



**INFORMATIK-BIBER SCHWEIZ  
CASTOR INFORMATIQUE SUISSE  
CASTORO INFORMATICO SVIZZERA**

## Quesiti e soluzioni 2019 3<sup>o</sup> e 4<sup>o</sup> anno scolastico

<https://www.castoro-informatico.ch/>

A cura di:

Lucio Negrini, Christian Datzko, Susanne Datzko, Juraj Hromkovič, Regula Lacher

010100110101011001001001  
010000010010110101010011  
010100110100100101000101  
001011010101001101010011  
010010010100100100100001

**SS! I**

[www.svia-ssie-ssii.ch](http://www.svia-ssie-ssii.ch)  
schweizerischerverein für informatik in d  
erausbildung // société suisse pour l'infor  
matique dans l'enseignement // società sviz  
zera per l'informatica nell'insegnamento







# Hanno collaborato al Castoro Informatico 2019

Christian Datzko, Susanne Datzko, Olivier Ens, Hanspeter Erni, Nora A. Escherle, Martin Guggisberg, Saskia Howald, Lucio Negrini, Gabriel Parriaux, Elsa Pellet, Jean-Philippe Pellet, Beat Trachsler.

Un particolare ringraziamento va a:

Juraj Hromkovič, Michelle Barnett, Michael Barot, Anna Laura John, Dennis Komm, Regula Lacher, Jacqueline Staub, Nicole Trachsler: ETHZ

Gabriel Thullen: Collège des Colombières

Valentina Dagienė: Bebras.org

Wolfgang Pohl, Hannes Endreß, Ulrich Kiesmüller, Kirsten Schlüter, Michael Weigend: Bundesweite Informatikwettbewerbe (BWINF), Germania

Chris Roffey: University of Oxford, Regno Unito

Carlo Bellettini, Violetta Lonati, Mattia Monga, Anna Morpurgo: ALaDDIn, Università degli Studi di Milano, Italia

Gerald Futschek, Wilfried Baumann, Florentina Voboril: Oesterreichische Computer Gesellschaft, Austria

Zsuzsa Pluhár: ELTE Informatikai Kar, Ungheria

Eljakim Schrijvers, Justina Dauksaite, Arne Heijenga, Dave Oostendorp, Andrea Schrijvers, Kyra Willekes, Saskia Zweerts: Cuttle.org, Paesi Bassi

Christoph Frei: Chragokyberneticks (Logo Castoro Informatico Svizzera)

Andrea Leu, Maggie Winter, Brigitte Manz-Brunner: Senarclens Leu + Partner

L'edizione dei quesiti in lingua tedesca è stata utilizzata anche in Germania e in Austria.

La traduzione francese è stata curata da Elsa Pellet mentre quella italiana da Veronica Ostini.



**INFORMATIK-BIBER SCHWEIZ**  
**CASTOR INFORMATIQUE SUISSE**  
**CASTORO INFORMATICO SVIZZERA**

Il Castoro Informatico 2019 è stato organizzato dalla Società Svizzera per l'Informatica nell'Insegnamento SSII con il sostegno della fondazione Hasler.

## HASLERSTIFTUNG

Nota: Tutti i link sono stati verificati l'01.11.2019. Questo quaderno è stato creato il 2 gennaio 2020 col sistema per la preparazione di testi  $\text{\LaTeX}$ .



I quesiti sono distribuiti con Licenza Creative Commons Attribuzione – Non commerciale – Condividi allo stesso modo 4.0 Internazionale. Gli autori sono elencati a pagina 28.



## Premessa

Il concorso del “Castoro Informatico”, presente già da diversi anni in molti paesi europei, ha l’obiettivo di destare l’interesse per l’informatica nei bambini e nei ragazzi. In Svizzera il concorso è organizzato in tedesco, francese e italiano dalla Società Svizzera per l’Informatica nell’Insegnamento (SSII), con il sostegno della fondazione Hasler nell’ambito del programma di promozione “FIT in IT”.

Il Castoro Informatico è il partner svizzero del Concorso “Bebras International Contest on Informatics and Computer Fluency” (<https://www.bebas.org/>), situato in Lituania.

Il concorso si è tenuto per la prima volta in Svizzera nel 2010. Nel 2012 l’offerta è stata ampliata con la categoria del “Piccolo Castoro” (3<sup>o</sup> e 4<sup>o</sup> anno scolastico).

Il “Castoro Informatico” incoraggia gli alunni ad approfondire la conoscenza dell’informatica: esso vuole destare interesse per la materia e contribuire a eliminare le paure che sorgono nei suoi confronti. Il concorso non richiede alcuna conoscenza informatica pregressa, se non la capacità di “navigare” in internet poiché viene svolto online. Per rispondere alle domande sono necessari sia un pensiero logico e strutturato che la fantasia. I quesiti sono pensati in modo da incoraggiare l’utilizzo dell’informatica anche al di fuori del concorso.

Nel 2019 il Castoro Informatico della Svizzera è stato proposto a cinque differenti categorie d’età, suddivise in base all’anno scolastico:

- 3<sup>o</sup> e 4<sup>o</sup> anno scolastico (“Piccolo Castoro”)
- 5<sup>o</sup> e 6<sup>o</sup> anno scolastico
- 7<sup>o</sup> e 8<sup>o</sup> anno scolastico
- 9<sup>o</sup> e 10<sup>o</sup> anno scolastico
- 11<sup>o</sup> al 13<sup>o</sup> anno scolastico

Alla categoria del 3<sup>o</sup> e 4<sup>o</sup> anno scolastico sono stati assegnati 9 quesiti da risolvere, di cui 3 facili, 3 medi e 3 difficili. Alla categoria del 5<sup>o</sup> e 6<sup>o</sup> anno scolastico sono stati assegnati 12 quesiti, suddivisi in 4 facili, 4 medi e 4 difficili. Ogni altra categoria ha ricevuto invece 15 quesiti da risolvere, di cui 5 facili, 5 medi e 5 difficili.

Per ogni risposta corretta sono stati assegnati dei punti, mentre per ogni risposta sbagliata sono stati detratti. In caso di mancata risposta il punteggio è rimasto inalterato. Il numero di punti assegnati o detratti dipende dal grado di difficoltà del quesito:

	Facile	Medio	Difficile
Risposta corretta	6 punti	9 punti	12 punti
Risposta sbagliata	−2 punti	−3 punti	−4 punti

Il sistema internazionale utilizzato per l’assegnazione dei punti limita l’eventualità che il partecipante possa ottenere buoni risultati scegliendo le risposte in modo casuale.

Ogni partecipante ha iniziato con un punteggio pari a 45 punti (risp., Piccolo Castoro: 27 punti, 5<sup>o</sup> e 6<sup>o</sup> anno scolastico: 36 punti).

Il punteggio massimo totalizzabile era dunque pari a 180 punti (risp., Piccolo castoro: 108 punti, 5<sup>o</sup> e 6<sup>o</sup> anno scolastico: 144 punti), mentre quello minimo era di 0 punti.

In molti quesiti le risposte possibili sono state distribuite sullo schermo con una sequenza casuale. Lo stesso quesito è stato proposto in più categorie d’età.



---

**Per ulteriori informazioni:**


SVIA-SSIE-SSII Società Svizzera per l'Informatica nell'Insegnamento

Castoro Informatico

Lucio Negrini

<https://www.castoro-informatico.ch/it/kontaktieren/>

<https://www.castoro-informatico.ch/>

 <https://www.facebook.com/informatikbiberch>



# Indice

Hanno collaborato al Castoro Informatico 2019	i
Premessa	ii
Indice	iv
1. Estate, sole, lido!	1
2. Immagine da grattare	5
3. Chiosco	9
4. Beavercoins	11
5. Segnali di fumo	13
6. Il timbro	15
7. Quale torre?	19
8. Viaggiando nell'universo	21
9. Robot disegnatore	25
A. Autori dei quesiti	28
B. Sponsoring: concorso 2019	29
C. Ulteriori offerte	31



# 1. Estate, sole, lido!

È estate e la dodicenne Anita vuole andare a nuotare al lido. Prende con sé il suo fratellino Hans di sei anni. All'entrata del lido c'è questa regola:

- Età minima 8 anni; bambini sotto gli 8 anni solo se accompagnati da una persona che ha più di 10 anni.

