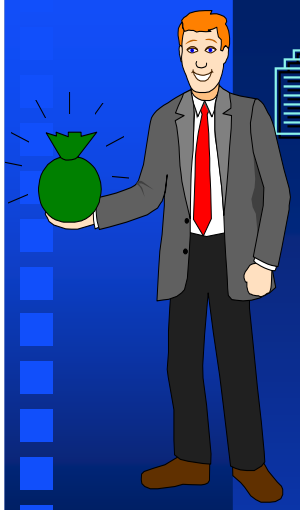
Obiettivo: Stabilire se il sig. Rossi è in grado di acquistare l'appartamento.

Dati:
- l'appartamento è in vendita;
- il signor Rossi dispone di una data somma S e non intende ricorrere a prestiti o a rateazioni;
- il sig. Rossi dispone di una pianta dell'appartamento;

- il Sig. Rossi conosce il prezzo per metro quadrato.

Strategia risolutiva:

- ricavare dalla pianta l'area della superficie dell'appartamento;
- calcolare il costo dell'appartamento moltiplicando l'area per il costo al metro quadro;
- confrontare il costo C con la [illegible]a S: se C è minore o uguale a S [illegible]to può essere effettuato.

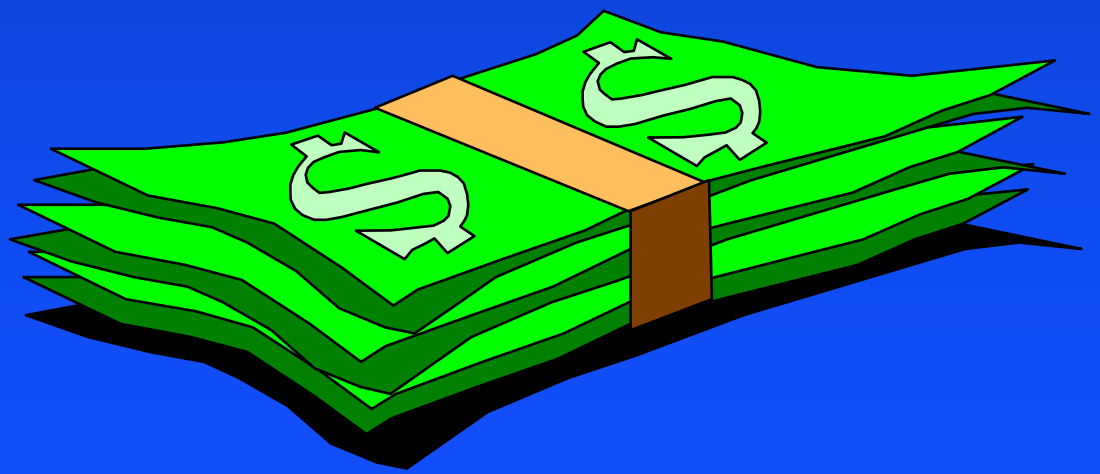


2°Problema (di tipo non deterministico): *Investire una certa somma di denaro nel miglior modo possibile.*

- Obiettivo: Realizzare l'investimento più conveniente. Tale obiettivo potrà essere realizzato con un grado di probabilità più o meno elevato.
- Dati: Informazioni di carattere statistico, come, ad esempio, il rendimento medio negli ultimi sei mesi dei titoli o delle azioni nelle quali si intende investire.

Strategia risolutiva:

- deposito bancario;
- buoni del tesoro;
- obbligazioni;
- azioni;
- fondi comuni.



PERCIÒ

- Per risolvere un problema occorrono determinate informazioni.
- I dati iniziali determinano la strategia oppure la strategia determina i dati iniziali.
- Nei problemi di tipo deterministico, una vasta gamma di informazioni iniziali permette di avere più soluzioni alternative per poi scegliere quella ottimale.

