

Informatica
SS1
necessità e
opportunità

Numeri, geometria, strumenti informatici: proposte laboratoriali per l'insegnamento della matematica nella secondaria di I grado

**COME INSEGNARE INFORMATICA NELLA SCUOLA DEL
PRIMO CICLO
SABATO 8 FEBBRAIO 2020**

Dipartimento di Informatica
Giovanni Degli Antoni
Università degli Studi di Milano
Via Celoria 18, Milano



Ubaldo Pernigo
www.ubimath.org

Informatica e pensiero computazionale sono trasversali



in matematica sono molte le occasioni per l'introduzione dell'informatica e l'avvio al pensiero computazionale



certificato delle competenze
(D.M. 742/2017)



esame di Stato
(D.M. 741/2017)



indicazioni nazionali e nuovi scenari
(22 febbraio 2018)



Certificato delle competenze

DM 742/2017

► Le istituzioni scolastiche devono certificare l'acquisizione delle **competenze digitali** progressivamente acquisite

- Al termine della scuola primaria
- Al termine del primo ciclo di istruzione con il superamento dell'esame di Stato

	Competenze chiave europee	Competenze dal Profilo dello studente al termine del primo ciclo di istruzione	Livello (1)
1	Comunicazione nella madrelingua o lingua di istruzione		
2	Comunicazione nelle lingue straniere		
3	Competenza matematica e competenze di base in scienza e tecnologia		
4	Competenze digitali		
5	Imparare ad imparare		
6	Competenze sociali e civiche		
7	Spirito di iniziativa*		
8	Consapevolezza ed espressione culturale		
9	L'alunno/a ha inoltre mostrato significative competenze nello svolgimento di attività scolastiche e/o extrascolastiche, relativamente a:		

Utilizza con consapevolezza e responsabilità le tecnologie per ricercare, produrre ed elaborare dati e informazioni, per interagire con altre persone, come supporto alla creatività e alla soluzione di problemi.



prova scritta
relativa alle
competenze
logico
matematiche



- 1.
La prova scritta relativa alle competenze logico matematiche accerta la capacità di rielaborazione e di organizzazione delle conoscenze, delle abilità e delle competenze acquisite dalle alunne e dagli alunni nelle seguenti aree:
 - numeri;
 - spazio e figure;
 - relazioni e funzioni;
 - dati e previsioni.
- 3.
Nella predisposizione delle tracce la commissione può fare riferimento anche ai metodi di analisi, organizzazione e rappresentazione dei dati, caratteristici del **pensiero computazionale**.

INDICAZIONI NAZIONALI E NUOVI SCENARI

22 febbraio 2018

Documento a cura del
Comitato Scientifico
Nazionale per le
Indicazioni Nazionali
per il curriculum della
scuola dell'infanzia e
del primo ciclo di
istruzione