*Chi può entrare al lido?*

- A) Anita e Hans.
- B) Anita, ma non Hans.
- C) Anita no, ma Hans sì.
- D) Né Anita, né Hans.





## Soluzione

La regola ha due significati:

1. Tutte le persone che hanno 8 anni o sono più grandi possono entrare al lido. Siccome Anita ha più di 8 anni può entrare al lido.
2. Le persone che hanno meno di 8 anni possono entrare al lido se sono accompagnate da una persona che ha più di 10 anni. Siccome Hans viene accompagnato da Anita, e Anita ha più di 10 anni, anche Hans può entrare al lido.

Quindi la risposta esatta è A) Anita e Hans.

## Questa è l'informatica!

La regola del lido stabilisce delle *condizioni*, per le quali qualcosa è permesso oppure vietato. In questo caso vengono stabilite due condizioni per le quali una persona può entrare oppure no. *Quando* la condizione è rispettata, *allora* la persona può entrare al lido. Quindi la regola può essere formulata anche in questo modo:

*Quando* la persona ha 8 anni o è più grande:

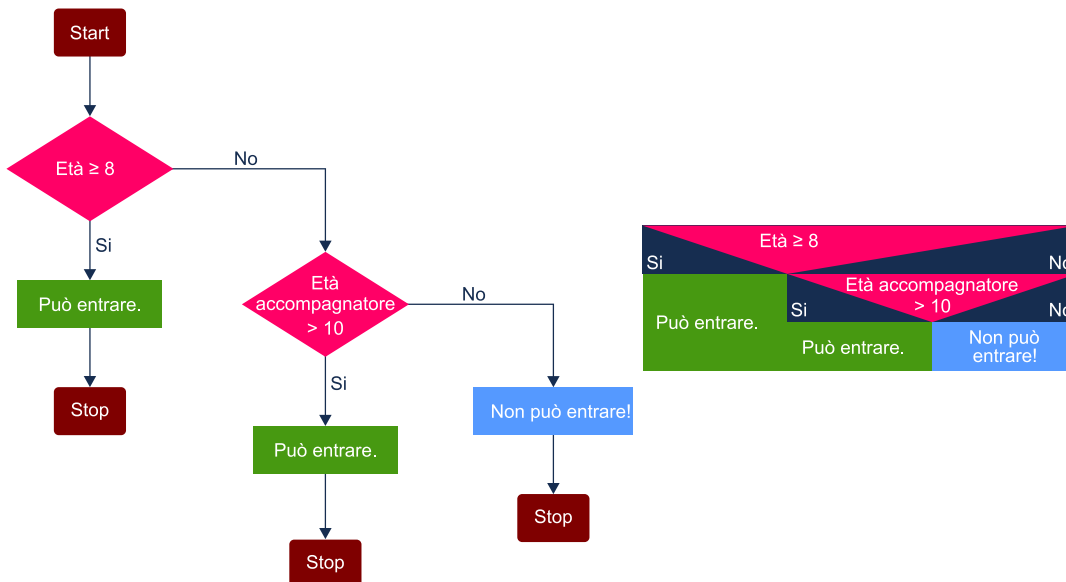
*Allora* può entrare al lido

*Se no: Quando* la persona è accompagnata da una persona che ha più di 10 anni:

*Allora* può entrare al lido

*Se no* non può entrare al lido

Il tutto si può anche rappresentare come *diagramma di flusso* e *diagramma di Nassi-Schneiderman*:



Questi processi decisionali sono chiamati *strutture di controllo alternative* in informatica. Vengono usate molto spesso in questo ambito.





## Parole chiave e siti web

Struttura di controllo alternativa, Diagramma di flusso, Diagramma di Nassi-Shneiderman

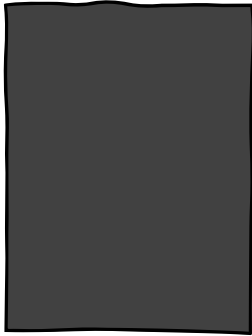
- [https://it.wikipedia.org/wiki/Struttura\\_di\\_controllo](https://it.wikipedia.org/wiki/Struttura_di_controllo)
- [https://it.wikipedia.org/wiki/Diagramma\\_di\\_flusso](https://it.wikipedia.org/wiki/Diagramma_di_flusso)
- [http://www.treccani.it/enciclopedia/diagramma-di-nassi-shneiderman\\_%28Enciclopedia-della-Matematica%29/](http://www.treccani.it/enciclopedia/diagramma-di-nassi-shneiderman_%28Enciclopedia-della-Matematica%29/)



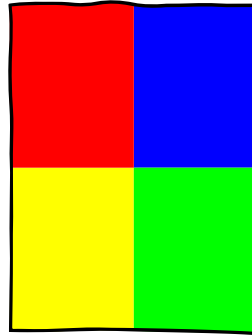


## 2. Immagine da grattare

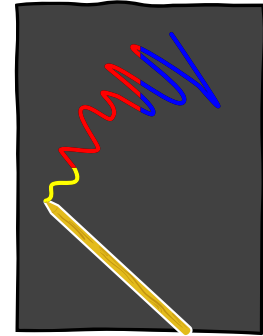
Con la carta da grattare puoi facilmente disegnare delle immagini variopinte. Con l'aiuto di un bastoncino di legno rimuovi lo strato superiore e la base colorata diventa visibile.



All'inizio la carta da grattare è ricoperta di nero ovunque.



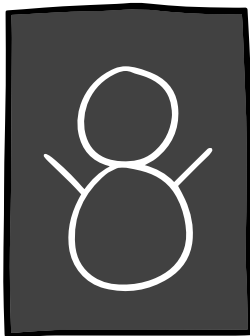
Dietro lo strato nero sono nascosti questi quattro colori.



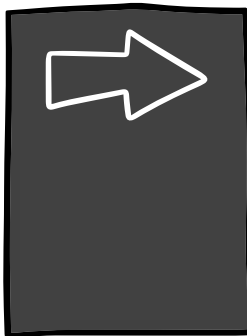
Con il bastoncino di legno, una parte dello strato nero viene grattata via. Così vedi i colori nascosti dietro.

*In quale delle quattro immagini disegnate appaiono esattamente tre colori?*

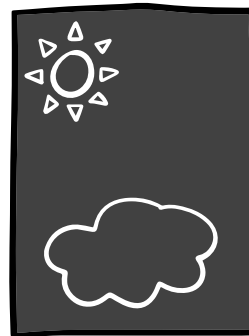
A)



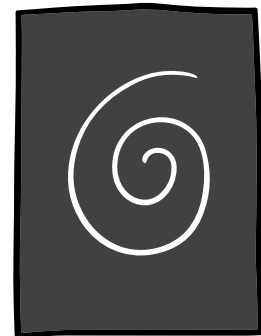
B)



C)

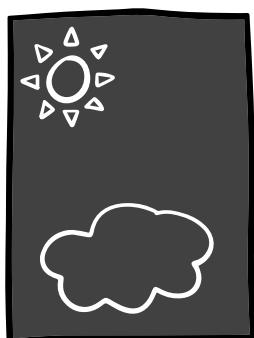


D)