Concetto di **ALGORITMO**

La parola “algoritmo” è legata al nome del celebre matematico arabo *Mohammed ibn Musà al Khuwarizmi.*

L’algoritmo è una:

Strategia risolutiva

Sequenza ordinata di istruzioni
che definiscono
operazioni (o azioni) per
raggiungere l'obiettivo

Esempi di algoritmi

- regole per eseguire le quattro operazioni;
- regola per determinare il M.C.D. o il m.c.m. di due numeri naturali;
- regole per il calcolo frazionario;
- regole di un gioco di carte;
- istruzioni per utilizzare un telefono pubblico;
- istruzioni per una ricetta culinaria, ecc.

Caratteristiche di un algoritmo

Un algoritmo è effettivamente eseguibile da un esecutore umano oppure da dispositivi meccanici o elettronici (automi) se:

- ha carattere di finitezza;
- non è ambiguo;
- ha carattere di generalità.

Finitezza

L’algoritmo deve essere composto da un numero finito di azioni (o passi) e il numero di volte che questi passi vengono eseguiti deve essere finito.

Non ambiguità

Le istruzioni devono essere formulate in modo da essere interpretate univocamente da esecutori diversi.

Generalità

L’algoritmo deve risolvere una classe di proble_ mi e non un solo problema.

L’algoritmo scritto in modo comprensibile allo esecutore diventa un PROGRAMMA e il lin_ guaggio usato è detto LINGUAGGIO DI PROGRAMMAZIONE.

Per esecutori umani è il linguaggio naturale op_ portunamente modificato (LINGUAGGIO DI PROGETTO).

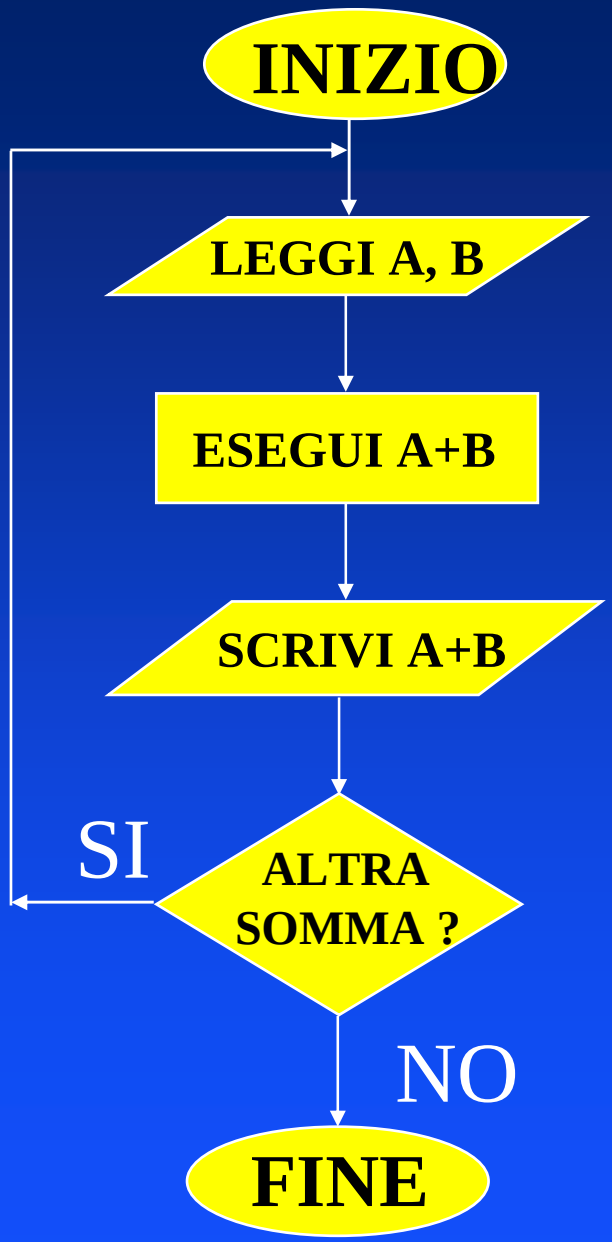
Per esecutori non umani (COMPUTER) è il LINGUAGGIO MACCHINA.