Presentazione del
documento il
22 febbraio 2018

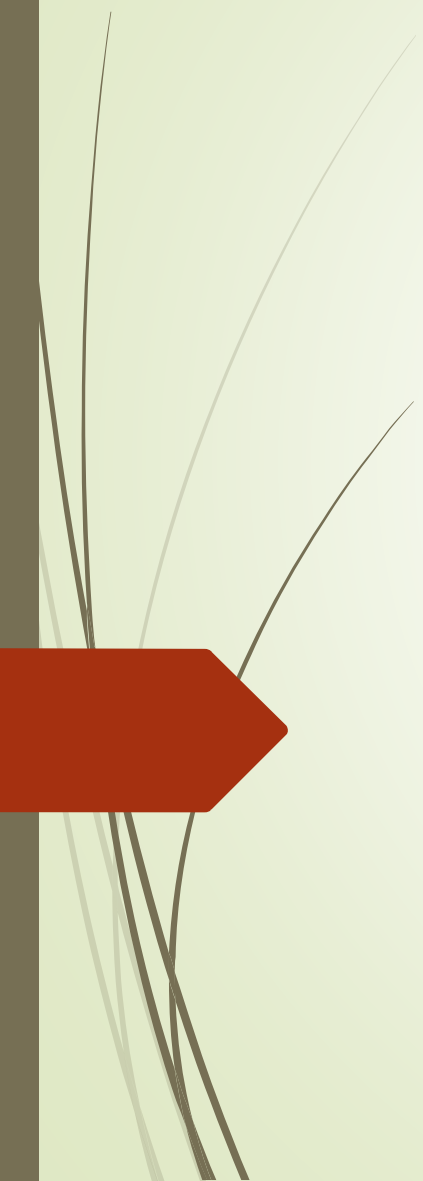


➤ 5. Gli strumenti culturali per la cittadinanza

➤ 5.3 Il pensiero matematico

➤ 5.4 Il pensiero computazionale

- Lingua e matematica, apparentate, sono alla base del **pensiero computazionale**, altro aspetto di apprendimento che le recenti normative, la legge 107/2015 e il decreto legislativo n. 62/2017 chiedono di sviluppare. Attività legate al pensiero computazionale sono previste nei Traguardi delle Indicazioni in particolare nell'ambito della Tecnologia, tuttavia se ne possono prevedere in ogni ambito del sapere.
- Per pensiero computazionale si intende un processo mentale che consente di risolvere problemi di varia natura seguendo metodi e strumenti specifici pianificando una strategia.
- È un **processo logico creativo** che, più o meno consapevolmente, viene messo in atto nella vita quotidiana per affrontare e risolvere problemi.



matematica e informatica SS1

opportunità, sperimentazione ed esperienze in classe



non mancano occasioni e opportunità

- L'insegnante di matematica ha molte occasioni, nella secondaria di primo grado, per l'introduzione sia della tecnologia sia di metodologie legate all'informatica e al pensiero computazionale
- Tenendo in mente che la tecnologia è uno strumento non un fine
- Questo risponde
 - alla necessità di sviluppare e valutare competenze in un ambito diverso da quello tradizionale disciplinari
 - alla necessità di avere elementi utili per valutare le competenze digitali

Programma il futuro

- www.programmailfuturo.it
- Si basa sul progetto code.org
- Il MIUR, in collaborazione con il CINI – Consorzio Interuniversitario Nazionale per l'Informatica, ha avviato questa iniziativa con l'obiettivo di fornire alle scuole una serie di strumenti semplici, divertenti e facilmente accessibili per formare gli studenti ai concetti di base dell'informatica.



L'Ora del Codice

[Altri corsi dell'Ora del Codice](#)

Se non hai tempo per un corso completo, prova un'esercitazione di una singola ora progettata per qualsiasi età. Unisciti a milioni di studenti e insegnanti in oltre 180 paesi del mondo iniziando con un'Ora del Codice. Trovi una descrizione completa anche sul sito di Programma il Futuro: programmailfuturo.it/come/ora-del-codice

Ballando col ... Featuring Katy Perry, Shawn Mendes, Panic! At The Disco, Lil Nas X, Jonas Brothers, Nicki Minaj, and 34 more!	Minecraft Usa la tua creatività e le competenze di problem solving per esplorare e costruire mondi sottomarini con la programmazione!	Frozen Usa la programmazione per unirti ad Anna ed Elsa mentre esplorano la magia e la bellezza del ghiaccio.	Labirinto Clas... Impara i concetti base dell'informatica. Milioni di persone l'hanno già fatto.
--	---	---	--



Proposta alle classi

- Ora del codice
- Costituzione della classe
- Attività dimostrativa
- Lavoro personale a casa
- Possibilità di monitoraggio docente
- Vi sono ancora problemi di accesso alle risorse on line da casa per alcuni allievi