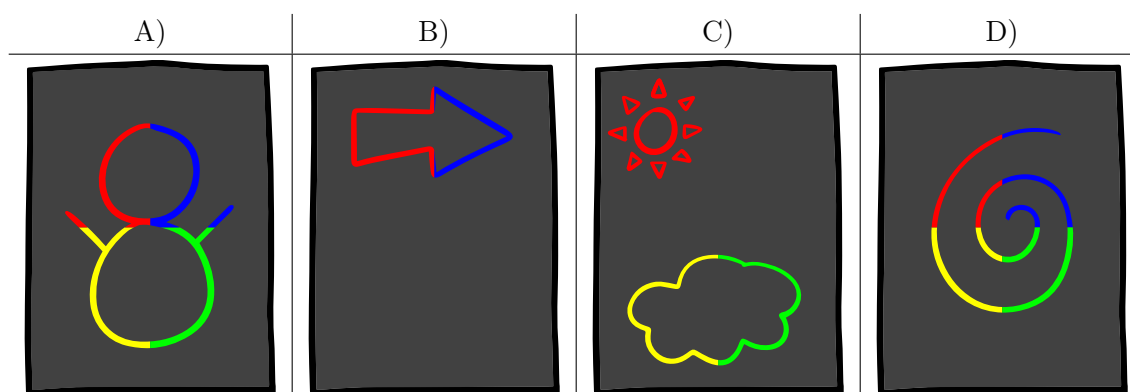


## Soluzione



La risposta corretta è C)

Quando si grattano le quattro immagini appaiono questi colori:



La risposta corretta è quindi C): appaiono i colori rosso, giallo e verde. Il quarto colore blu non appare, siccome il quarto in alto a destra rimane libero.

Nelle risposte A) e D) appaiono tutti i quattro colori, nella risposta B) solo i due colori rosso e blu.

## Questa è l'informatica!

Attraverso la rimozione dello strato superiore della carta da grattare questa diventa *trasparente*, in quel punto, si può quindi vedere attraverso il livello (a volte anche chiamato *layer*) e si vede il colore che sta dietro. In molti programmi di elaborazione di immagini vengono usati spesso dei livelli che sono trasparenti in alcuni punti. Solitamente, però, si usano al contrario: ad esempio si ha una foto come sfondo e si aggiunge un testo come nuovo livello. Questo livello, quindi, è trasparente dappertutto tranne dove c'è il testo. Naturalmente si potrebbe anche sovrascrivere direttamente l'immagine con il testo. Se però si usano più livelli, si può in seguito modificare un livello e tutti gli altri livelli restano uguali.

In questo problema bisognava immaginarsi come appare lo strato dietro quando si rende lo strato superiore trasparente. Questo è molto più facile se si divide l'immagine in immagini più piccole. Poi per ogni quarto si può considerare se lo strato superiore è trasparente da una qualche parte o no. Poi si sa, se il colore corrispondente è presente nell'immagine completa o no. Questo processo si chiama *decomposizione* e si usa spesso in informatica.

## Parole chiave e siti web

Elaborazione di immagini, Livelli (Layer), Decomposizione

- [https://en.wikipedia.org/wiki/Layers\\_\(digital\\_image\\_editing\)](https://en.wikipedia.org/wiki/Layers_(digital_image_editing))




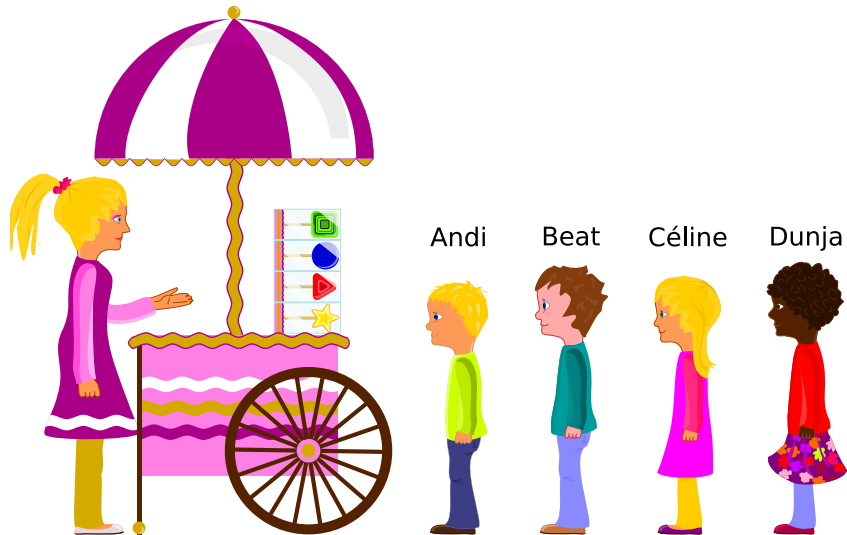
- [https://en.wikipedia.org/wiki/Decomposition\\_\(computer\\_science\)](https://en.wikipedia.org/wiki/Decomposition_(computer_science))






### 3. Chiosco

Andi, Beat, Céline e Dunja sono in coda al chiosco. La venditrice ha una pila di dolciumi davanti a sé. Vende sempre il dolciame della pila più in alto. Andi riceve il dolciame quadrato verde , siccome è il primo della coda e riceve quindi il dolciame più in alto.



Chi riceve il dolciame triangolare rosso ?

- A) Andi
- B) Beat
- C) Céline
- D) Dunja



## Soluzione

Andi è il primo: come già descritto nel compito riceve il dolcime quadrato verde più in alto 🟩. Dopo che il dolcime quadrato verde 🟩 è stato venduto, è il dolcime rotondo blu il più in alto 🟦. Questo lo riceve Beat siccome è il secondo in coda.

Dopo il dolcime rotondo blu 🟦 è il dolcime triangolare rosso 🟥 quello più in alto. Questo lo riceve Céline siccome è la terza in coda. Quindi la risposta corretta è la C) Céline.

Anche Dunja non se ne va a mani vuote: per lei rimane il dolcime a forma di stella giallo 🟡.

## Questa è l'informatica!

Andi, Beat, Céline e Dunja aspettano in una *coda*. Se Eddie vuole mettersi in coda, deve mettersi *dietro* dopo Dunja. Come primo *davanti* alla coda c'è Andi.

I dolciumi, però, sono messi in una *pila*. Quando la venditrice vuole vendere un altro dolcime lo prende *sopra* la pila. Lei vende sempre il dolcime che prende *da sopra* la pila.

Una coda di attesa (ingl. *queue*) può aggiungere elementi dal dietro (ingl. *enqueue*) e può togliere elementi dal davanti (ingl. *dequeue*). Funziona quindi secondo il principio per il quale il primo elemento che viene aggiunto è il primo che viene tolto (ingl. “*First-In-First-Out*”, in breve “*FIFO*”).

Una *pila* (ingl. *stack*) può aggiungere elementi sopra (ingl. *push*) e toglierli da sopra (ingl. *pop*). Funziona quindi secondo il principio per il quale l'ultimo elemento aggiunto è il primo a essere tolto (ingl. “*Last-In-First-Out*”, in breve “*LIFO*”). Spesso le code e le pile offrono delle funzioni aggiuntive, in modo da poter mostrare il prossimo elemento da togliere (ingl. *front* rispettivamente *top*), oppure per poter chiedere se la coda o la pila sono vuote (ingl. *empty*).

Entrambe queste strutture vengono usate spesso nei computer, siccome sono molto facili. Le pile per esempio sono state usate già nel 1945 da Konrad Zuse, in uno dei primi computer al mondo.

