

# DAL PENSIERO COMPUTAZIONALE AL CODING



# CONTENUTI

**I problemi**

**Soluzioni: risultati**

**Soluzioni: algoritmi**

***Problem Solving* & algoritmi**

**Gli algoritmi**

**Il pensiero computazionale**

**I costrutti fondamentali**

**Code Studio**

**Le sequenze in Code Studio**

**I cicli definiti in Code Studio**

**I cicli indefiniti in Code Studio**

**Le scelte semplici in Code Studio**

**Le scelte con alternativa**



# I PROBLEMI

Possono essere:

**Risolvibili**

Risolvibili, con dati superflui

**Non risolvibili** perché mal formulati

**Non risolvibili**

Risolvibili ma con tempi enormi

La soluzione può essere:



Un **RISULTATO**

un numero, una frase, un percorso, un colore,...



Un **PROCEDIMENTO**

una sequenza di istruzioni che portano alla soluzione (algoritmo)

## PROBLEM SOLVING:





# SOLUZIONI: RISULTATI

- **Quanti litri di vernice occorrono per una parete di 4x3 metri, se ogni litro copre 2 metri quadri?**
- **Quale percorso deve effettuare il caricatore per arrivare al furgone nel labirinto a fianco?**
- **Che colore ottieni mescolando eguali quantità di pittura gialla e blu?**
- **Se tre ragazzi salgono su una bilancia, questa segna 120 kg. Se ne scende uno, la bilancia segna 80 kg; se lui risale e ne scende un altro, la bilancia segna 82 kg. Quanto pesano i tre ragazzi?**

La soluzione (se esiste) è una risposta, cioè un risultato



# SOLUZIONI: ALGORITMI

- **Come puoi moltiplicare due numeri interi utilizzando soltanto operazioni di somma?**
- **Hai 27 palline simili, ma una pesa leggermente di più delle altre. Come puoi trovarla con tre pesate su una bilancia a bracci?**
- **Davanti a te hai tre interruttori posizionati su *spento* e una porta chiusa. Solo uno dei tre interruttori accende la lampadina che c'è sulla scrivania nella stanza oltre la porta. Non puoi premere gli interruttori mentre la porta è aperta e una volta entrato non puoi tornare indietro. Come fai a capire quale interruttore accende la lampadina?**



**La soluzione (se esiste) è un procedimento, cioè un algoritmo**

# PROBLEM SOLVING E ALGORITMI

**Problema:** *Un atleta, un ragazzo, un bambino e un anziano vogliono attraversare un ponte che può sopportare al massimo il peso di due persone alla volta. È notte, per attraversare occorre usare una lampada e loro ne hanno una sola. L'atleta può attraversare in 1 minuto, il ragazzo in 2, il bambino in 5 e l'anziano in 10 minuti.*

*Qual è il tempo minimo che possono impiegare per arrivare all'altra sponda?*

Passaggio n°	Verso	Chi	Tempo tot
1			
2			
3			
4			
5			
6			
7			
8			
9			

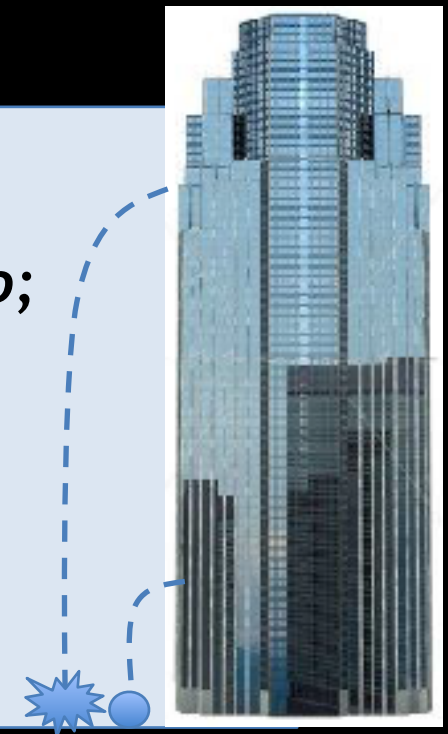


# GLI ALGORITMI

**Problema** : *Ho una sola sfera di metallo; so che se la lascio cadere dal primo piano di un grattacielo non si romperà, ma so che si romperà se la lancio dall'ultimo. Qual è il piano più basso lanciandola dal quale la sfera si romperà?*