Descrizione di algoritmi

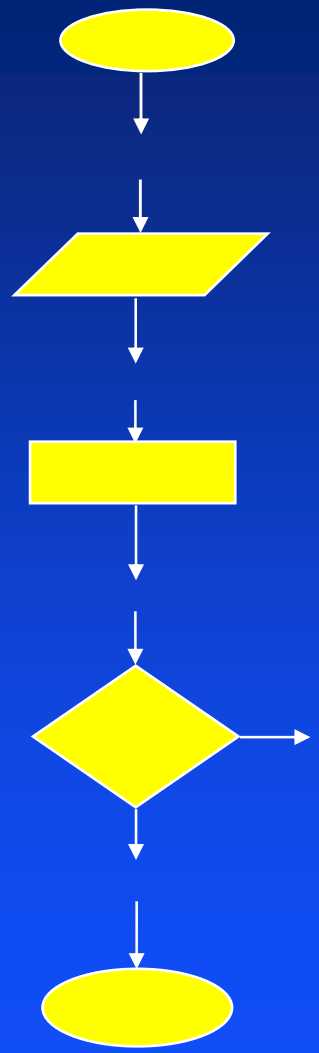
Un algoritmo può essere descritto mediante un diagramma a blocchi o FLOW-CHART che rappresenta graficamente le istruzioni da eseguire e la loro attivazione.
Ad ogni tipo di istruzione è associato un parti_ colare blocco; i blocchi sono collegati da linee munite di frecce che definiscono il flusso di attivazione.

ESEMPIO

Algoritmo per la somma di due numeri.



S I M B O L I



Indica l’ inizio (Start, via, inizio)

Racchiude istruzioni di lettura dei dati e istruzioni di scrittura dei dati (input ,output)

Racchiude istruzioni di assegnazione dei dati ed operazioni matematiche

Indica la fase di controllo e la strada da percorrere al verificarsi o no della condizione scritta internamente al rombo

Indica la fine (End, stop, fine)

Rappresentazione degli algoritmi

LINGUAGGIO DI PROGETTO

(PSEUDOLINGUAGGIO)

Tecnica di rappresentazione degli algoritmi che utilizza vocaboli ed espressioni molto vicine al linguaggio naturale e una serie di strutture verbali con le quali si descrivono le strutture fondamentali della programmazione e, quindi, qualsiasi algoritmo realizzato con esse.

Sarà espresso in lingua italiana e conterrà un numero finito di caratteri, simboli e parole “riservate”.

Alfabeto

- caratteri alfabetici, maiuscoli o minuscoli;

  A, B, … , X, Y, Z; **a, b, c, … , x, y, z;**

- cifre numeriche **0, 1, 2, 3, …….,9** ;
- simboli speciali (**= > < ‘ / ? * - + ( ) , ; :=** …)

Parole riservate

Sono sequenze di lettere aventi un preciso significato e non utilizzabili con significato diverso da quello previsto (**inizia, se, allora, altrimenti, mentre, esegui, ripeti, finché**,…)

DATI

Insieme di oggetti con i quali l'algoritmo opera.

I tipi di dati consentiti nel L.P. sono:

- **interi** (numeri interi con segno);
- **reali** (numeri con virgola e con segno);
- **alfanumerici** (dati che contengono caratteri alfanumerici).

I dati possono essere :

- **costanti** (sono rappresentati da un valore determinato)

variabili (sono rappresentati da simboli a cui possono essere attribuiti valori diversi scelti in un determinato insieme.)

Se i valori che le variabili possono assumere appartengono agli insiemi N, Q e R ➔ variabili numeriche

Se i valori che le variabili possono assumere sono stringhe di numeri e lettere ➔ variabili alfanumeriche

Se i valori che le variabili possono assumere sono due soli *vero* o *falso* ➔ variabili logiche

Le variabili si rappresentano con stringhe formate da lettere e numeri dette nomi delle variabili (o identificatori).

Esempio: SOMMA, TOT, D3

Le costanti numeriche, se sono reali, si scrivono con il punto decimale.