## Parole chiave e siti web

Coda, Pila

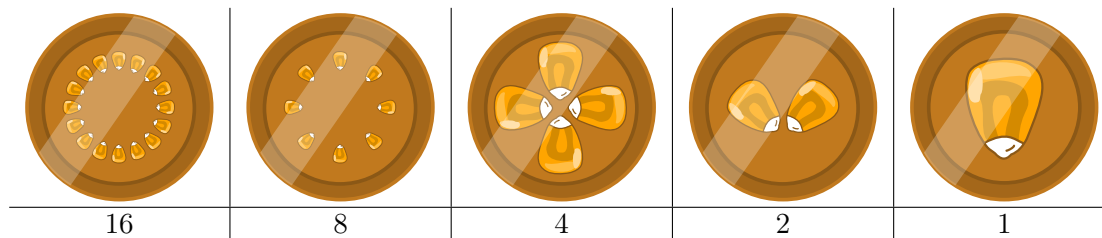
- [https://it.wikipedia.org/wiki/Coda\\_\(informatica\)](https://it.wikipedia.org/wiki/Coda_(informatica))
- [https://it.wikipedia.org/wiki/Pila\\_\(informatica\)](https://it.wikipedia.org/wiki/Pila_(informatica))
- [https://it.wikipedia.org/wiki/Z4\\_\(computer\)](https://it.wikipedia.org/wiki/Z4_(computer))





## 4. Beavercoins




Nel paese dei castori si usano le “Beavercoins” come valuta. Le monete hanno i seguenti valori:



I castori non si portano dietro volentieri molte monete e quindi pagano con meno monete possibili. *Con quali monete pagheresti 13 Beavercoins usando meno monete possibili?*



## Soluzione

La migliore soluzione, e quindi quella corretta, è di pagare con  ,  e  , quindi con una moneta da 8-Beavercoins, una moneta da 4-Beavercoins e una moneta da 1-Beavercoins. La somma delle monete corrisponde a  $8 + 4 + 1 = 13$ . Con meno monete non è possibile, perché una moneta più grossa della moneta da 8-Beavercoins sarebbe già quella da 16-Beavercoins e non c'è nessuna moneta del valore dei mancanti 5 Beavercoins. La seconda moneta più piccola è quella da 4-Beavercoins, quindi si ha bisogno, assieme alla moneta da 1-Beavercoins, proprio di queste tre monete.

Per trovare la soluzione corretta si può anche iniziare con un'altra combinazione, ad esempio con due monete da 4-Beavercoins, una da 2-Beavercoins e tre da 1-Beavercoins. Successivamente si possono sostituire due monete dello stesso valore con una moneta di valore doppio fino a quando si raggiunge il risultato corretto.

## Questa è l'informatica!

Informatiche e informatici sono esperti nel rappresentare informazioni come sequenze di simboli. Ciò include anche la rappresentazione dei numeri. In questo problema si tratta di una somma di denaro che può essere pagata con diverse combinazioni di monete. Questa combinazione non è unica, diverse combinazioni con monete di diverso valore danno la stessa somma di denaro. Quindi in questo problema si tratta anche di trovare la combinazione con il minor numero di monete.

In questo problema le monete sono scelte in modo che il valore di due monete con lo stesso valore assieme corrisponde sempre a quello della prossima moneta più grossa. Il risultato è il *sistema binario* con i valori 1, 2, 4, 8, 16 eccetera. Nel sistema numerico binario la rappresentazione di qualsiasi numero come 13 è sempre unica: un valore è utilizzato oppure no.

Anche l'abaco, una macchina per calcolare che si usava molte centinaia di anni fa e che viene usata ancora oggi nell'epoca delle calcolatrici in alcune regioni del mondo, funziona in modo simile.

## Parole chiave e siti web





Sistema numerico binari, Abaco

- [https://it.wikipedia.org/wiki/Sistema\\_numerico\\_binario](https://it.wikipedia.org/wiki/Sistema_numerico_binario)
- <https://it.wikipedia.org/wiki/Abaco>



## 5. Segnali di fumo

Un castoro si siede sempre sulla montagna e osserva il tempo. Dice ai castori nella valle come sarà il tempo. Usa segnali di fumo che consistono in cinque nuvole di fumo successive. Una nuvola di fumo o è piccola o è grande. I castori hanno concordato i seguenti segnali di fumo:

			
Sarà temporalesco	Sarà piovoso	Sarà nuvoloso	Sarà soleggiato

In un giorno ventoso i castori nella valle non riescono a riconoscere bene le nuvole di fumo. Sono solo sicuri che la seconda e la quarta nuvola di fumo sono grandi, le altre le hanno sostituite con dei punti di domanda:



*Cosa potrebbe significare?*

- A) “Sarà temporalesco” o “Sarà piovoso”.
- B) “Sarà piovoso” o “Sarà nuvoloso”.
- C) “Sarà piovoso” o “Sarà soleggiato”.
- D) “Sarà temporalesco” o “Sarà nuvoloso”.



## Soluzione

I castori nella valle hanno riconosciuto alla seconda e alla quarta posizione una nuvola di fumo grande. Anche per i segnali di fumo “Sarà temporalesco” e “Sarà nuvoloso” ci sono alla seconda e alla quarta posizione delle nuvole grandi. Per “Sarà piovoso” e “Sarà soleggiato” in queste posizioni ci sono delle nuvole di fumo piccole, quindi questi segnali di fumo non coincidono con l’osservazione dei castori nella valle.

Quindi la risposta corrette è la D) “Sarà temporalesco” o “Sarà nuvoloso”.

## Questa è l’informatica!

Quando si vuole trasmettere un messaggio si vuole che arrivi correttamente al destinatario. In questo problema il messaggio viene trasmesso con l’aiuto di nuvole di fumo grandi e piccole. Nel caso generale si parla di *simboli*. È quindi sensato scegliere una sequenza di simboli in modo che il messaggio trasmesso sia comprensibile anche se viene danneggiato durante la trasmissione. Ciò si può ottenere comunicando più informazioni di quelle strettamente necessarie. Queste informazioni aggiuntive vengono chiamate *ridondanti*.

Quando si riesce a ricostruire l’informazione danneggiata con al massimo  $n$  errori, si parla di codificazione  $n$ -autoregolante. Rappresentare messaggi come sequenze di simboli in modo da potere ricostruire il messaggio, anche quando la sua rappresentazione viene danneggiata durante la trasmissione, è un compito tipico degli informatici. Ad esempio rendono possibile riprodurre correttamente musica da dei CD o video da dei DVD anche quando nella trasmissione si presentano alcuni errori. D’altronde, per questo problema sarebbero abbastanza due nuvole di fumo per trasmettere i quattro diversi messaggi:

Sarà temporalesco.	Sarà piovoso.	Sarà nuvoloso.	Sarà soleggiato.

I castori, però, usano cinque nuvole di fumo. Ciò permette loro di capire i messaggi correttamente anche in casi dove due nuvole di fumo, o addirittura in certi casi tre nuvole di fumo, sono “illeggibili”. D’altronde, i castori hanno pensato ai messaggi in modo tale che ogni due messaggi si differenziano in almeno tre posizioni.

## Parole chiave e siti web

Rilevazione e correzione d’errore

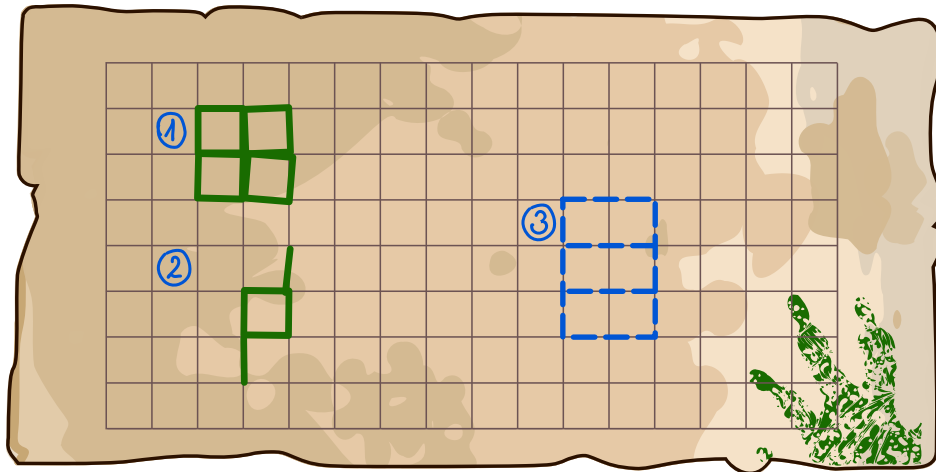
- [https://it.wikipedia.org/wiki/Rilevazione\\_e\\_correzione\\_d'errore](https://it.wikipedia.org/wiki/Rilevazione_e_correzione_d'errore)



