Il pensiero computazionale

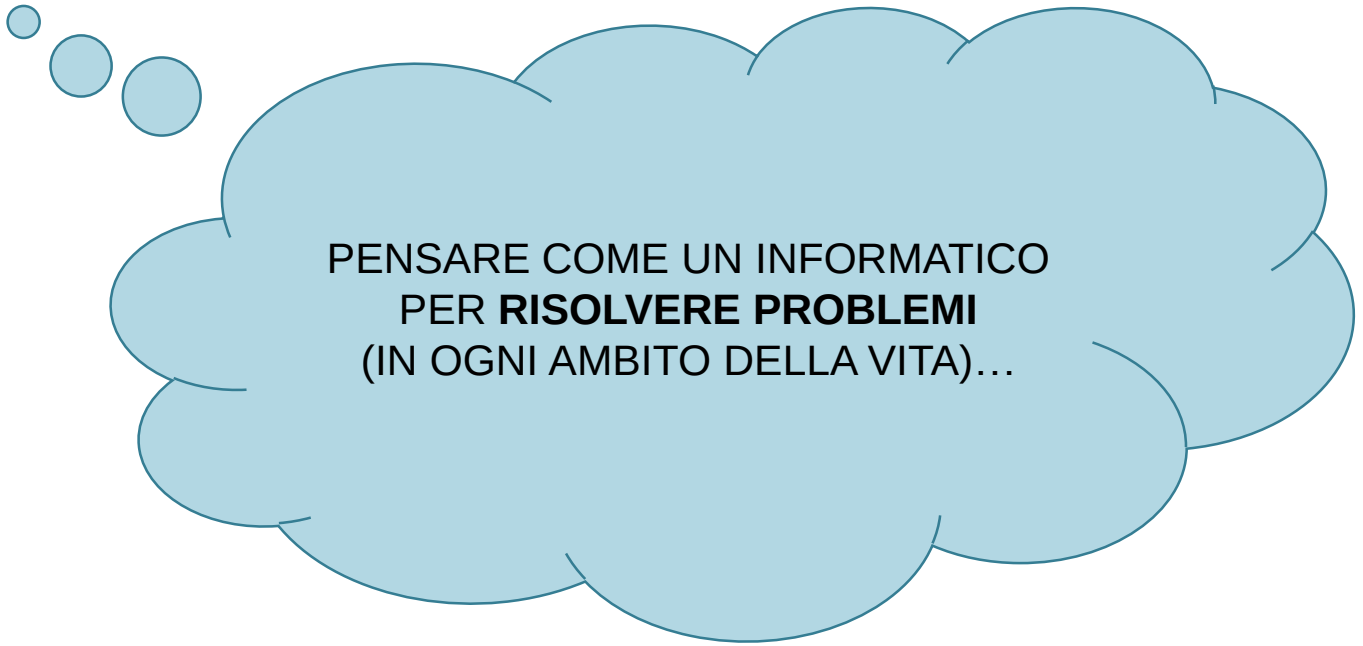
LINKYOU
WEBINAR

Prof. Piero Gallo

28 settembre 2017

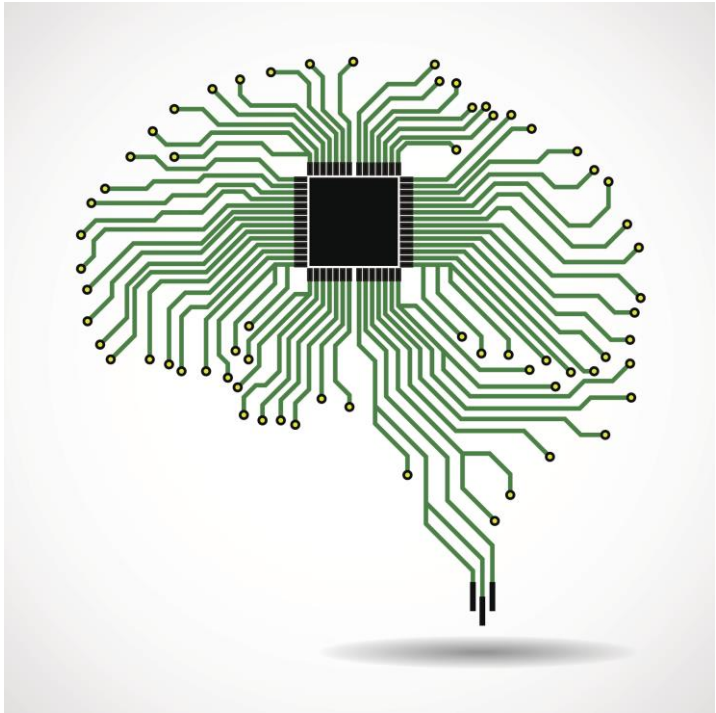
MONDADORI
EDUCATION

«Sono le persone che nessuno immagina
che possano fare certe cose, quelle che fanno
cose che nessuno può immaginare»
Dal film *The Imitation Game* (A. Turing)

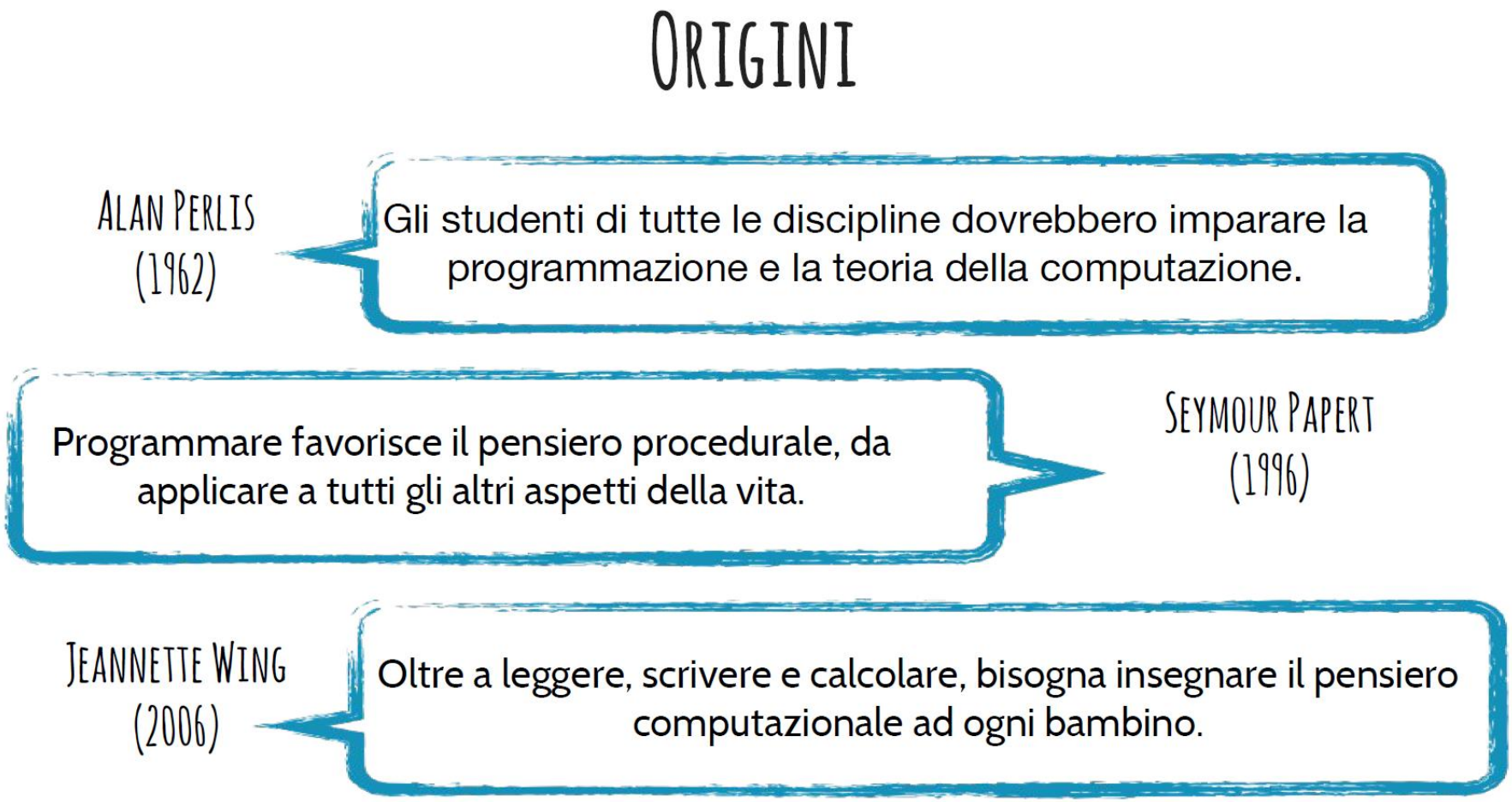


«Sono le persone che nessuno immagina
che possano fare certe cose, quelle che fanno
cose che nessuno può immaginare»
Dal film *The Imitation Game* (A. Turing)

… NON EQUIVALE A PENSARE
COME UN COMPUTER



Riflessioni



Riflessioni

- Viviamo in un mondo scientifico/tecnologico, ma l'**interesse per la scienza/tecnologia** non è diffuso come potrebbe e dovrebbe.
- Questo è un problema per la scienza e per la tecnologia… ma è un problema anche per il nostro **futuro**.

Riflessioni

- Perché succede?
- Ci sono tante cause: una tra le più importanti è la **mancanza di stimoli** adeguati nell'età dell'apprendimento.

Riflessioni

- L’**attitudine verso la scienza** è un atteggiamento che il bambino acquisisce (o non acquisisce) presto, di solito tra gli 8 e gli 11 anni, e tipicamente non lo cambia più.
- Se il bambino non prova interesse per la scienza, sentirà anche che discipline come la fisica, la matematica e l’informatica, sono **troppo ardue**, e le troverà difficili da imparare in futuro.

Quindi?

- Imparare il pensiero computazionale.
- Imparare a programmare.

Il pensiero computazionale
per ogni disciplina e per chiunque.

Il Piano Nazionale Scuola Digitale

Il Piano, presentato il 27 ottobre 2015
dal ministro Giannini, riprende ed esplicita quanto
già annunciato nella legge 107 Buona Scuola.

L'azione #17 del PNSD **introduce di fatto il pensiero computazionale nella scuola primaria** con una duplice valenza:

- avviare alla comprensione e alla conoscenza delle **potenzialità della rete** e della tecnologia fin da giovani;
- delineare gli **indirizzi strategici per l'innovazione digitale** facendo riferimento alle otto competenze chiave tra cui la competenza digitale.

Che cos’è il pensiero computazionale?

Definizione formulata da Jeannette Wing:

“È il processo mentale che sta alla base della formulazione dei problemi e delle loro soluzioni così che le soluzioni siano rappresentate in una forma che può essere implementata in maniera efficace da un elaboratore di informazioni, sia esso umano o artificiale”.

In altre parole è lo **sforzo** che un individuo deve mettere in atto per fornire a un altro individuo o macchina **tutte e sole le “istruzioni” necessarie** affinché questi, eseguendole, sia in grado di portare a termine il compito dato.

Il pensiero computazionale

- È una competenza imprescindibile perché costituisce un metodo di **ragionamento e di risoluzione dei problemi** che si applica con successo anche al di fuori della disciplina informatica intesa come tecnologia.
- **Va considerata come quarta abilità di base per ogni individuo, a fianco di leggere, scrivere e calcolare.**

In parole semplici…

Quando affrontiamo un problema o abbiamo un'idea, può accadere che intuiamo la soluzione, ma non siamo in grado di formularla in modo operativo e metterla in pratica.

Il pensiero computazionale è la capacità di descrivere un procedimento costruttivo che porti a una soluzione creativa, efficace e non ambigua.