La soluzione è un *procedimento* (un algoritmo):

- *So che non la sfera non si rompe se la lascio cadere dal primo piano; quindi inizio dal secondo piano.*
- *Fino a che la sfera non si rompe:*
  - *la lancio dal piano nel quale mi trovo.*
  - *Se si rompe questa è la soluzione;*
  - *Se non si rompe salgo di un piano e riprovo.*



**Le istruzioni di un algoritmo devono essere:**

<b>chiare</b>	chi deve risolvere il problema deve <b>capire</b> ogni singolo passo
<b>eseguibili</b>	chi deve risolvere il problema deve <b>essere in grado</b> di eseguirle
<b>non ambigue</b>	per ogni singolo passo ci deve essere <b>univocità di interpretazione</b>

# IL PENSIERO COMPUTAZIONALE

L'informatica ha introdotto

**nuovi strumenti**  
(computer, *smartphone*, *tablet*,  
navigatori satellitari, ... )

**nuovi metodi**  
per affrontare i problemi:  
il pensiero computazionale.

**Affrontare un problema mediante il pensiero computazionale significa:**

<b>Analizzare</b>	cogliere gli aspetti importanti del problema distinguendoli dai superflui; individuare le risorse necessarie alla sua risoluzione;
<b>Schematizzare</b>	rappresentarlo in maniera schematica;
<b>Scomporre</b>	suddividere il problema in una serie di <i>sotto problemi</i> più semplici da risolvere;
<b>Programmare</b>	ideare la sequenza di passi elementari che risolvono ciascuno dei sotto problemi;
<b>Valutare</b>	esaminare la soluzione trovata in termini di correttezza ed efficienza;
<b>Generalizzare</b>	fare in modo che la soluzione trovata sia applicabile a una vasta gamma di problemi dello stesso tipo.



# I COSTRUTTI FONDAMENTALI

I passi elementari che risolvono un problema si scompongono in:

<b>sequenze</b>	<i>prima mi spoglio, poi indosso il pigiama, poi vado a letto</i>
<b>cicli definiti</b>	<i>fino a che non ho percorso 10 giri continuo a correre</i>
<b>cicli indefiniti</b>	<i>fino a che non ho capito continuo a studiare</i>
<b>scelte</b>	<i>se domani ho compito allora studio altrimenti guardo un film</i>
<b>eventi</b>	<i>quando sono le cinque inizio a preparare la borsa</i>
<b>attività contemporanee</b>	<i>faccio merenda e nel frattempo ascolto la musica</i>

# CODE STUDIO

Lezione

Esercizio

The screenshot shows the Code Studio interface for a lesson titled "Lezione 2: Il labirinto". The interface is divided into several sections:

- Game Area (Left):** A maze game with a character and various plants. A red arrow points to this area with the text "Spazio di gioco: qui si muovono i personaggi".
- Block Palette (Middle-Left):** A list of blocks to be used, including "vai avanti", "gira a sinistra", "gira a destra", "ripeti fino a che", and "se c'è strada a destra". A red arrow points to this area with the text "Blocchi da utilizzare".
- Work Area (Right):** A workspace where blocks are dragged and assembled into a script. A red arrow points to this area with the text "Spazio di lavoro: qui trascini i blocchi".
- Header:** Shows the lesson title "Lezione 2: Il labirinto" and a progress indicator with 17 steps. A red arrow points to the progress indicator.
- Footer:** A message from a character: "Ok, esercitiamoci di nuovo per far pratica: questo esercizio non è molto diverso, ma stai attento alle piante carnivore!".