## 6. Il timbro

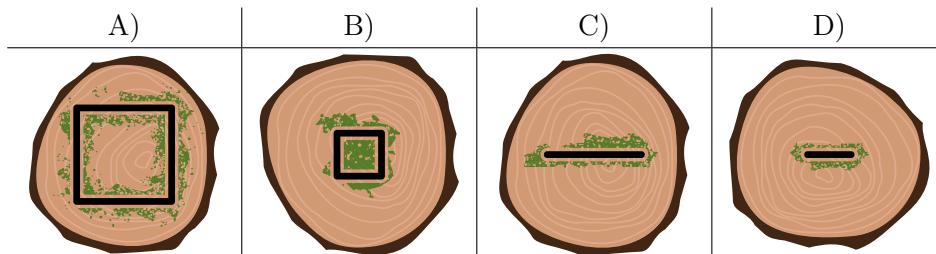
Il castoro Paul ha quattro timbri A, B, C e D, come mostrato sotto. Con questi timbri Paul ha timbrato le due figure ① e ②.

- Per la figura ① Paul ha usato quattro volte il timbro B.
- Per la figura ② Paul ha usato una volta il timbro B e due volte il timbro D.



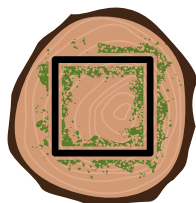
Adesso Paul vuole avere la figura ③. Maria, la sorella di Paul, afferma che lei per creare la figura deve solo timbrare due volte.

*Che timbro utilizzerebbe Maria?*

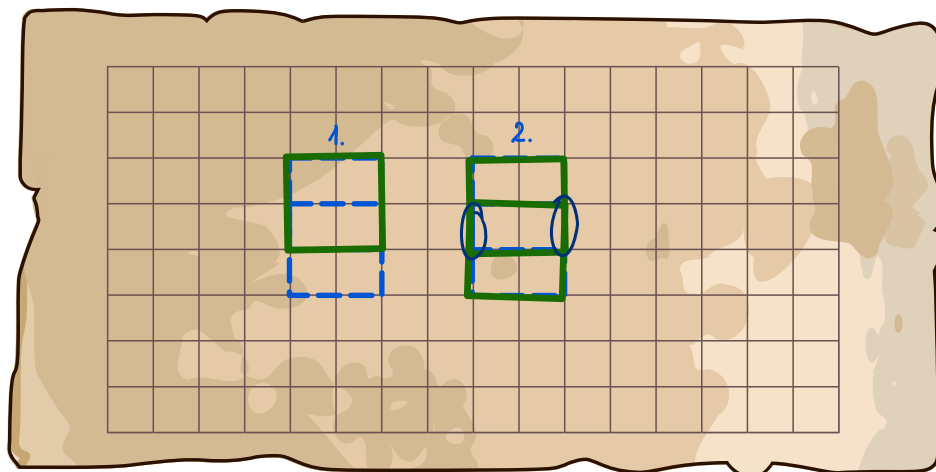




## Soluzione



La risposta corretta è A) . Se Maria ha timbrato un quadrato grande e sposta il timbro di un quadretto verso l'alto o verso il basso si forma esattamente la figura desiderata. Due parti delle linee si sovrappongono, ma se lei timbra in modo preciso non si vede:



Con gli altri timbri non si può ottenere la figura desiderata:

- Con il timbro B è impossibile timbrare un rettangolo largo due quadretti senza linea centrale.
- Con il timbro C potrebbe sicuramente timbrare la figura, ma poiché la figura ha linee della lunghezza totale di quattordici quadretti e timbrando una volta sola si copre solo la lunghezza di due quadretti, ha bisogno di timbrare almeno sette volte. Se si guarda attentamente si può constatare che ha perfino bisogno di timbrare otto volte, perché per riuscire a timbrare entrambi i tratti verticali ha bisogno di timbrare due volte (con sovrapposizioni) in aggiunta alle tre linee orizzontali.
- Con il timbro D potrebbe sicuramente timbrare la figura, ma poiché la figura ha linee della lunghezza totale di quattordici quadretti e timbrando una volta sola può timbrare solo la lunghezza di un quadretto, ha quindi bisogno di timbrare quattordici volte.

## Questa è l'informatica!

Per molti problemi ci sono molte soluzioni diverse che portano tutte all'obiettivo. Spesso alcune sono veloci da trovare, come per esempio una soluzione con l'aiuto del timbro C o D. Ma non tutte le soluzioni sono "buone" allo stesso modo: le diverse soluzioni si differenziano, ad esempio, nel numero di volte che bisogna timbrare.

Uno dei compiti dell'informatica è trovare tra molte soluzioni di un problema quella "migliore". Questo è molto importante nella pratica: quando si può risolvere un problema in un'ora al posto che in un giorno, restano molte ore per potersi occupare di altri problemi.

Per misurare l'efficienza gli informatici analizzano i processi e descrivono la loro durata in base alla quantità o alla grandezza dei dati da elaborare. Quando ad esempio si cerca una voce in un array



ordinato con 1'000'000 di voci o si può guardare array per array e fare in media 500'000 confronti, oppure si può iniziare in mezzo e continuare a cercare sempre nella rispettiva metà ...e dopo al massimo 20 confronti si è trovato l'elemento! Per 3 secondi a confronto questa sarebbe una differenza tra circa 17 giorni di ricerca ininterrotta o 1 minuto.

## Parole chiave e siti web

Efficienza, Teoria della complessità computazionale

- [https://it.wikipedia.org/wiki/Efficienza\\_\(informatica\)](https://it.wikipedia.org/wiki/Efficienza_(informatica))
- [https://it.wikipedia.org/wiki/Teoria\\_della\\_complessit%C3%A0\\_computazionale](https://it.wikipedia.org/wiki/Teoria_della_complessit%C3%A0_computazionale)



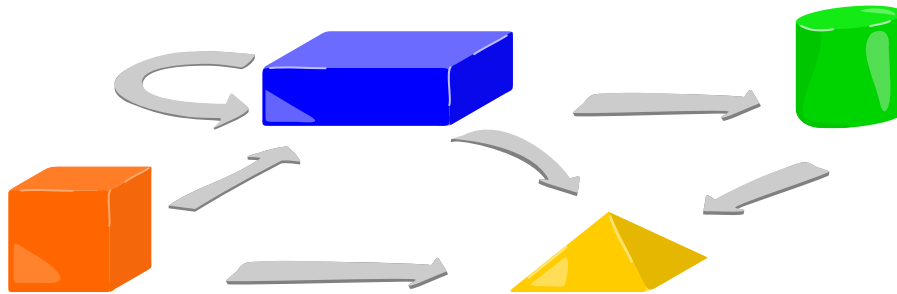




## 7. Quale torre?

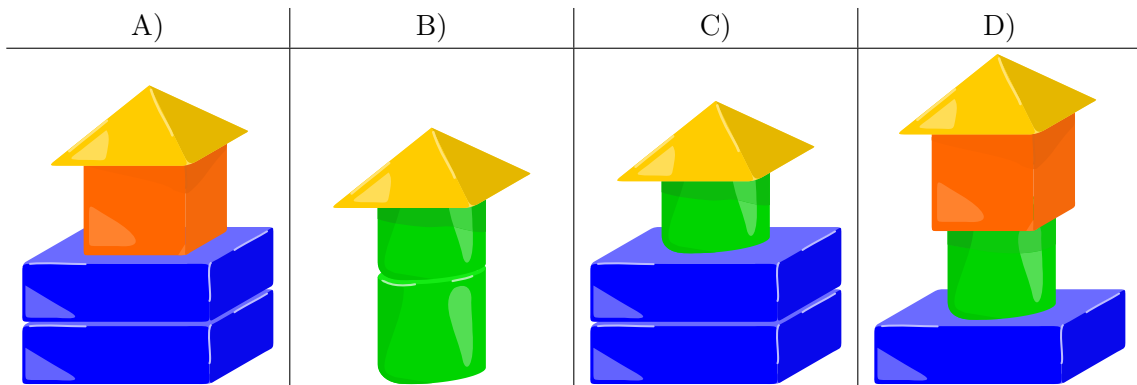
La sorellina piccola di Leon ha formulato delle regole su come i blocchetti del gioco delle costruzioni possono essere impilati. Queste regole sono state illustrate con delle frecce in un disegno. Inoltre vale:

- Leon può cominciare con qualsiasi blocchetto.
- Leon deve sempre seguire le frecce. Se da un blocchetto partono molte frecce lui ne può scegliere una. Quando una freccia indica indietro verso lo stesso blocchetto può impilarne un altro di quel tipo sulla torre.
- Leon deve smettere quando da un blocchetto appena impilato non parte più nessuna freccia.



Leon impila quattro torri diverse per la sua sorellina piccola.

Quale delle quattro torri ha costruito secondo le regole della sua sorellina piccola?





## Soluzione

La torre della risposta A) inizia correttamente con due parallelepipedi blu. Dopo il secondo parallelepipedo blu segue tuttavia un cubo arancione, ma non c'è nessuna freccia dal parallelepipedo blu verso il cubo arancione. Quindi la risposta A) è sbagliata.

La torre della risposta B) inizia correttamente con un cilindro verde. Dopo il cilindro verde segue tuttavia un ulteriore cilindro verde, ma non c'è nessuna freccia dal cilindro verde indietro verso il cilindro verde. Quindi la risposta B) è sbagliata.

La torre della risposta C) inizia correttamente con due parallelepipedi blu. Dopo il secondo parallelepipedo blu segue correttamente un cilindro verde e dopo il cilindro verde segue correttamente una piramide gialla. Siccome non parte nessuna freccia dalla piramide gialla è corretto che non venga impilato sopra nessun altro blocchetto. Quindi la risposta C) è giusta.

La torre della risposta D) inizia correttamente con un parallelepipedo blu. Dopo il parallelepipedo blu segue correttamente un cilindro verde. Dopo il cilindro verde segue tuttavia un cubo arancione, ma non c'è nessuna freccia dal cilindro verde al cubo arancione. Quindi la risposta D) è sbagliata.

## Questa è l'informatica!

Le regole per la costruzione di una torre si basano sul fatto che il blocchetto più in alto della torre decide sempre quali blocchetti sono permessi come prossimi blocchetti. Quindi il blocchetto più in alto della torre è lo *stato attuale* della torre. Le regole stabiliscono in quale prossimo stato può *trasformarsi* la torre. Il grafico con le frecce è quindi un *diagramma di stato* o anche *diagramma di transizione di stato*. Siccome tutti i blocchetti possono essere usati come blocchetti più in basso, tutti i blocchetti sono possibili *stati iniziali*. La piramide gialla è l'unico blocchetto che è uno *stato finale*, con il quale la torre viene terminata (se prima non è caduta). La decisione di impilare un altro blocchetto sulla torre è un *input* del costruttore.

Questi aspetti della costruzione di una torre descrivono un cosiddetto *automa a stati finiti non deterministico*. Si chiama non deterministico perché ci sono stati nei quali possono essere scelte diverse prossime strade: dopo un parallelepipedo blu possono seguire un ulteriore parallelepipedo blu, un cilindro verde o una piramide gialla. Si chiama finito perché c'è solo un insieme finito di possibili stati: uno dei quattro blocchetti come blocchetto superiore della torre. Teoreticamente, però, con esso si potrebbe costruire una torre infinita ... ma per farlo ci sarebbe bisogno di infiniti parallelepipedi blu e le torri alte a volte cadono (spesso per la grande gioia dei costruttori).

Il modello dell'automa a stati finiti non deterministico viene usato spesso in informatica. Si adatta bene alla descrizione di cose molto diverse: il comportamento di moduli di software o anche interi programmi, facili strutture linguistiche, l'interazione di componenti di un hardware e molto altro ancora. Con l'aiuto di una tale descrizione formale può anche essere testato se il software funziona come desiderato ... o se la torre è costruita in modo giusto.

## Parole chiave e siti web

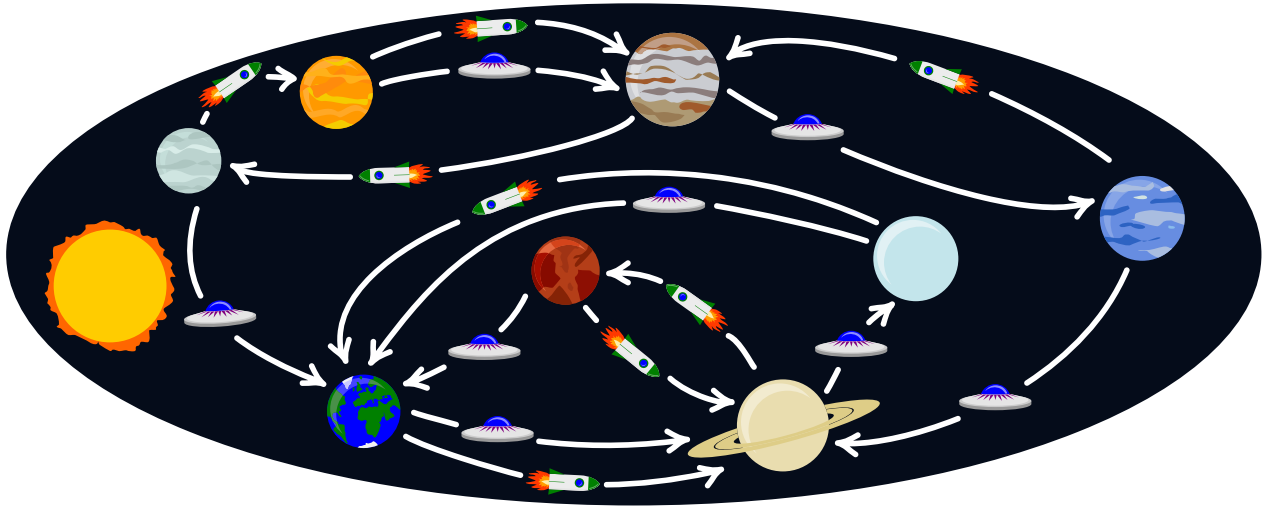
Automa a stati finiti non deterministico

- [https://it.wikipedia.org/wiki/Automa\\_a\\_stati\\_finiti\\_non\\_deterministico](https://it.wikipedia.org/wiki/Automa_a_stati_finiti_non_deterministico)
- <https://www.swisseduc.ch/informatik/karatojava/kara/index.html>
- <https://educ.ethz.ch/unterrichtsmaterialien/informatik/kara.html>



## 8. Viaggiando nell'universo

Gli astronauti possono volare tra i pianeti del nostro sistema solare con dei razzi 🚀 o degli UFO 🛸. La carta seguente raffigura le possibili rotte aeree:



Un astronauta che vuole viaggiare da Venere 🟠 a Saturno 🪐 può volare con un razzo 🚀 o con un UFO 🛸 fino a Giove 🪛. Poi può volare con un UFO 🛸 fino a Nettuno 🌌 e alla fine con un UFO 🛸 fino al suo pianeta di destinazione Saturno 🪐. Se l'astronauta vola prima con un razzo e poi con due UFO, si descrive il viaggio in questo modo:
















L'astronauta Heidi al momento è sul pianeta Nettuno 🌌 e vuole tornare sulla Terra 🌍. L'agenzia viaggi dello spazio le invia quattro proposte.