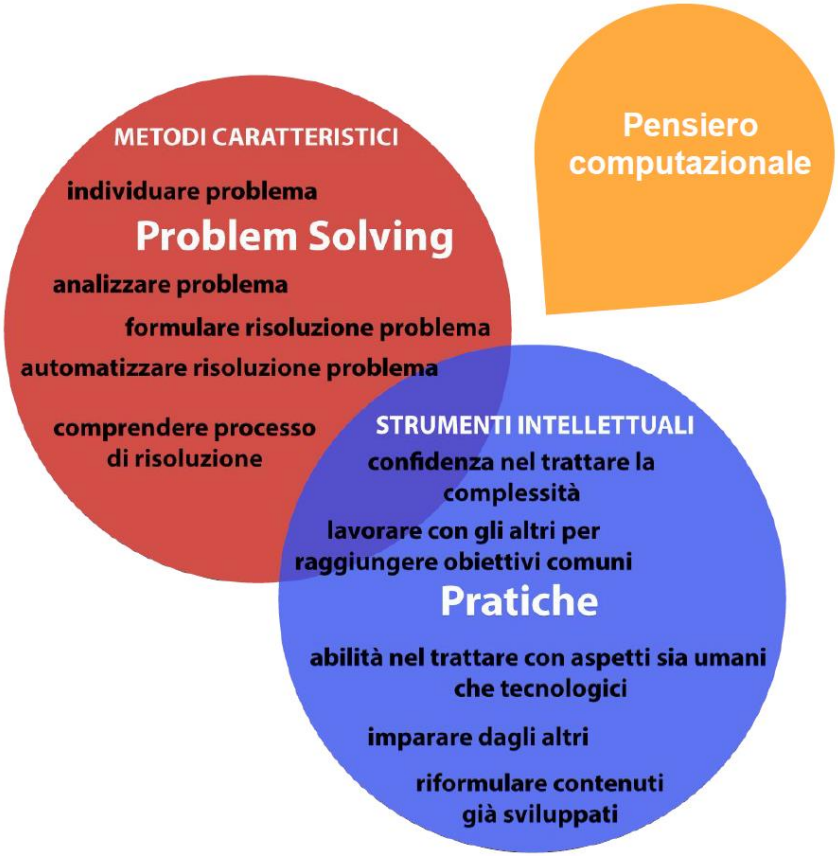
Qual è il legame tra computer, informatica e pensiero computazionale?

Come l'invenzione della stampa ha facilitato la diffusione dell'alfabetizzazione, oggi la programmazione e i computer facilitano l'acquisizione e la diffusione del pensiero computazionale.

Il pensiero computazionale sfrutta concetti e strumenti dell'informatica per trovare soluzioni innovative e creative ai problemi di ogni giorno.

Il pensiero computazionale e la computer science

La computer science
non si occupa di insegnare,
sviluppare e migliorare
i linguaggi di programmazione,
ma offre contributi importanti
per imparare a capire
come pensiamo,
come organizziamo
il nostro sapere,
come impariamo cose nuove,
come condividiamo
quello che sappiamo.



Che cosa sto imparando mentre studio, approfondisco e sviluppo questi temi?

Comprendere un problema in modo diretto

Imparare dagli altri

Esistono più soluzioni a un problema

Comprendere l'importanza dell'errore

Avere la libertà di sbagliare

Imparare per tentativi e strategie

Condividere ciò che si è imparato

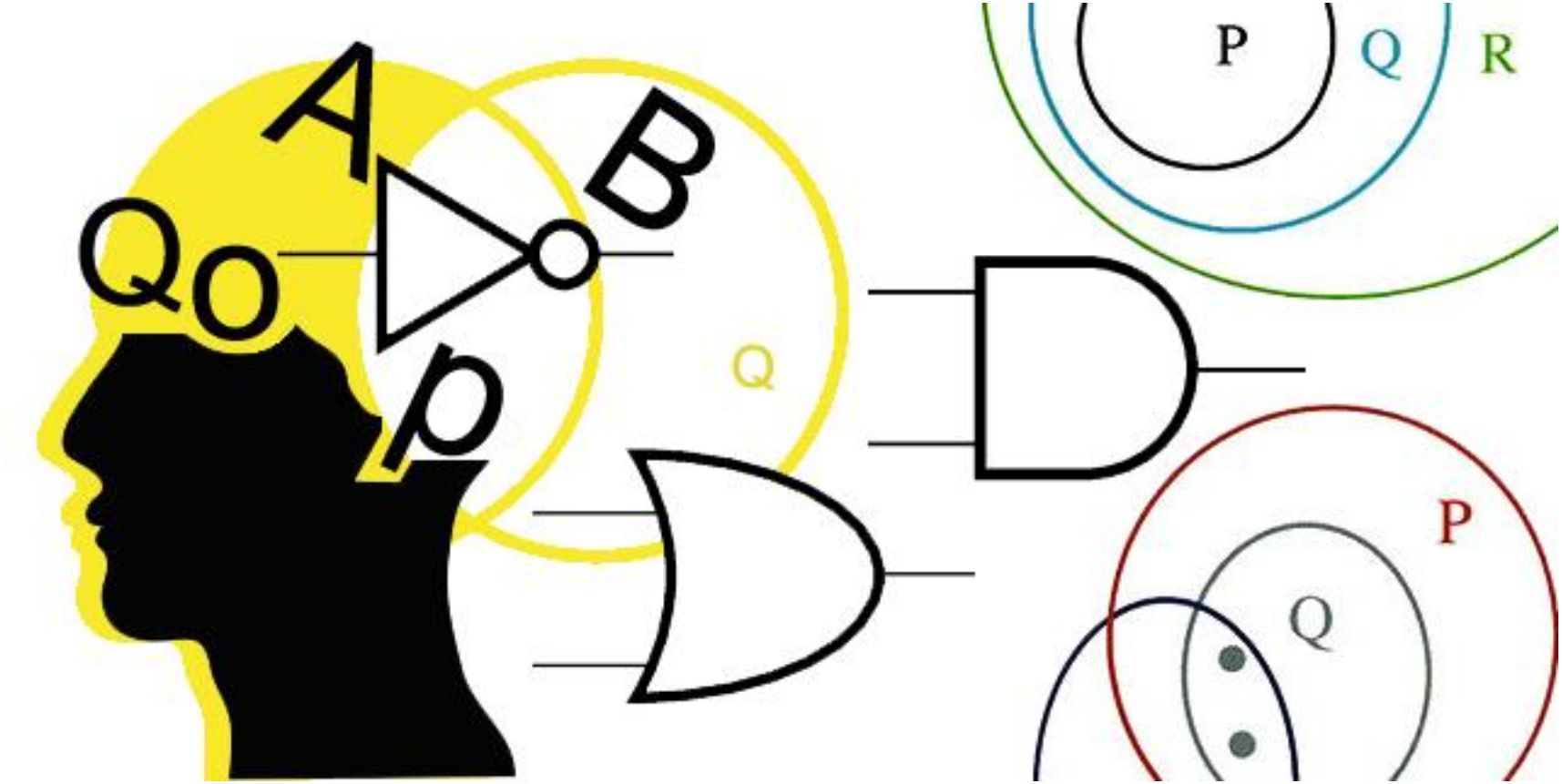
«Quando si diventa fluenti a leggere e scrivere non lo si fa solamente per diventare uno scrittore di professione. Ma imparare a leggere e scrivere è utile a tutti. Ed è la stessa cosa per la programmazione. La maggior parte delle persone non diventerà un esperto di informatica o un programmatore, ma l'abilità di pensare in modo creativo, pensare schematicamente, lavorare collaborando con gli altri […] sono cose che le persone possono usare, indipendentemente dal lavoro che fanno.»

Mitchel Resnick

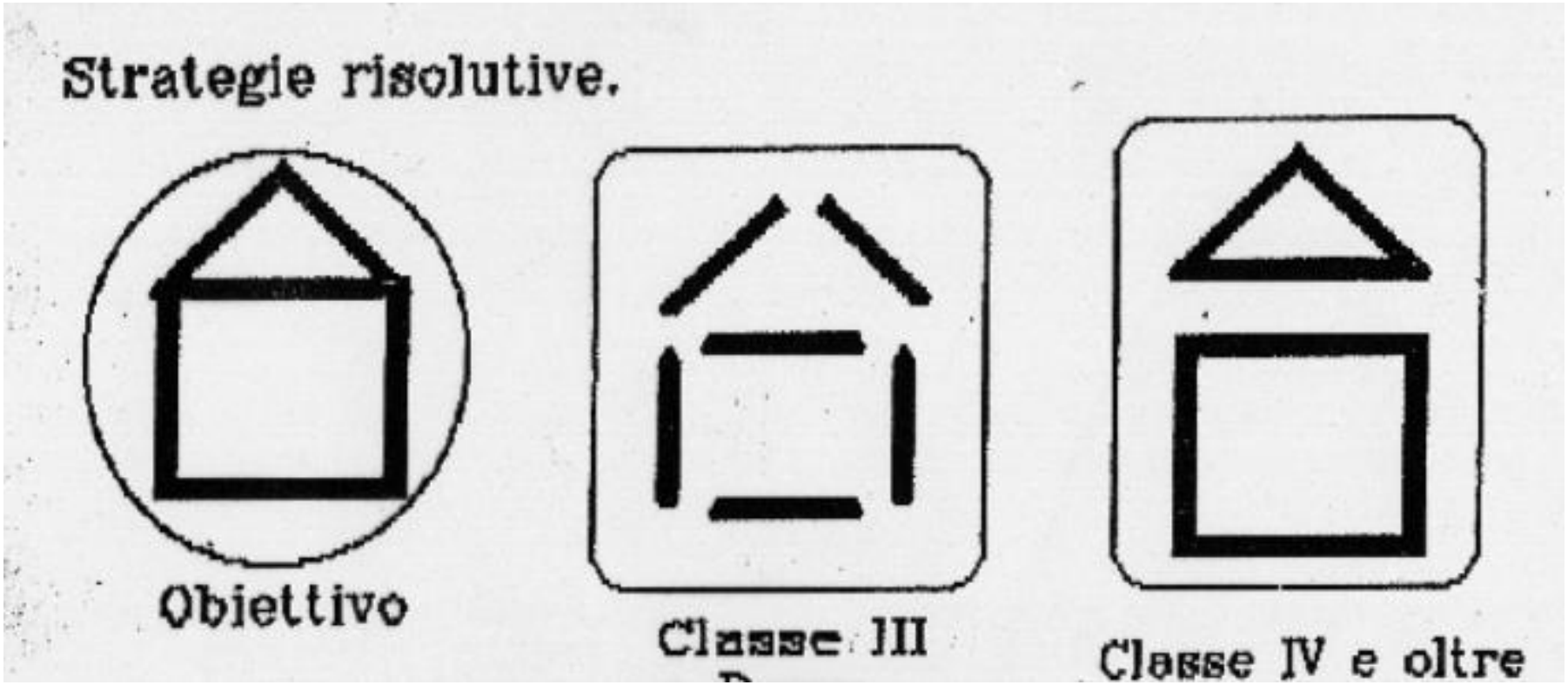
Che cosa rende universale il pensiero computazionale?

- Pensare in modo computazionale significa **suddividere il processo decisionale in singoli step**, ragionare passo passo sul modo migliore per ottenere un obiettivo.
- Si tratta di un comportamento che in realtà, quasi senza accorgercene, mettiamo in atto tutti i giorni, per esempio quando stabiliamo il percorso più breve per raggiungere una destinazione o quando giochiamo ai videogiochi e dobbiamo elaborare un piano per superare un livello.