Esempio: 5, 5.7, 0

Le costanti alfanumeriche si scrivono tra apici.

Esempio: “ANTONIO”, “M”

Espressioni aritmetiche

Con i dati possono essere formate espressioni di vario genere, utilizzando diversi tipi di operatori:

- **operatori aritmetici**

( )	parentesi tonde
-	meno unario (segno meno)
↑	elevamento a potenza
div	divisione tra interi con risultato intero
***, /**	moltiplicazione, divisione
+, -	addizione, sottrazione

Gli operatori aritmetici ammettono come operandi dati numerici, interi o reali, e forniscono un risultato numerico.

Le espressioni entro le parentesi tonde vengono eseguite per prime, e le altre operazioni vengono eseguite nell'ordine descritto nella tabella.

Le operazioni che hanno pari priorità vengono eseguite nell'ordine in cui si incontrano, leggendo l'espressione da sinistra verso destra.

Esempi

5/2 = 2.5 ; 5 div 2 = 2 ; 3+2/2 = 3+(2/2) = 4 ;
(3+2)/2 = 5/2 = 2.5 ; -3 ↑ 2 = 9 ; -(3 ↑ 2) = -9

operatori relazionali (sono gli operatori che consentono di eseguire confronti fra dati)

=	uguale
<	minore
>	maggiore
< =	minore o uguale
> =	maggiore o uguale
< > , > <	diverso

Gli operatori relazionali si possono utilizzare per confrontare sia dati numerici sia dati alfanumerici.
Il risultato di un'espressione relazionale può essere VERO o FALSO, e condiziona in genere la prosecuzione della procedura.

Esempi

3>5	FALSO
3<5	VERO
3=3	VERO
A > B	FALSO
A < B	VERO
A=A	VERO
A=AA	FALSO

operatori logici (sono operatori che ammettono come operandi i risultati di espressioni relazionali e danno come risultato un valore VERO o FALSO).

AND e

OR o

NOT non

Esempi

☆ **(A>B) AND (B>C)**

•dà un risultato VERO se A>B e B>C sono entrambe VERE (7>5 e 5>3 il risultato dell'espressione è VERO)

•altrimenti dà un risultato FALSO

(7>5 e 5>6 il risultato dell'espressione è FALSO)

(A>B) OR (B>C)

•dà un risultato VERO se almeno una delle due espressioni che si trovano ai lati di OR è VERA (7>5 o 5>6 il risultato sarà VERO)

•altrimenti dà un risultato FALSO (3>5 o 5>7 il risultato dell'espressione è FALSO)

NOT (A>B)

•dà un risultato FALSO se l’espressione è VERA (not(7>5) il risultato è falso)

•dà un risultato VERO se l’espressione è FALSA (not(5>7) il risultato è VERO)

ISTRUZIONI FONDAMENTALI

Istruzione di lettura

LEGGI (nome variabile**)**

Esempi: leggi (A) , leggi(A,B,C) , leggi(raggio)

Istruzione di scrittura

SCRIVI (nome variabile**)**

Esempi: scrivi(A) , scrivi(area) , scrivi('Il perimetro è')

Istruzione di assegnazione

nome variabile ← valore dell'insieme di definizione

Esempi: A ← 5 , B ← A , C ← A+B , C ← C+1

Costruzione strutturata di algoritmi

La costruzione degli algoritmi deve obbedire a tecniche e regole precise e rigorose

PROGRAMMAZIONE STRUTTURATA.