Quale delle quattro proposte non porta Heidi sulla Terra 🌍?








- A) 🛸 🛸 🚀
- B) 🚀 🛸 🚀 🛸
- C) 🚀 🛸 🛸 🛸 🚀
- D) 🚀 🚀 🚀 🛸














## Soluzione

La risposta B)     è l'unica sbagliata. Se Heidi segue questa proposta atterra alla fine ancora su Nettuno . Infatti, prima vola con un razzo  fino a Giove , poi con un UFO  indietro fino a Nettuno , poi ancora con un razzo  fino a Giove  alla fine ancora con un UFO  indietro fino a Nettuno .



Le altre tre proposte portano tutte indietro fino alla Terra .

Risposta A): Da Nettuno  con un UFO  fino a Saturno , con un UFO  fino a Urano  e con un razzo  fino alla Terra .

Risposta C) Da Nettuno  con un razzo  fino a Giove , con un UFO  fino a Nettuno , con un UFO  fino a Saturno , con un UFO  fino a Urano  e con un razzo  fino alla Terra .

Risposta D): Da Nettuno  con un razzo  fino a Giove , con un razzo  fino a Mercurio  e con un UFO  fino alla Terra .

## Questa è l'informatica!

La carta delle possibili rotte aeree da pianeta a pianeta ha una proprietà particolare: da ogni pianeta partono esattamente due rotte aeree, una con un razzo  e una con un UFO . Quindi è sempre chiaro su quale pianeta si vola quando viene indicato con quale mezzo di trasporto si vola.

Una carta di questo tipo descrive un *automa a stati finiti deterministico*. Un automa di questo tipo consiste in un insieme di possibili *stati* (in questo caso sono i nomi dei pianeti come posizione dell'astronauta), un insieme di possibili *transizioni di stato* (in questo caso sono le frecce sulla carta con il cui aiuto l'astronauta può cambiare la sua posizione) basati su determinati *input* ("razzo" o "UFO") come anche uno *stato iniziale* (in questo caso "Nettuno") e un insieme di *stati finiti* (in questo caso solo lo stato "Terra"). La carta si chiama anche un *diagramma di stato* o un *diagramma di transizione di stato*.

Gli automi a stati finiti deterministici vengono usati in diversi ambiti perché sono molto facili da programmare. Esempi tipici sono macchine del caffè, lavastoviglie o anche distributori di bevande. Ma vengono anche utilizzati per riconoscere delle parole correttamente (ad esempio se un testo inserito rappresenta un indirizzo e-mail). Gli automi a stati finiti deterministici si possono anche associare con una determinata classe di grammatiche (cosiddette *grammatiche regolari*) e una determinata classe di linguaggi artificiali (cosiddetti *linguaggi regolari*) e saltare da un "mondo" all'altro. Questo aiuta a risolvere molti problemi.

D'altronde l'agenzia viaggi dello spazio ha un altro compito: deve trovare nel diagramma di stato un possibile percorso da uno stato all'altro. Per questo aiuta vedere il diagramma di stato come *grafo orientato* e cercare un possibile percorso da un *nodo* a un altro nodo con aiuto degli *archi* dati. Per questo ci sono degli algoritmi standard, in modo che l'agenzia viaggi dello spazio non deve iniziare a cercare sempre da capo.

## Parole chiave e siti web

Automa a stati finiti deterministico, Grafi

- [https://it.wikipedia.org/wiki/Automa\\_a\\_stati\\_finiti\\_deterministico](https://it.wikipedia.org/wiki/Automa_a_stati_finiti_deterministico)
- <https://www.swisseduc.ch/informatik/karatojava/kara/index.html>
- <https://educ.ethz.ch/unterrichtsmaterialien/informatik/kara.html>



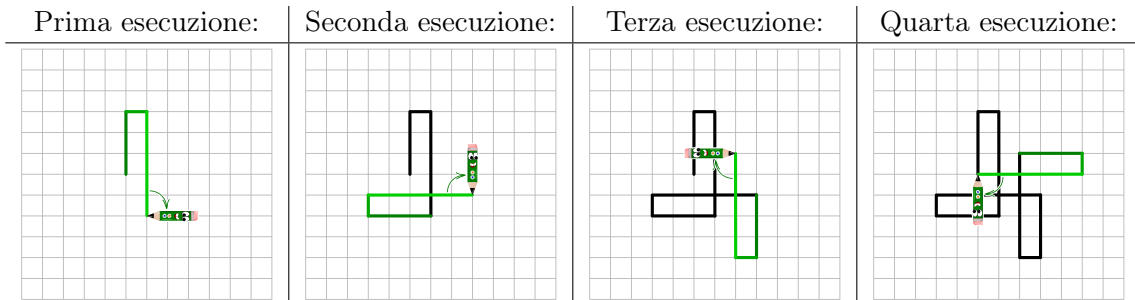
- <https://it.wikipedia.org/wiki/Grafo>





## 9. Robot disegnatore

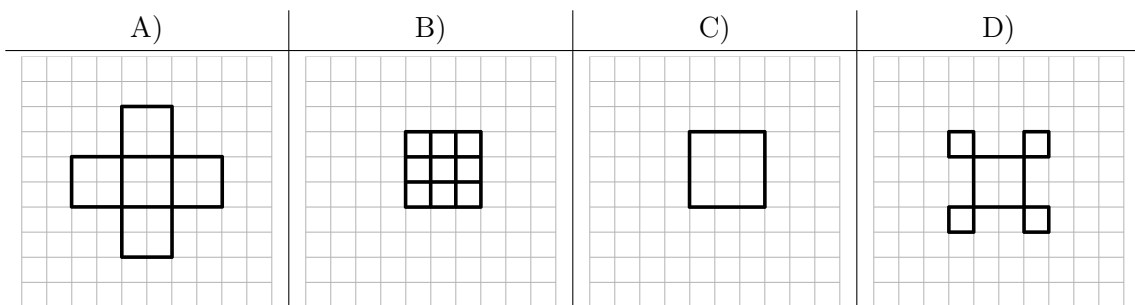
Un robot si muove su una griglia e disegna delle linee. Può essere pilotato con l'aiuto di tre numeri. Se gli si dà i numeri  $3\curvearrowright 1\curvearrowright 5\curvearrowright$  disegna la figura seguente:



In questo caso ripete i seguenti passaggi quattro volte:

- Va in avanti sulla griglia di quanti quadretti sono dati dal primo numero.
- Si ruota di un quarto di giro verso destra.
- Va in avanti sulla griglia di quanti quadretti sono dati dal secondo numero.
- Si ruota di un quarto di giro verso destra.
- Va in avanti sulla griglia di quanti quadretti sono dati dal terzo numero.
- Si ruota di un quarto di giro verso destra.

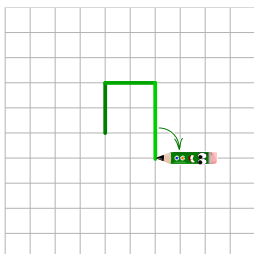
Al robot vengono dati i numeri  $2\curvearrowright 2\curvearrowright 3\curvearrowright$ . Come appaiono le linee disegnate?