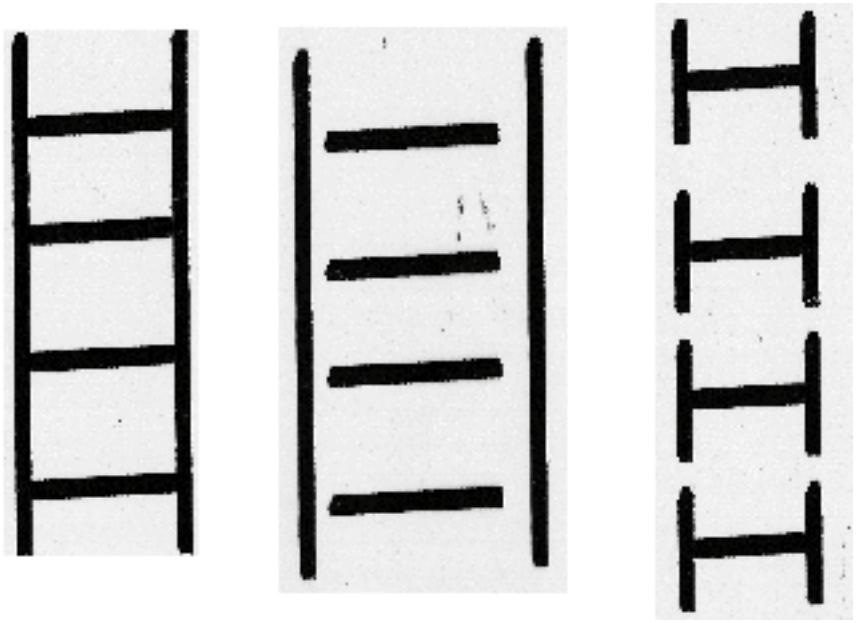
Da dove si parte? Il ragionamento logico



Strategia risolutiva 1



Strategia risolutiva 2



Trova la soluzione

Un **gioco di logica matematica** da proporre ai bambini per aiutarli ad imparare a concentrarsi, ragionare, osservare e trovare la soluzione più adatta. Ognuno di questi quattro folletti deve trovare posto in un triangolo ma... ce ne sono solo tre... Inoltre, tutti i triangoli dovranno avere la stessa dimensione! Dove trovare il quarto?

Stampate la scheda e chiedete ai bambini di ritagliare i diversi elementi prima di cercare di comporli in modo da trovare la soluzione più adatta.

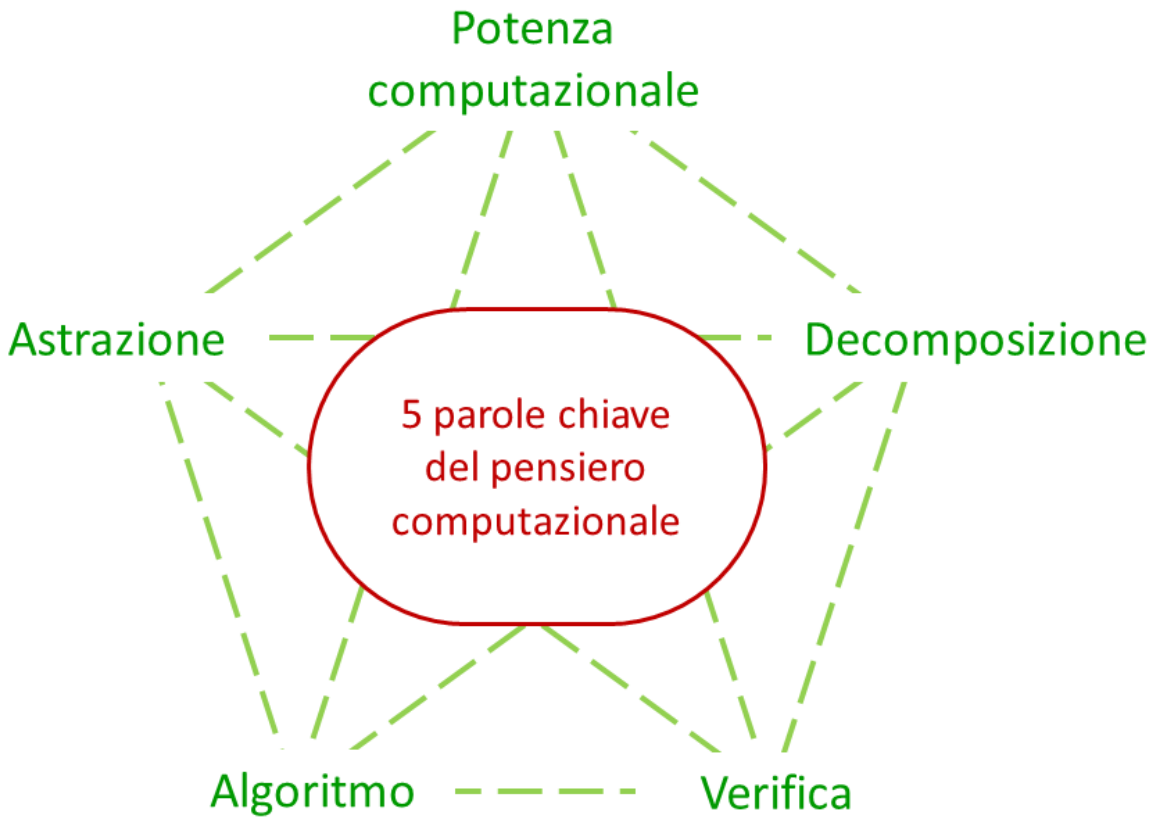
Algoritmi

- La **soluzione di un problema** deve passare attraverso il pensiero algoritmico.
- Un **algoritmo** è una sequenza di passi che devono essere eseguiti secondo un ordine prefissato per raggiungere il risultato voluto.
- Un algoritmo può essere rappresentato con schemi a blocchi ed è stato dimostrato (Teorema di Jacopini-Bohm) che qualunque algoritmo si bassa su tre strutture fondamentali: **sequenziale**, **condizionale**, **iterativa**.

Alcuni esempi di algoritmi

- Quando organizziamo la nostra giornata.
- Quando condividiamo i passi di una ricetta che abbiamo sperimentato.
- Quando facciamo le operazioni aritmetiche.
- Quando spieghiamo un gioco agli altri.
- Quando dobbiamo fornire delle istruzioni per raggiungere un luogo.
- Quando vogliamo costruire una mappa concettuale.

Le parole chiave



http://link-and-think.blogspot.it/2016/04/cinque-parole-chiave-pensiero-computazionale.html

Le parole chiave: livello di astrazione

Come gli spieghi le cose? Il linguaggio deve essere dettagliato?

Per esempio: “riempi la pentola di acqua” - Riempi la pentola a metà, metti la pentola sotto il rubinetto, apri l’acqua fredda, chiudi quando l’acqua raggiunge il livello.

Le parole chiave: decomposizione

Scomporre il problema in tanti piccoli problemi. In questa fase la decomposizione del problema dipende dalla potenza e dal livello di astrazione dell'esecutore: fare una torta al cioccolato.

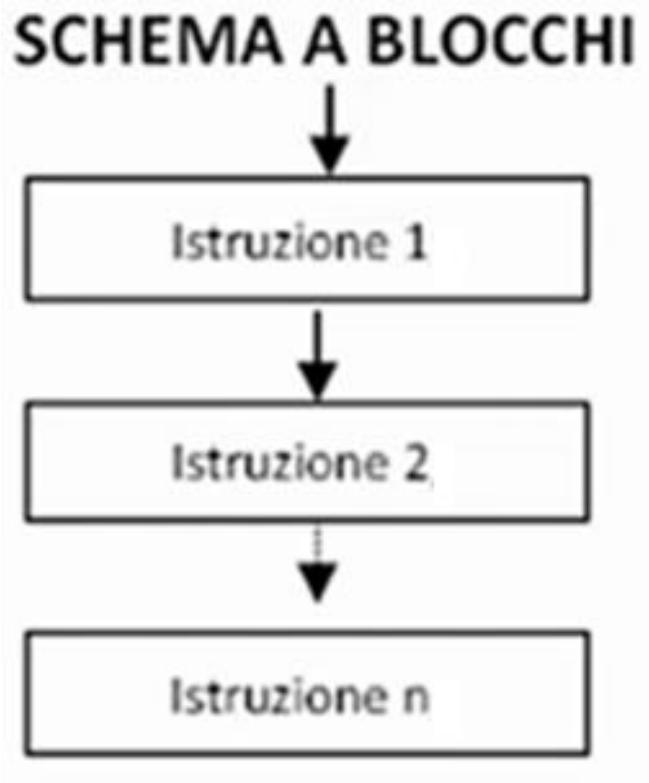
Le parole chiave: algoritmo

Le istruzioni devono essere sequenziate in un ordine logico e preciso per risolvere in maniera efficace il problema.

Le parole chiave: verifica

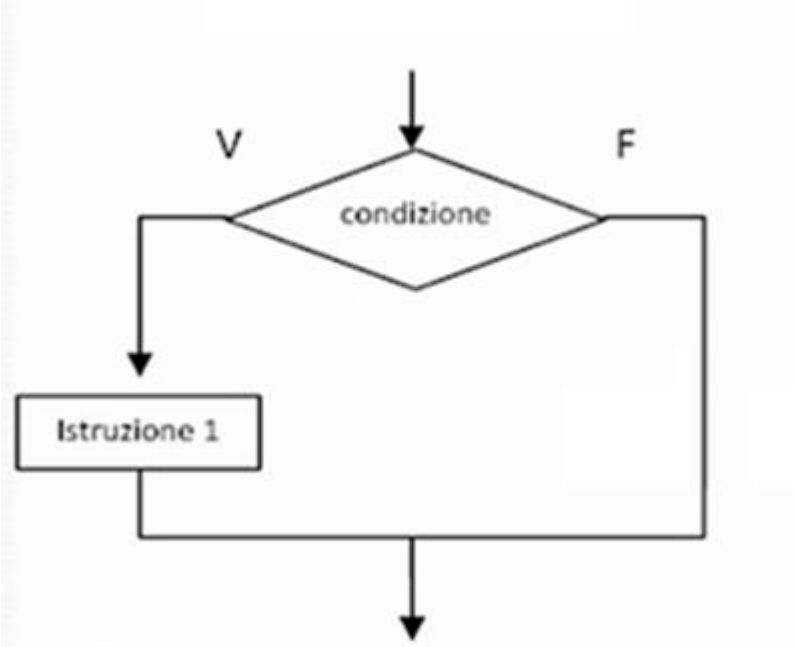
Verificare la correttezza della procedure e delle istruzioni mettendosi nei panni dell'esecutore, importanza del **debugging** cioè del rilevamento e della correzione dell'errore e riformulazione delle procedure.