STRUTTURE DI CONTROLLO FONDAMENTALE

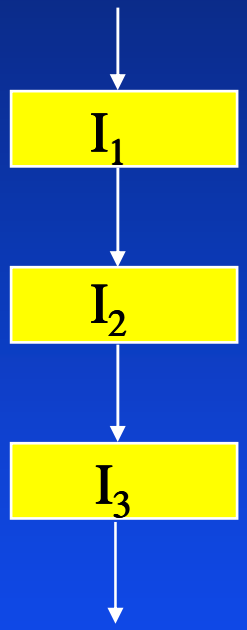
Schema di sequenza

Schema di selezione

Schema di iterazione

Schemi fondamentali

- SCHEMA DI SEQUENZA



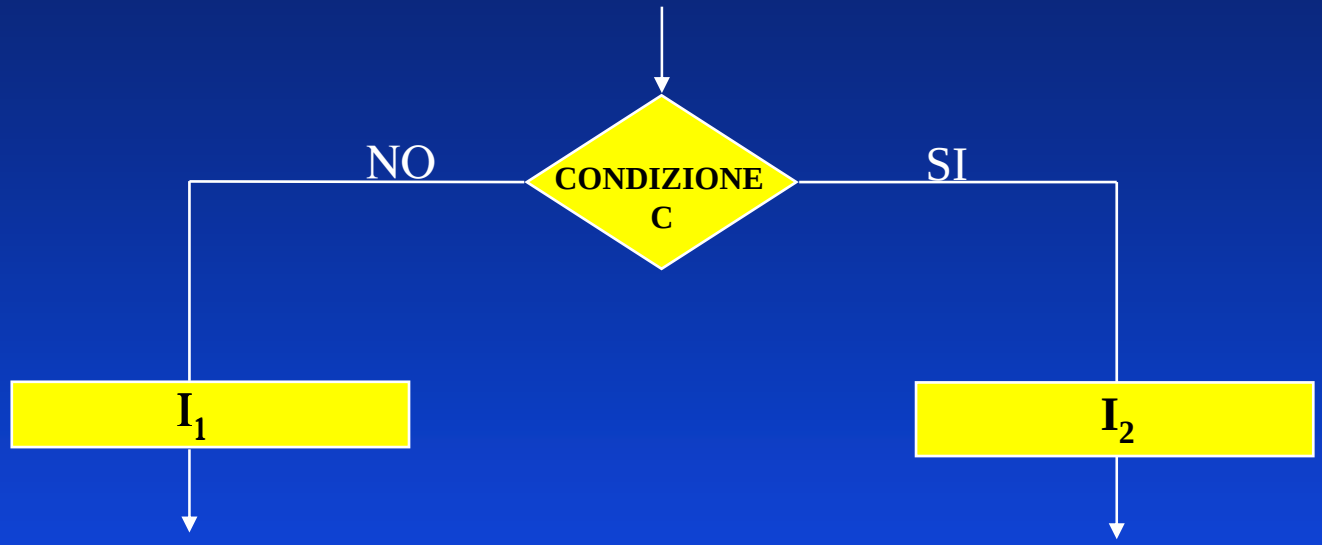
Linguaggio di progetto

inizio <I_1>; <I_2>; <I_3>; … **fine**

Le istruzioni I_1, I_2, I_3,... vengono eseguite una per volta nell'ordine in cui sono scritte.