Spazio di gioco:  
qui si muovono i  
personaggi

Blocchi da  
utilizzare

Spazio di lavoro: qui  
trascini i blocchi

# LE SEQUENZE IN CODE STUDIO

Qual è la corretta sequenza di istruzioni da fornire al pulcino affinché possa raggiungere il maialino?

**1**

- Gira a sinistra di 90°
- Vai avanti
- Gira a destra di 90°
- Vai avanti
- Vai avanti
- Vai avanti
- Gira a sinistra di 90°
- Vai avanti
- Vai avanti
- Vai avanti

**2**

- Gira a sinistra di 90°
- Vai avanti
- Gira a destra di 90°
- Vai avanti
- Vai avanti
- Gira a sinistra di 90°
- Vai avanti
- Vai avanti
- Vai avanti

Poste in ordine diverso, le istruzioni corrette portano al medesimo risultato?



# I CICLI DEFINITI IN CODE STUDIO

Qual è il blocco di istruzioni più semplici da fornire al pulcino affinché possa raggiungere il maialino?

The image shows a Code Studio interface. On the left is a 10x10 green grid. A yellow chick is at (1,2), a pink pig is at (4,2), and several black and red striped obstacles are placed at (1,5), (2,3), (2,4), (2,5), (3,1), (3,2), (3,3), (3,4), (4,3), (4,4), and (5,3), (5,4). A red arrow points from the grid to a code block labeled '1'. The code block is a pink 'Ripeti 3 volte' (Repeat 3 times) block containing four blue blocks: 'Gira a destra di 90°', 'Vai avanti', 'Vai avanti', and 'Vai avanti'. To the right of the code block is a vertical stack of blue blocks labeled '2', representing the execution of the loop: 'Gira a destra di 90°', 'Vai avanti', 'Vai avanti', 'Vai avanti', 'Gira a destra di 90°', 'Vai avanti', 'Vai avanti', 'Vai avanti', 'Gira a destra di 90°', 'Vai avanti', 'Vai avanti', 'Vai avanti'.

In questo modo si specifica quante volte le istruzioni del blocco saranno eseguite



# I CICLI INDEFINITI IN CODE STUDIO

Qual è il blocco di istruzioni più semplici da fornire al pulcino affinché possa raggiungere il maialino?

The image shows a 10x10 grid maze. A yellow chick is at the bottom center (row 8, column 3). A pink pig is at row 4, column 2. Black obstacles are at (3,4), (4,1), (4,4), (4,7), (5,1), (5,2), (5,3), (5,7), and (6,4). A red arrow points from the text below to the 'esegui' block in the code editor.

**1**

- Gira a sinistra di 90°
- Vai avanti
- Vai avanti
- Gira a sinistra di 90°
- Vai avanti
- Vai avanti
- Gira a sinistra di 90°
- Vai avanti
- Vai avanti

**2**

Fino a

- Gira a sinistra di 90°
- Vai avanti
- Vai avanti

esegui

In questo modo si specifica quando le istruzioni del blocco *esegui* non saranno più eseguite

# LE SCELTE SEMPLICI IN CODE STUDIO

Qual è il blocco di istruzioni corretto da fornire al pulcino affinché possa raggiungere il maialino?

The grid world contains a chicken at (row, col) (6, 7), a pig at (6, 2), and two obstacles at (2, 3) and (7, 2). The chicken is currently at (6, 7) and needs to reach the pig at (6, 2).

**1** Fino a che non raggiunge il maialino  
Se c'è strada a sinistra  
esegui  
Gira a sinistra di 90°  
Vai avanti

**2** Fino a che non raggiunge il maialino  
Se c'è strada a sinistra  
esegui  
Gira a sinistra di 90°  
Vai avanti

Le istruzioni del blocco *esegui* saranno eseguite solo se la condizione specificata è vera