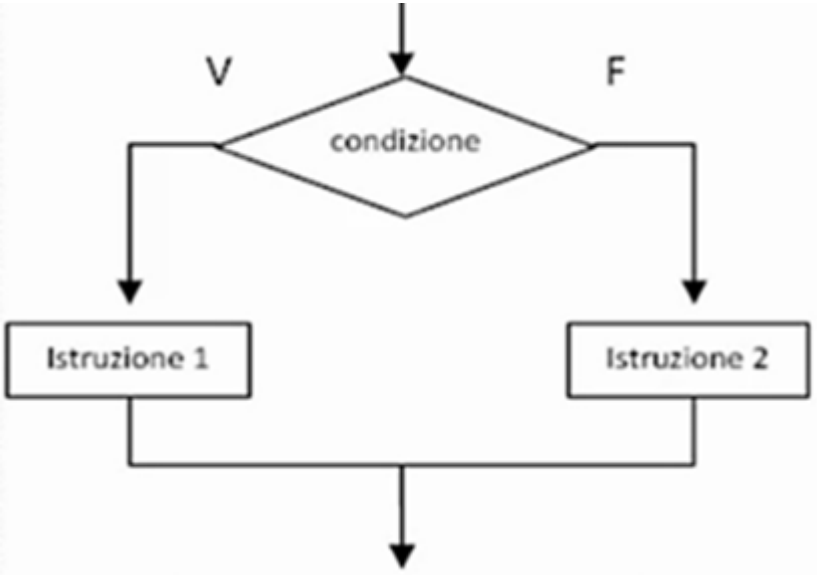
Struttura sequenziale



Struttura di selezione

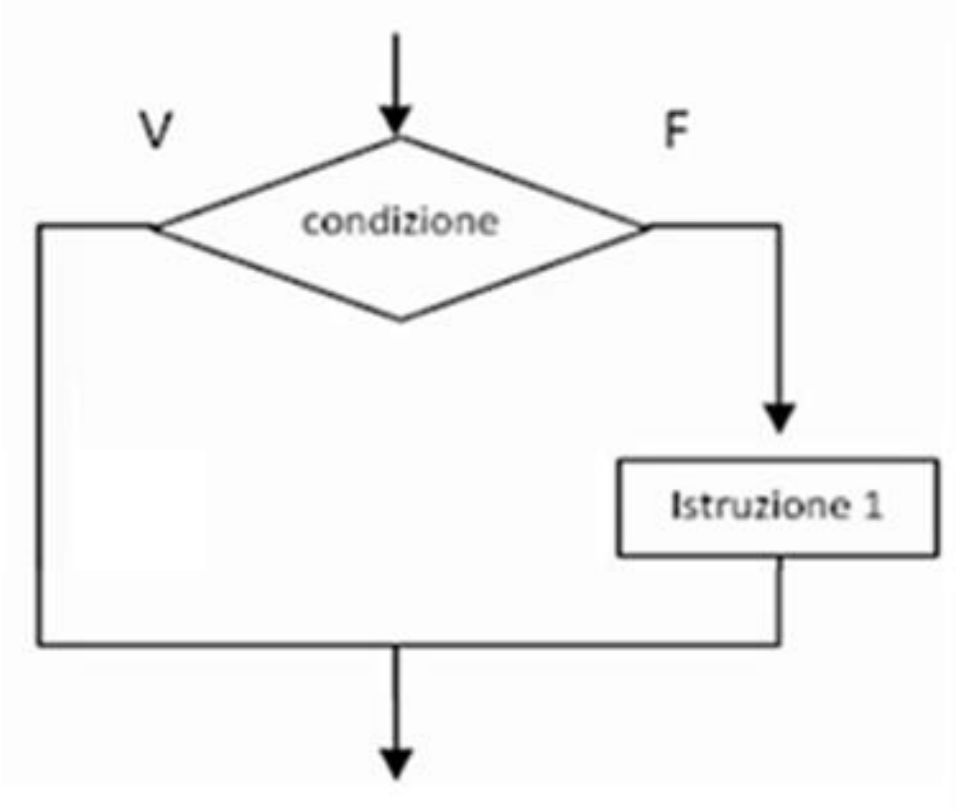
SCHEMA A BLOCCHI





Struttura di selezione

SCHEMA A BLOCCHI



Se… allora…

Se **piove** allora

Se… allora…

Se **piove** allora

altrimenti

Se… allora…

Se **piove** allora

altrimenti

Se… allora…

Se **piove** allora 

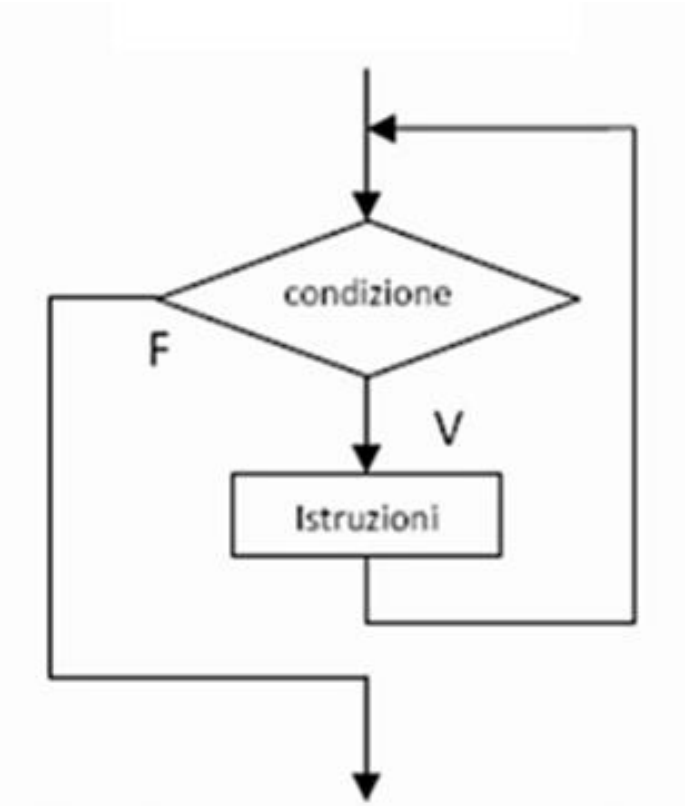
altrimenti 

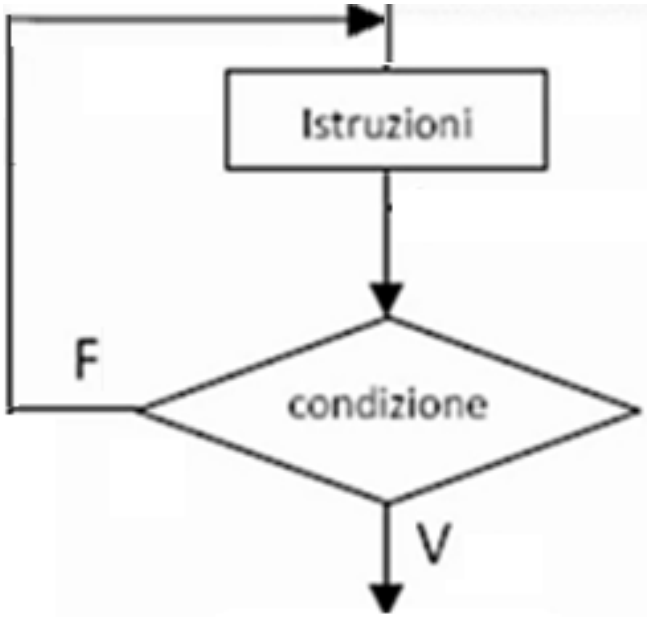
Se **non piove**

allora 

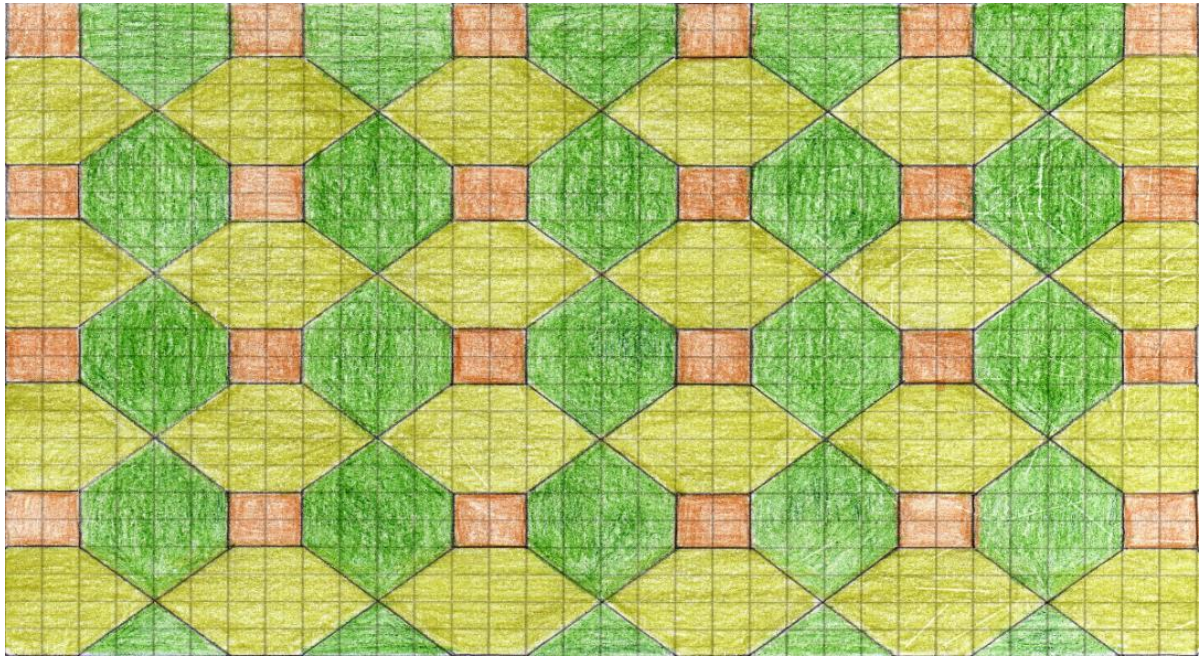
Struttura iterativa

SCHEMA A BLOCCHI



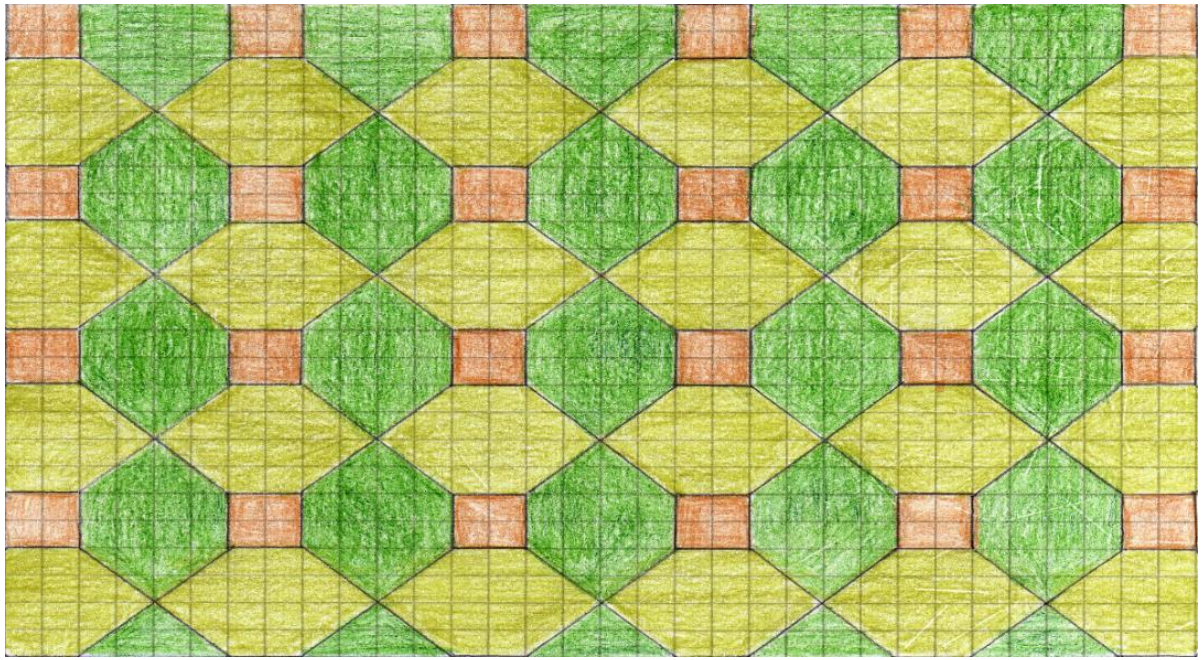


Ripetizione

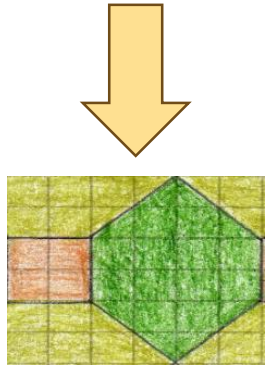


RIPETI n volte… che cosa?