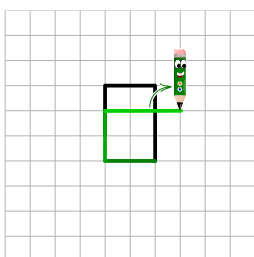


## Soluzione

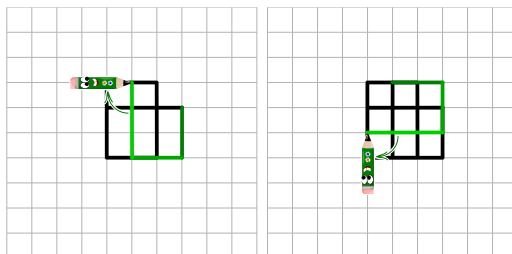
I numeri  $2\curvearrowright 2\curvearrowright 3\curvearrowright$  significano che prima il robot va in avanti di 2 quadretti, si ruota di un quarto di giro verso destra, va ancora avanti di 2 quadretti, si ruota di un quarto di giro verso destra, poi va avanti di 3 quadretti, e si ruota ancora di un quarto di giro verso destra. Quindi ha disegnato le linee seguenti:



Quando ciò viene ripetuto ha disegnato in totale le linee seguenti:

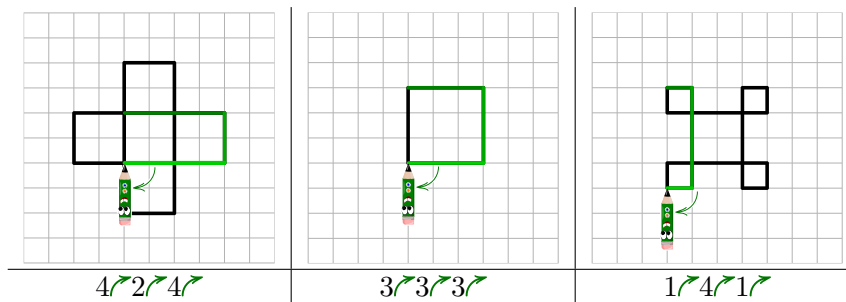


Dopo due ripetizioni l'immagine appare così:



Quindi la risposta B) è corretta.

D'altronde le altre tre figure si lasciano anche disegnare con l'aiuto del robot, necessitano semplicemente di altri numeri:







## Questa è l'informatica!

Il robot disegnatore in questo problema può eseguire soltanto programmi molto semplici. Il linguaggio di programmazione che capisce il robot, riconosce solo numeri come istruzioni. Ogni programma deve consistere in esattamente questi tre numeri seguiti dal simbolo ↗. Inoltre è incorporato in modo fisso che il robot ripete le istruzioni contenute nel programma quattro volte, sia che se lo si voglia o meno.

La maggior parte dei robot e dei computer comprende linguaggi (di programmazione) molto più complessi. La maggior parte di questi linguaggi ha le stesse caratteristiche di base:

1. I programmi possono essere costituiti da un numero qualsiasi di istruzioni che vengono eseguite una dopo l'altra come una *sequenza*.
2. Istruzioni ripetute di diverso tipo, i cosiddetti *cicli*, possono, ma non devono, essere utilizzati.
3. Inoltre, ci sono istruzioni condizionali che consentono diverse esecuzioni del programma a seconda dello stato del programma, le cosiddette *diramazioni*.

Se un linguaggio di programmazione contiene cicli e diramazioni, con il suo aiuto si può calcolare tutto ciò che è calcolabile. In informatica tali linguaggi di programmazione vengono designati come universali – o anche come *Turing equivalenti*.

D'altronde il robot in questo problema è un classico contesto nel quale si può imparare a programmare. Al posto del robot ci si può immaginare una tartaruga che disegna delle linee. Tali *grafiche della tartaruga* possono essere compilate ad esempio con XlogoOnline o anche con TigerJython.

## Parole chiave e siti web

Grafica della tartaruga

- [https://it.wikipedia.org/wiki/Programmazione\\_strutturata](https://it.wikipedia.org/wiki/Programmazione_strutturata)
- [https://it.wikipedia.org/wiki/Turing\\_equivalenza](https://it.wikipedia.org/wiki/Turing_equivalenza)
- [https://en.wikipedia.org/wiki/Turtle\\_graphics](https://en.wikipedia.org/wiki/Turtle_graphics)
- <https://xlogo.inf.ethz.ch/>
- <http://www.tigerjython.ch/>



## A. Autori dei quesiti

 Haim Averbuch  
 Michelle Barnett  
 Michael Barot  
 Daniela Bezáková  
 Anton Chukhnov  
 Allira Crowe  
 Andrew Csizmadia  
 Christian Datzko  
 Susanne Datzko  
 Marissa Engels  
 Olivier Ens  
 Martin Guggisberg  
 Vernon Gutierrez  
 Juraj Hromkovič  
 Alisher Ikramov  
 Thomas Ioannou

 Tiberiu Iorgulescu  
 Yong-ju Jeon  
 Felipe Jiménez  
 Anna Laura John  
 Mile Jovanov  
 Adem Khachnaoui  
 Injoo Kim  
 Vaidotas Kinčius  
 Jia-Ling Koh  
 Dennis Komm  
 Anja Koron  
 Bohdan Kudrenko  
 Regula Lacher  
 Karolína Mayerová  
 Anna Morpurgo  
 Tom Naughton

 Pia Niemelä  
 Henry Ong  
 Wolfgang Pohl  
 Stavroula Prantsoudi  
 Nol Premasathian  
 J.P. Pretti  
 Taras Shpot  
 Jacqueline Staub  
 Bundit Thanasopon  
 Monika Tomcsányiová  
 Peter Tomcsányi  
 Nicole Trachsler  
 Troy Vasiga  
 Ela Veza  
 Florentina Voboril  
 Khairul A. Mohamad Zaki



## B. Sponsoring: concorso 2019

**HASLERSTIFTUNG**

<http://www.haslerstiftung.ch/>

**ROBOROBO**

<http://www.robobo.ch/>

**bischof  
berger**

<http://www.baerli-biber.ch/>

**verkehrshaus.ch**

<http://www.verkehrshaus.ch/>  
Musée des transports, Lucerne



**Kanton Zürich  
Volkswirtschaftsdirektion  
Amt für Wirtschaft und Arbeit**

Standortförderung beim Amt für Wirtschaft und Arbeit  
Kanton Zürich



i-factory (Musée des transports, Lucerne)

**UBS**

<http://www.ubs.com/>

**bbv**  
Software Services

<http://www.bbv.ch/>

**PRESENTEX**  
Das Geschenk - die gute Werbung

<http://www.presentex.ch/>

**OXOCARD**

<http://www.oxocard.ch/>  
OXOcard  
OXON

**DIARTIS**

<http://www.diartis.ch/>  
Diartis AG



<https://educatec.ch/>  
educaTEC



<http://senarclens.com/>  
Senarclens Leu & Partner



AUSBILDUNGS- UND BERATUNGSZENTRUM  
FÜR INFORMATIKUNTERRICHT

<http://www.abz.inf.ethz.ch/>  
Ausbildungs- und Beratungszentrum für Informatikunterricht der ETH Zürich.



<http://www.hepl.ch/>  
Haute école pédagogique du canton de Vaud



<http://www.phlu.ch/>  
Pädagogische Hochschule Luzern



<https://www.fhnw.ch/de/die-fhnw/hochschulen/ph>  
Pädagogische Hochschule FHNW

Scuola universitaria professionale  
della Svizzera italiana



<http://www.supsi.ch/home/supsi.html>  
La Scuola universitaria professionale della Svizzera italiana (SUPSI)



<https://www.zhdk.ch/>  
Zürcher Hochschule der Künste



## C. Ulteriori offerte

010100110101011001001001  
010000010010110101010011  
010100110100100101000101  
001011010101001101010011  
01001001010010010010001

**SS!**

[www.svia-ssie-ssii.ch](http://www.svia-ssie-ssii.ch)  
schweizerischervereinfürinformatikind  
erausbildung//société suisse pour l'infor  
matique dans l'enseignement//società sviz  
zera per l'informatica nell'insegnamento

Diventate membri della SSII <http://svia-ssie-ssii.ch/verein/mitgliedschaft/> sostenendo in questo modo il Castoro Informatico.

Chi insegna presso una scuola dell'obbligo, media superiore, professionale o universitaria in Svizzera può diventare membro ordinario della SSII.

Scuole, associazioni o altre organizzazioni possono essere ammesse come membro collettivo.