SCHEMA DI SELEZIONE

Primo caso



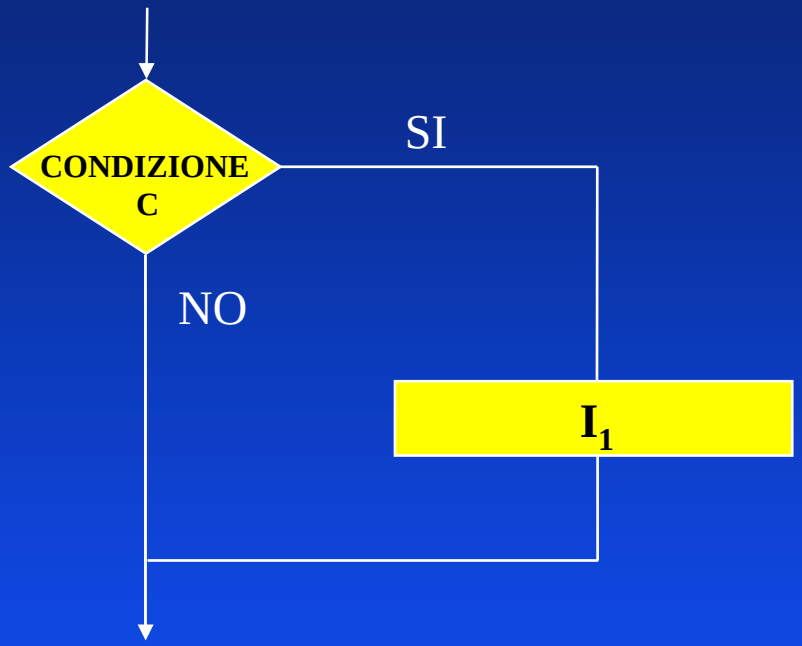
Linguaggio di progetto

SE condizione **C**

ALLORA istruzione I_1

ALTRIMENTI istruzione I_2

Secondo caso



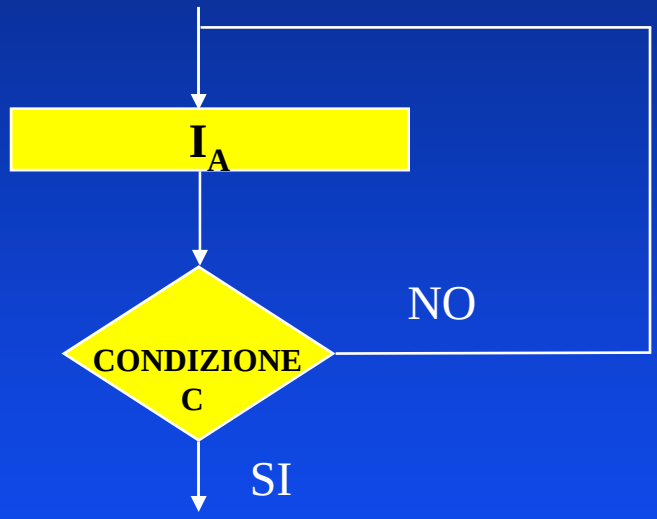
Linguaggio di progetto

SE condizione C

ALLORA istruzione I_1

SCHEMA DI ITERAZIONE (CICLO)

- Primo caso

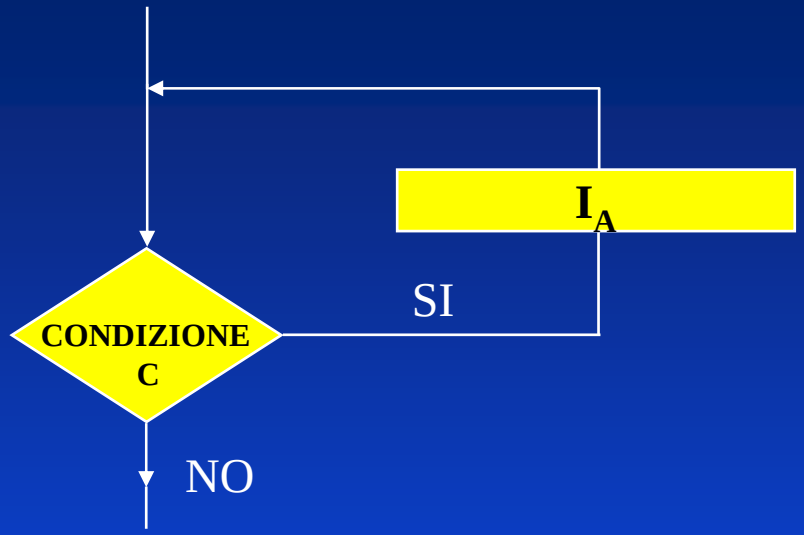


Linguaggio di progetto

RIPETI istruzione A FINCHE' condizione C

N.B. L'istruzione A viene eseguita almeno una volta

Secondo caso



Linguaggio di progetto

MENTRE condizione C ESEGUI istruzione A

N.B. Poichè il test sulla condizione viene eseguito prima delle istruzioni, esse saranno eseguite solo se la condizione risulterà vera.

The

M-film