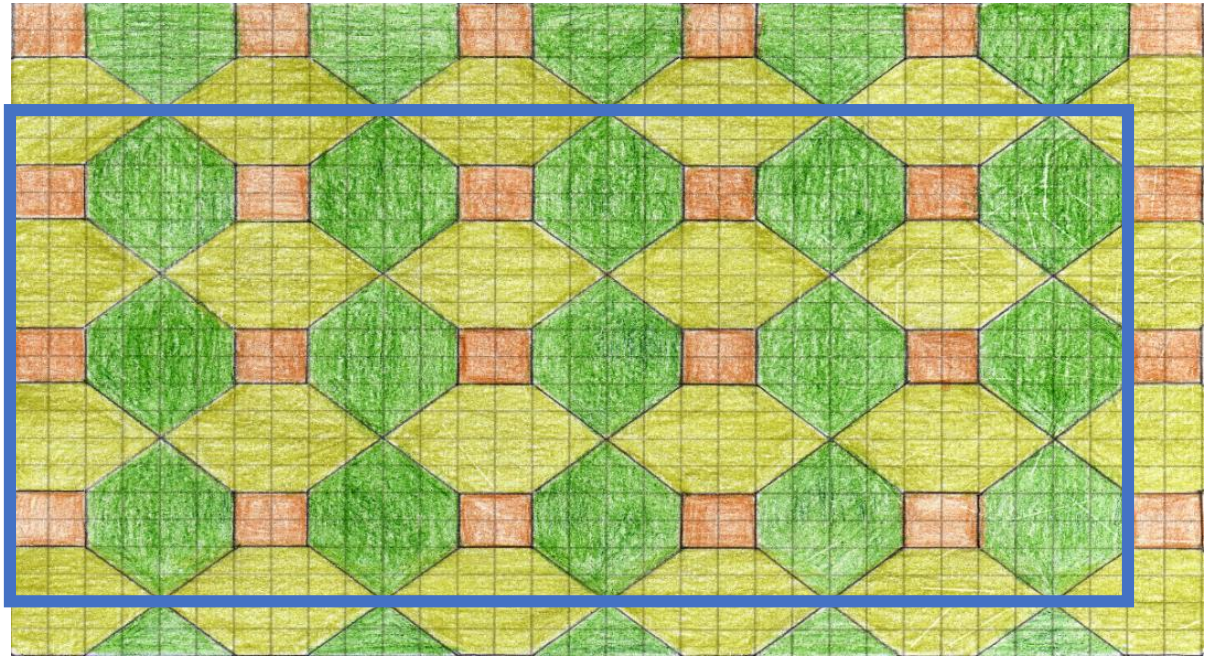
Ripetizione



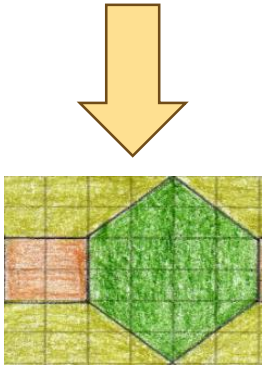
RIPETI
n volte



Ripetizione



RIPETI
n volte



Azioni elementari

- Le indicazioni fornite a un esecutore meccanico (il computer) devono essere **semplici** e **chiare**. A un essere umano, dotato di intelligenza, si possono impartire istruzioni generiche o complesse.
- Per esempio gli si può dire «aggiungi un po' di sale», che è un' istruzione generica perché **«un po'»** non è una quantità precisamente determinata. Oppure gli si può dire «telefonami verso le sette». Anche questa è un'istruzione generica, perché **«verso le sette»** indica un'ora approssimativa, e poi è **complessa**, perché l'azione del telefonare comporta una serie di atti più elementari: alzare la cornetta, comporre il numero, attendere la risposta, chiedere di parlare con una determinata persona.

Azioni elementari o no?

Telefona a Marco

Vai a casa

Apri il libro

Guardiamo il film

Aggiungi 2 a 3

Azioni elementari

Telefona a Marco

- Prendi il telefono
- Accendi il telefono
- Fai il numero
- Premi il tasto  per terminare la chiamata

Istruzioni

L'ISTRUZIONE è un COMANDO che descrive in modo **NON AMBIGUO** un'azione ELEMENTARE

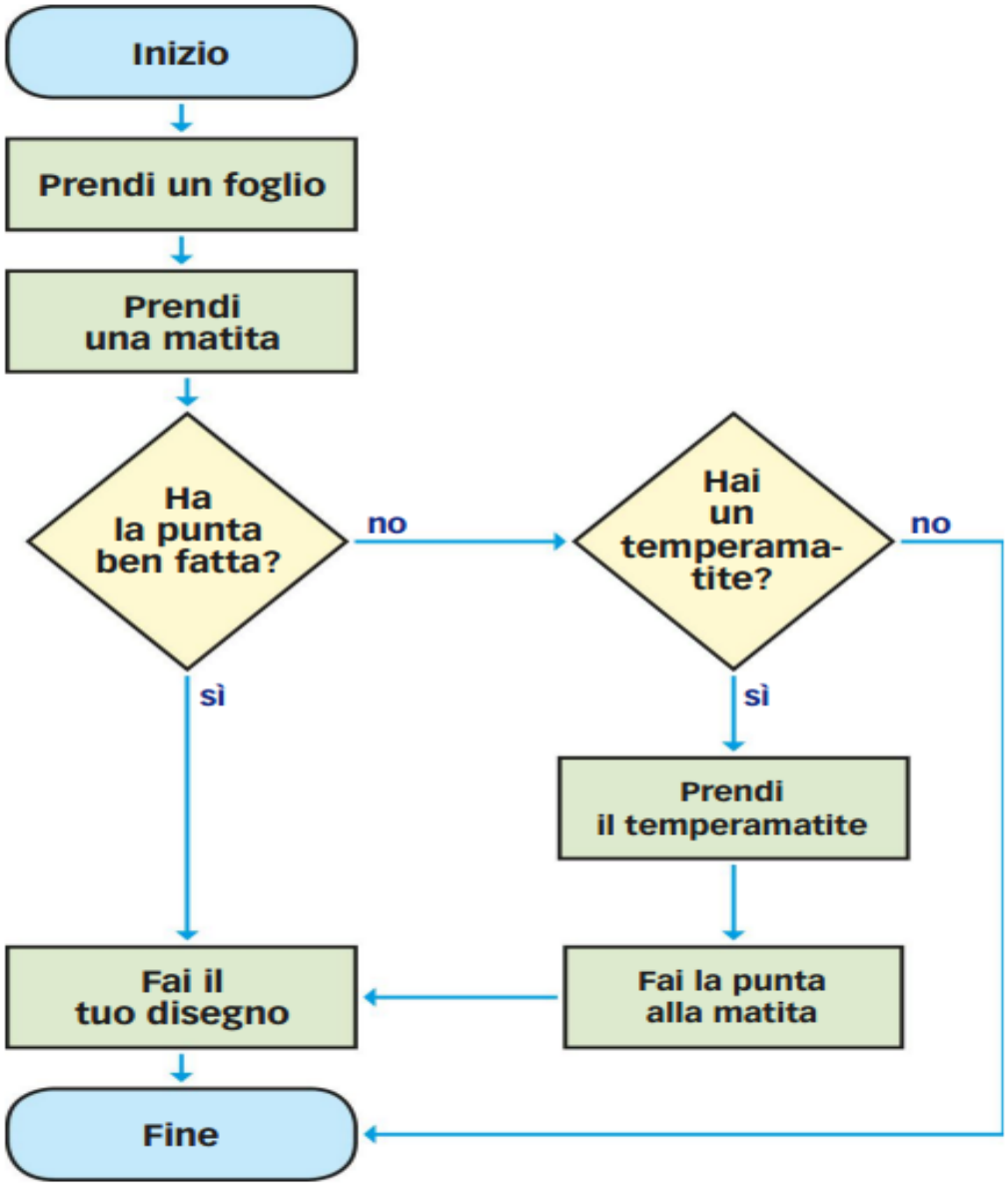
- Qualcuno impartisce
- Qualcuno esegue

} in un determinato contesto

Telefona a Marco

- Prendi il telefono
- Accendi il telefono
- Fai il numero
- Premi il tasto  per terminare la chiamata

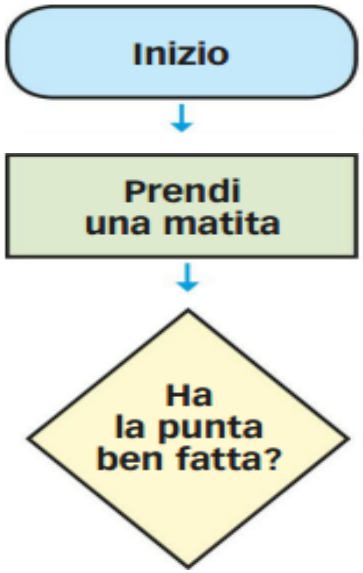
Fare la punta alla matita



Fare la punta alla matita

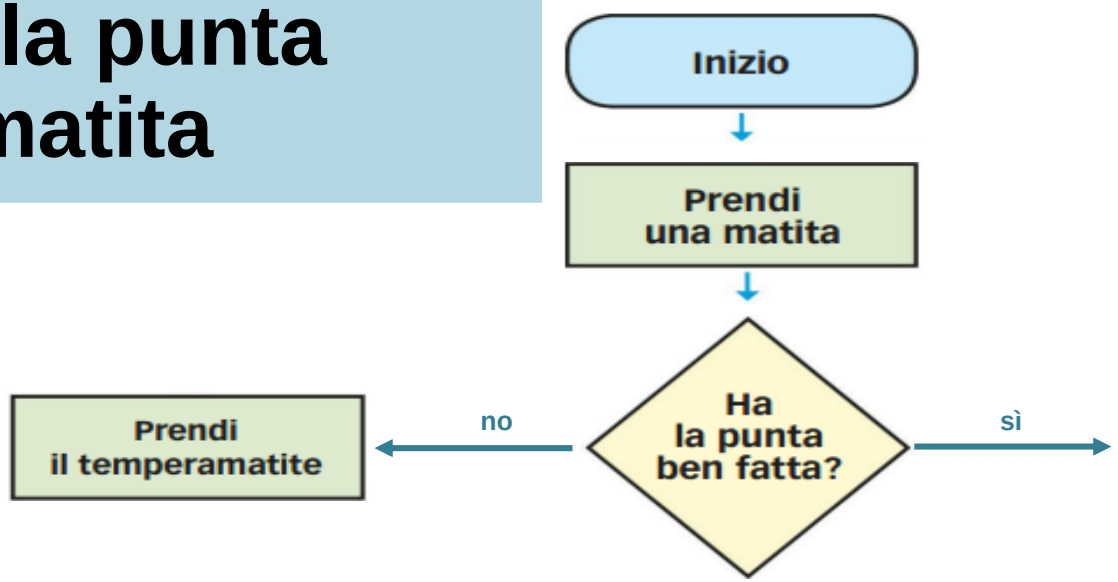
Fare la punta alla matita

Fare la punta alla matita

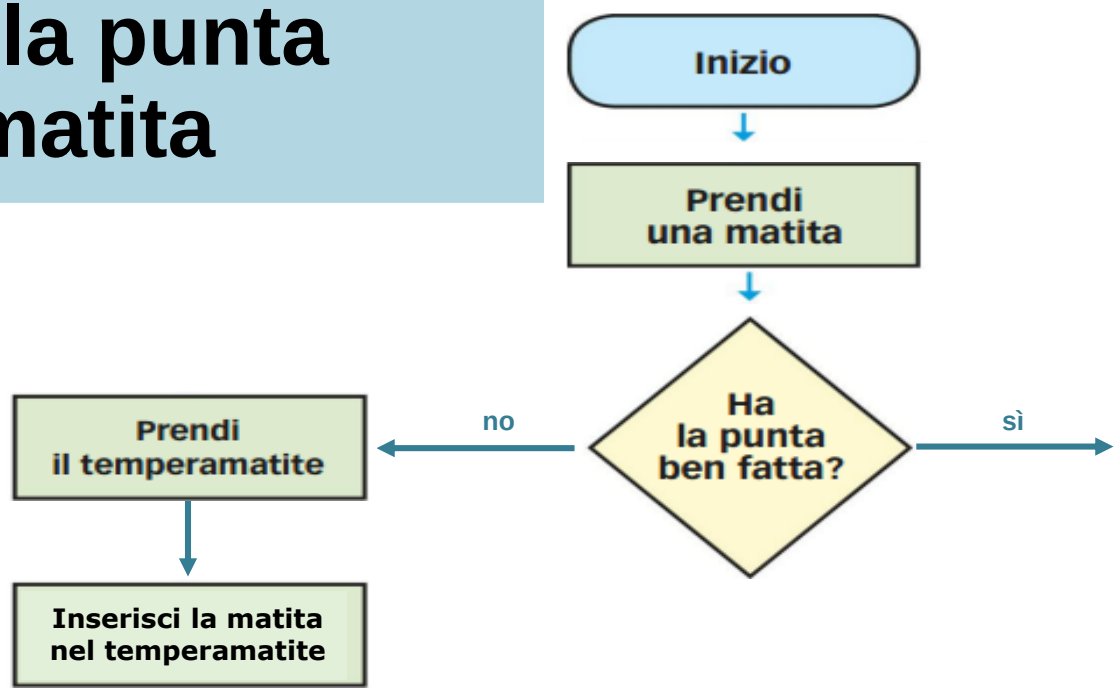


Fare la punta alla matita

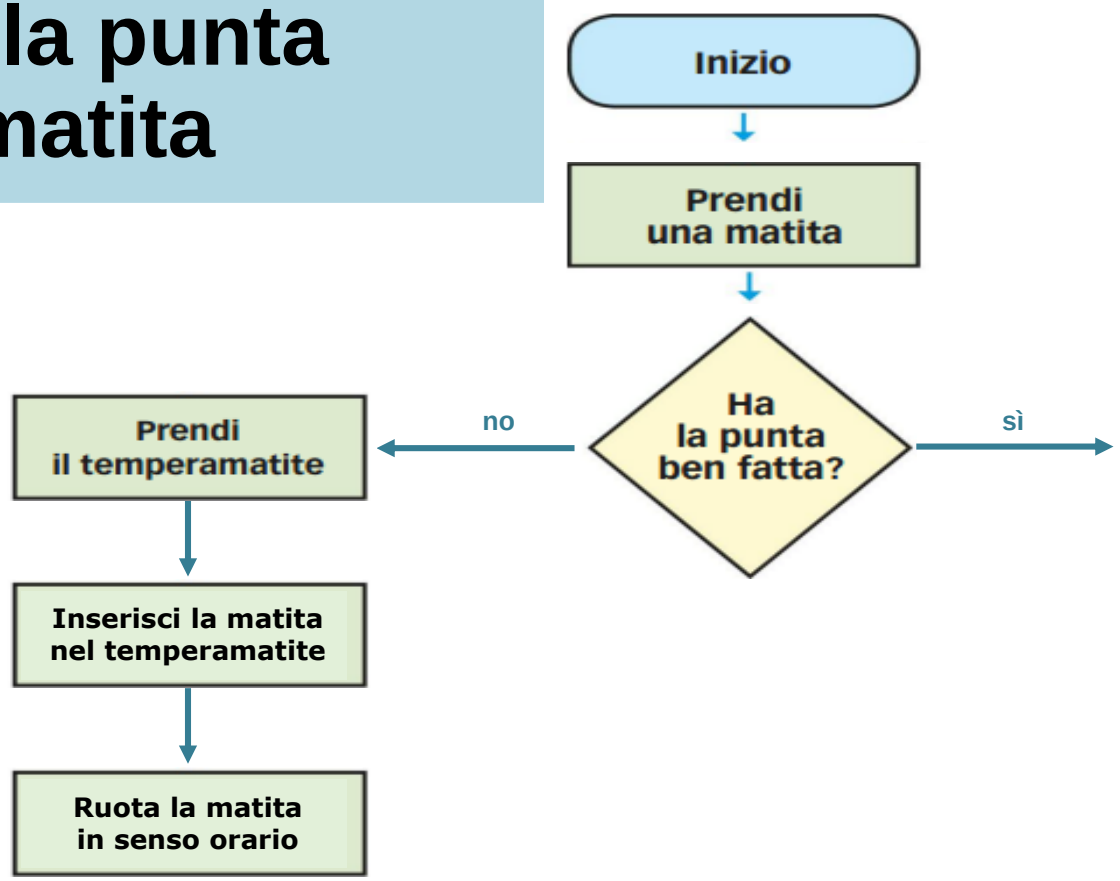
Fare la punta alla matita



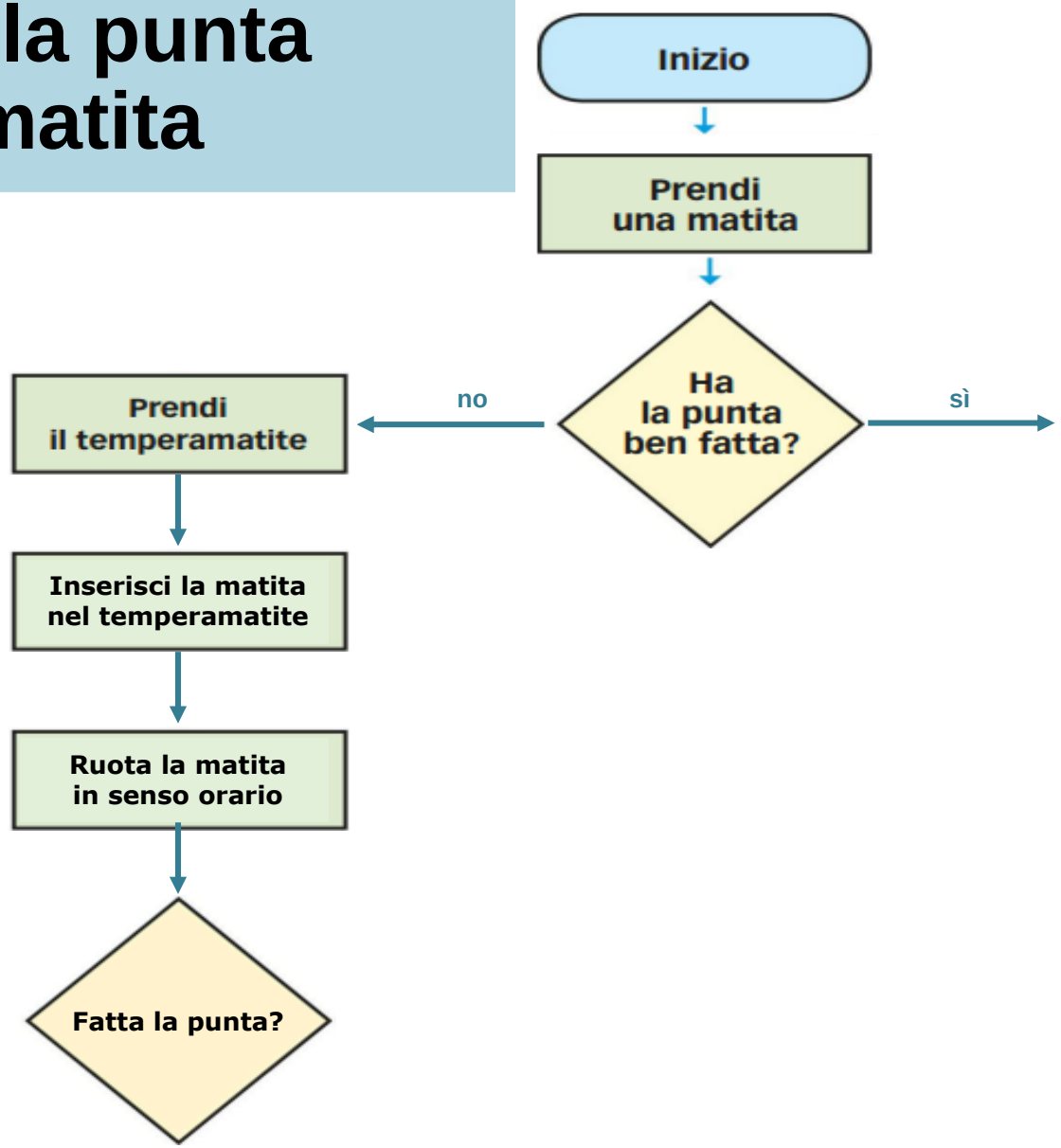
Fare la punta alla matita



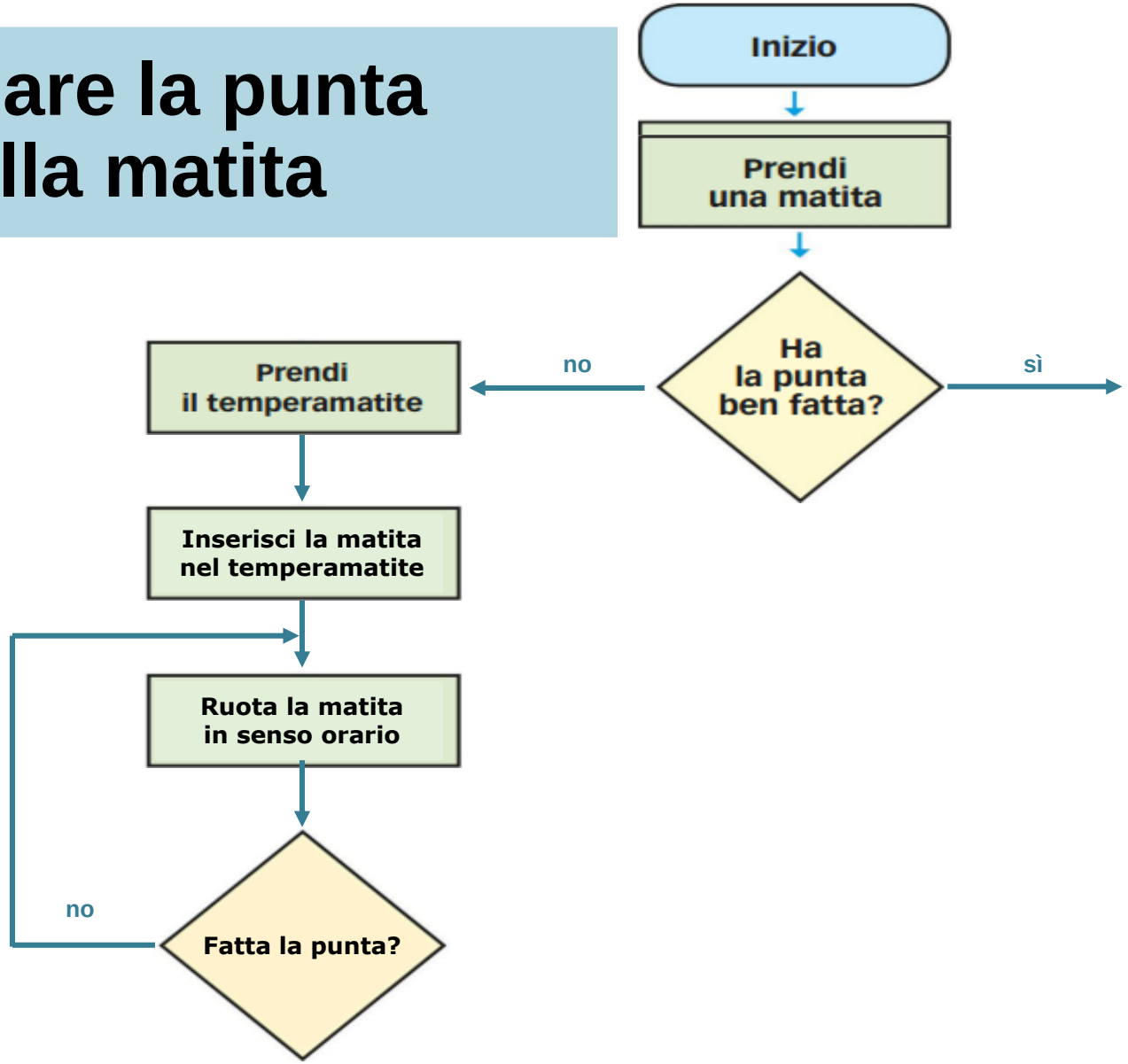
Fare la punta alla matita



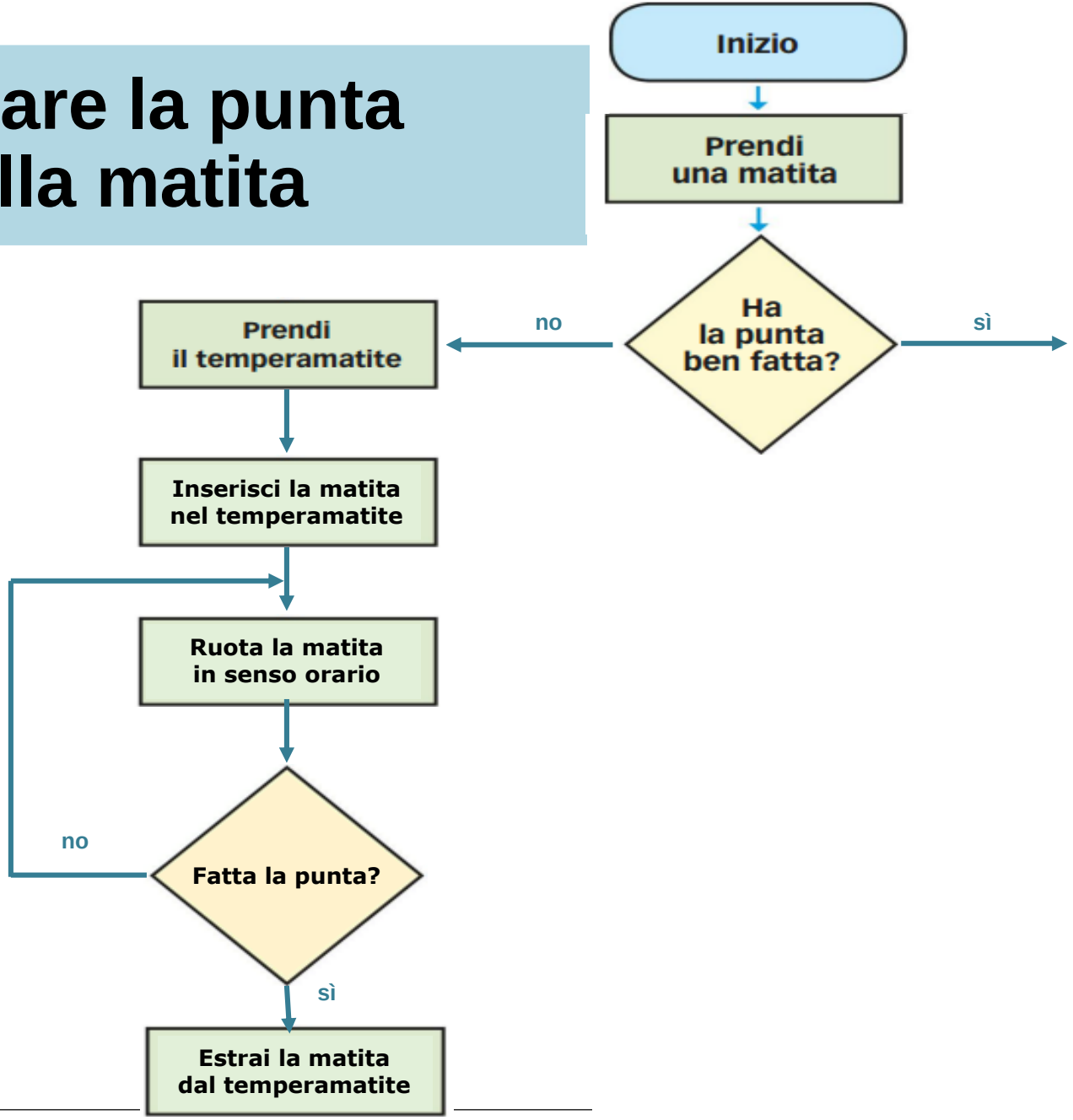
Fare la punta alla matita



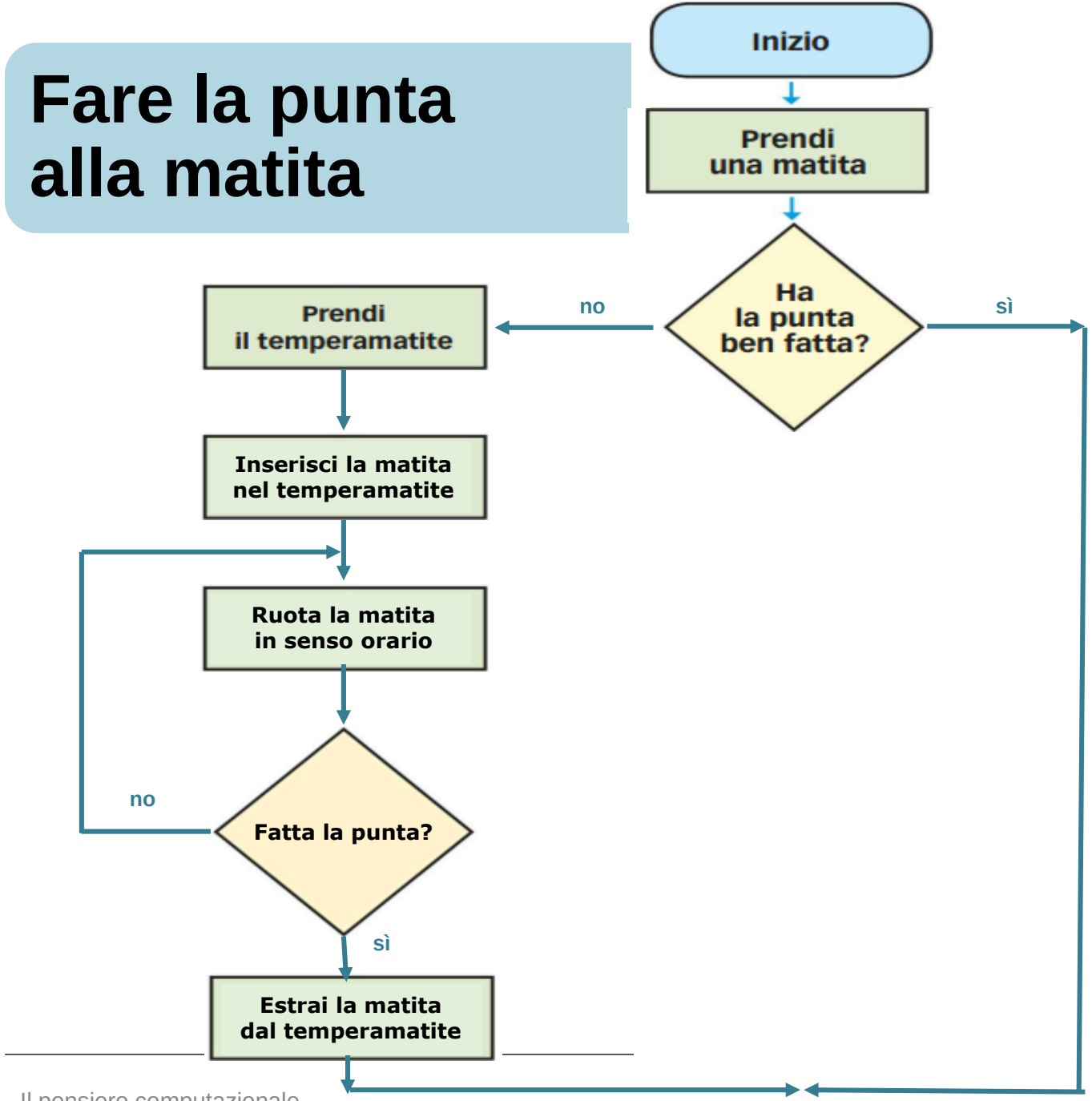
Fare la punta alla matita



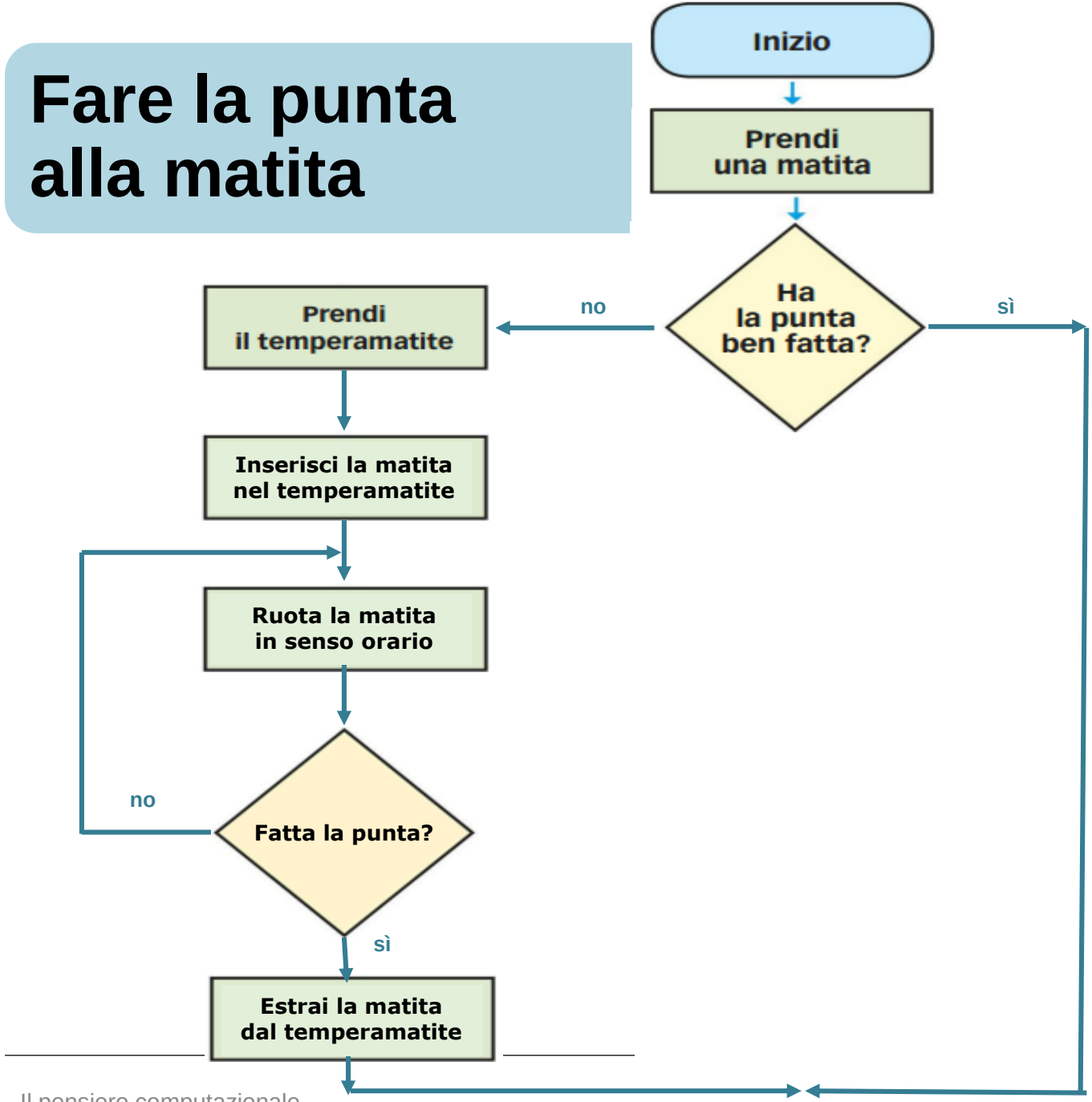
Fare la punta alla matita



Fare la punta alla matita



Fare la punta alla matita



?

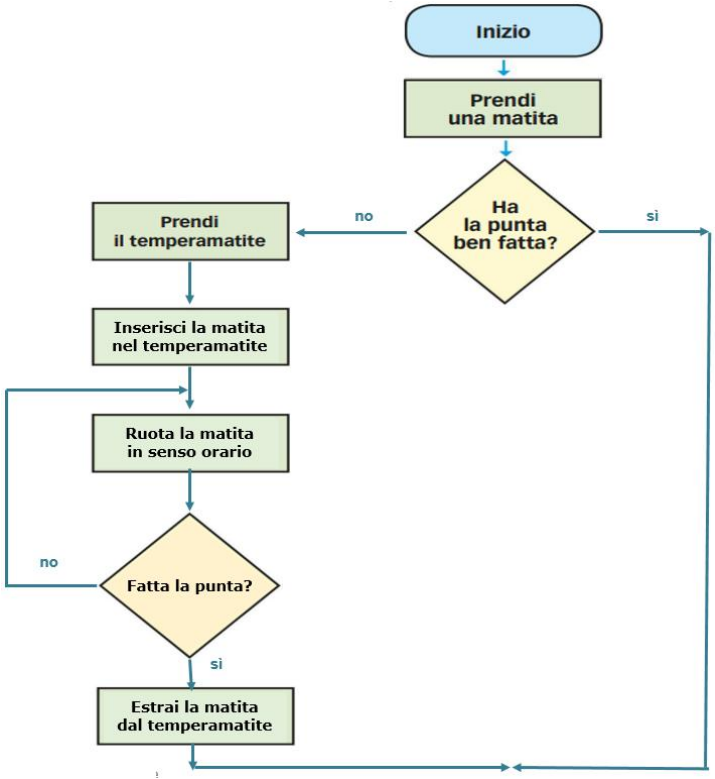
Non va bene! Dobbiamo negare tutto!

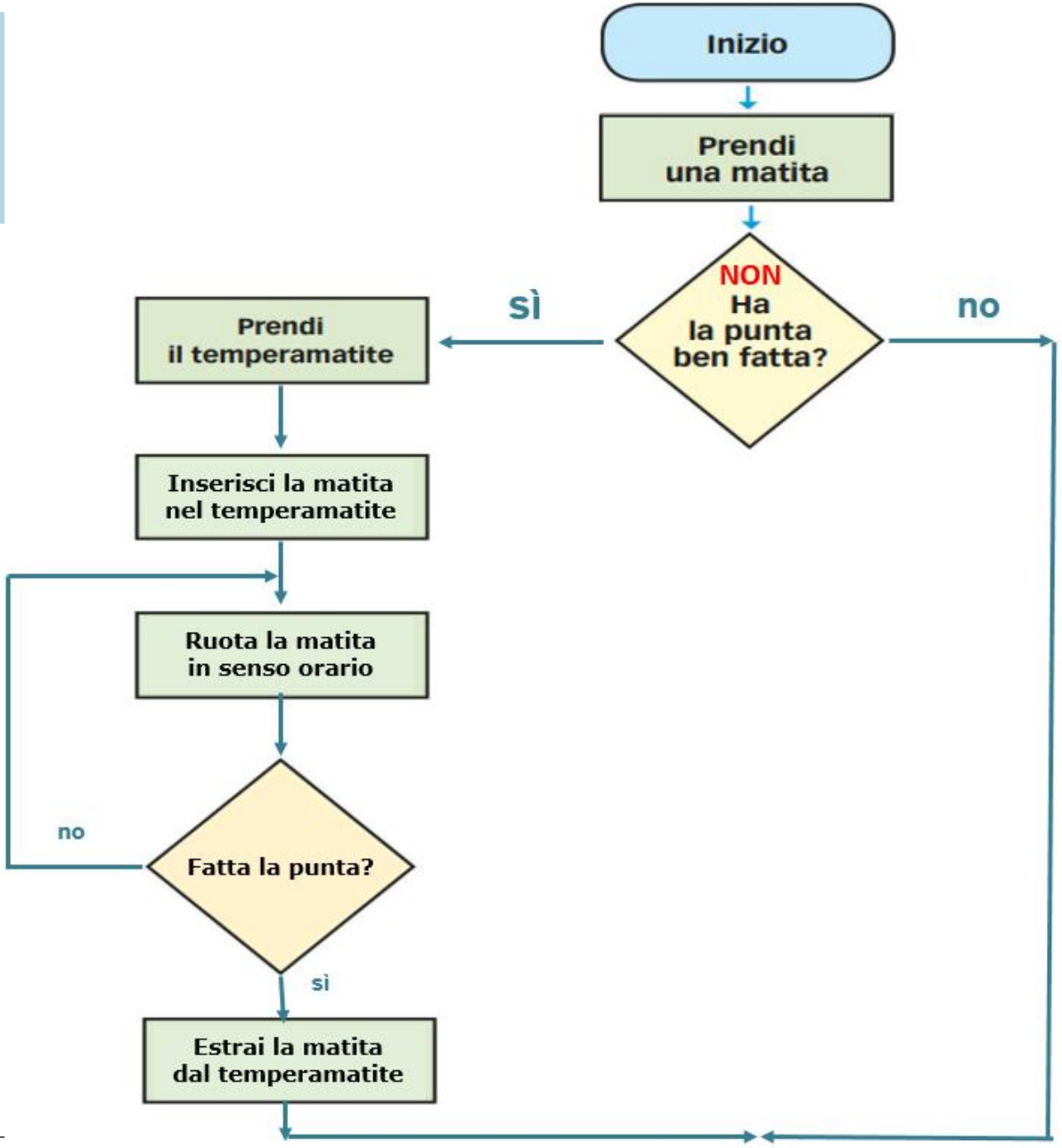
Se **piove** allora

altrimenti

EDUCATION

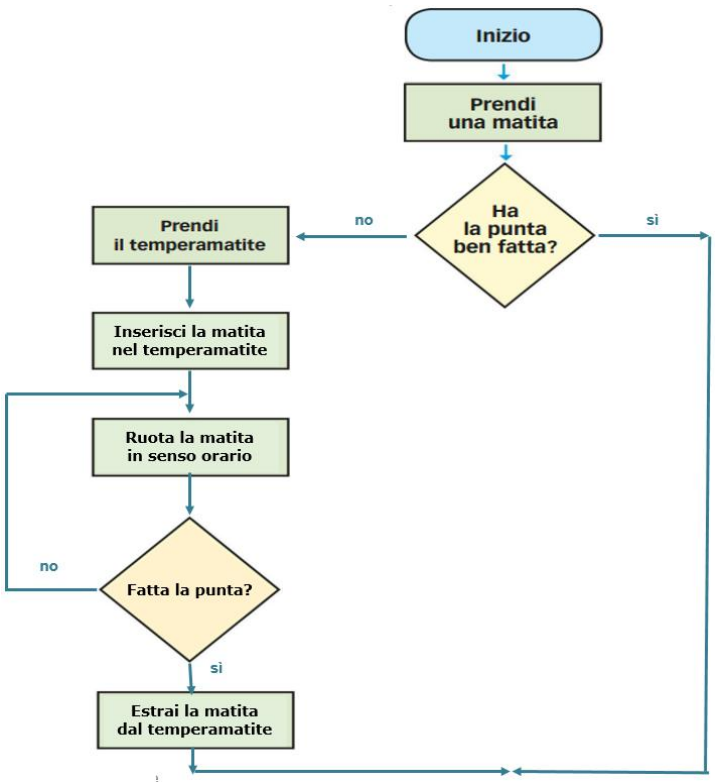
Fare la punta alla matita

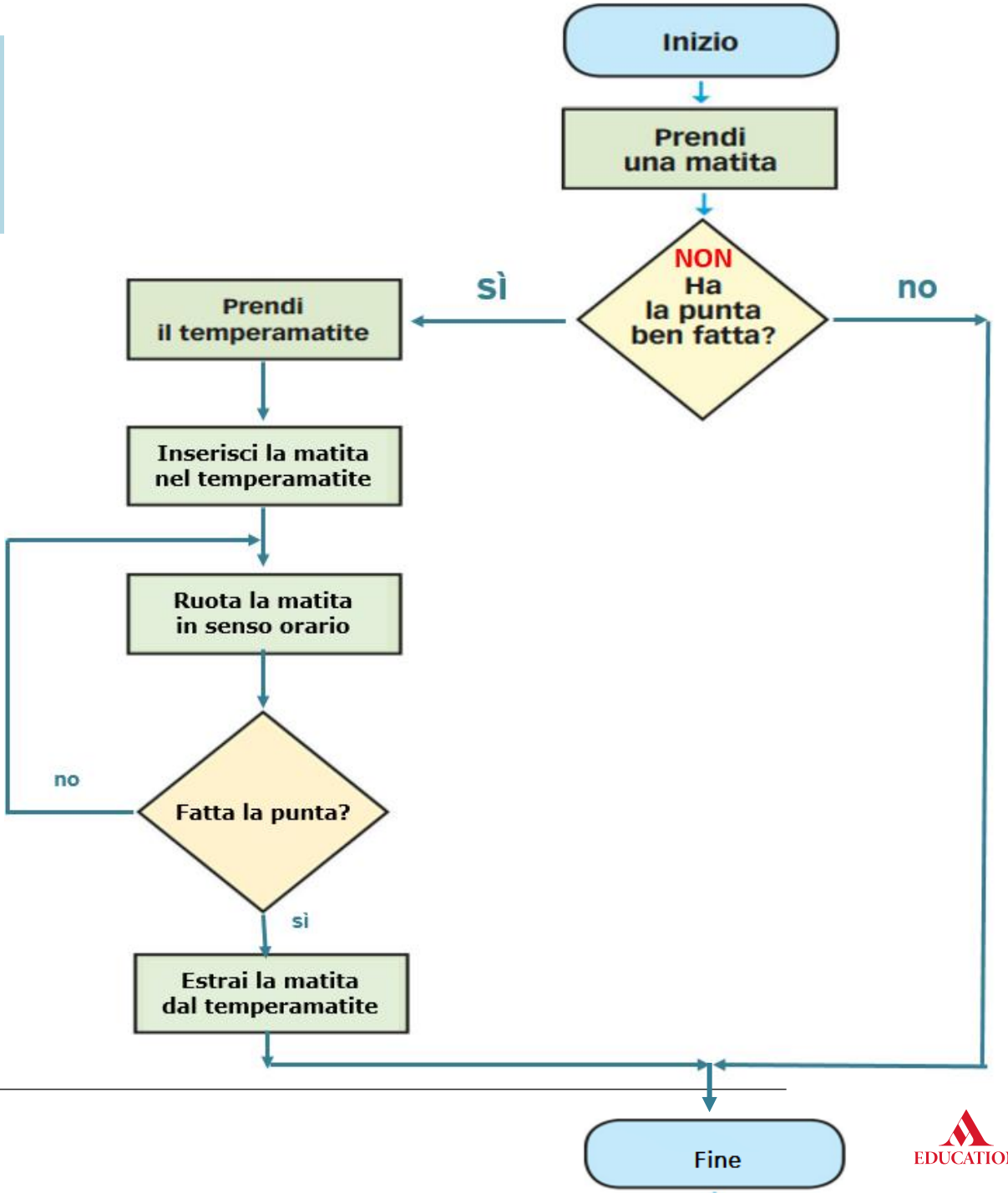




EDUCATION

Fare la punta alla matita





EDUCATION

Il prossimo Webinar sul coding

Coding unplugged: se il computer non c'è!

Elisa Pettinari - martedì 3 ottobre, ore 16.30

L'educazione digitale nella società moderna è diventata un passaggio indispensabile. Introdurre attività che sviluppino le competenze digitali a scuola è facile e non deve spaventare nessuno. Con il progetto **Programma il futuro** il MIUR ha messo a disposizione molti strumenti per farlo che non richiedono specifiche conoscenze, perché non mirano a fornire un contenuto tecnologico, ma a trasmettere l'aspetto scientifico-culturale dell'informatica. Questo aspetto è ciò che viene chiamato **pensiero computazionale**, un modo di pensare e agire che tocca trasversalmente tutti gli ambiti disciplinari.

Vedremo tre esempi concreti di attività che si possono svolgere in classe tra le varie possibili: un'attività di **coding unplugged**, un progetto di **coding** vero e proprio e un esempio di **robotica educativa**.

Il pensiero computazionale è un approccio metodologico e pertanto può essere proposto con modalità diverse. Le attività educative di coding svolte senza l'uso di un mezzo informatico sono dette coding unplugged. Con particolare riferimento alla scuola primaria, vedremo quali sono le proposte di coding unplugged già presenti sui **libri di scuola** e giocheremo insieme a **CodyRoby**, per valutare le opportunità offerte dalla piattaforma ministeriale.

Elisa Pettinari è laureata in ingegneria elettronica al Politecnico di Milano e ha conseguito un master in Comunicazione della Scienza alla SISSA di Trieste. Per Mondadori Education coordina progetti editoriali relativi a informatica, competenze digitali e nuove tecnologie nella didattica, e tiene seminari e corsi di formazione per i docenti. Ha insegnato coding e robotica presso diverse scuole di Milano, sia primaria che secondaria di primo grado, partecipando a progetti volti a introdurre nella programmazione didattica il coding come attività interdisciplinare.

Numero Verde

800 12 39 31

webinar@mondadorieducation.it

www.mondadorieducation.it

MONDADORI EDUCATION