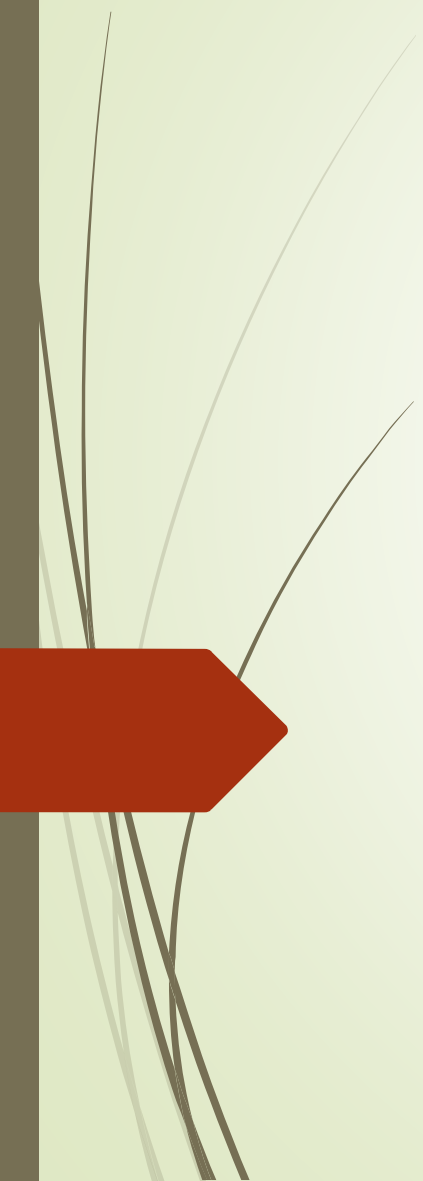
- Ora del codice
- Visione docente





In classe anche attività unplugged

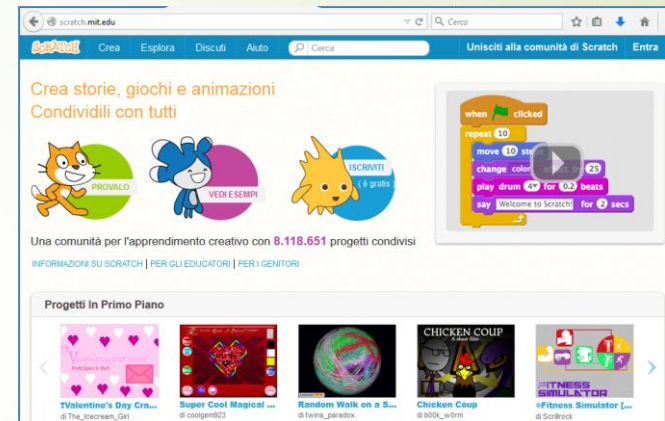
- [Kit fai da te Cody&Roby](#)
- [Computer Science Unplugged](#)
Raccolta di attività didattiche per l'insegnamento dell'Informatica utilizzando giochi, puzzle, carte, pastelli, Università di Canterbury
- [Computer Science Unplugged](#)
[Un testo per imparare l'informatica divertendosi](#)



Programmazione a blocchi

Scratch

- scratch.mit.edu
- www.scratchjr.org
- Comunità per l'apprendimento
- Puoi programmare storie interattive, giochi ed animazioni — e puoi condividere le tue creazioni.
- Aiuta i giovani ad imparare a pensare in modo creativo, a ragionare in maniera sistematica e a lavorare in maniera collaborativa — tutte queste sono capacità essenziali nel 21-esimo secolo.
- E' un progetto del Lifelong Kindergarten Group dei Media Lab del MIT. È reso disponibile in maniera completamente gratuita.
- Tutti i progetti condivisi possono essere remixati (studiati, adattati, modificati, ecc.).





Differenze

code.org è un ambiente di apprendimento e non è possibile programmare liberamente

Scratch è usabile per creare applicativi in modo libero, seppure legato a un ambiente

Possibilità di accedere con Scratch a una community e di fare il remix

Esiste una versione off-line utile per le aule didattiche e il lavoro a casa.



Esempi e schede di lavoro

- Ambiente di lavoro Scratch
- Ogni attività ha una sua scheda alunno per il lavoro di classe e la ripetizione a casa, con possibilità di inviare il lavoro al docente anche per correzione e suggerimenti
- ESEMPI
 - Numero primo o composto
 - Disegno di un poligono

Dato un numero stabilire se sia o meno primo

La ricerca dei divisori di un numero può essere fatta dividendo un numero dato per i numeri naturali da 1 a n . Per trovare i divisori di 12 serve quindi verificare quali valori da 1 a 12 usati come divisori con 12 come dividendo diano resto 0.

12:1=12 r=0	12:2=6 r=0	12:3=4 r=0	12:4=3 r=0	12:5=2 r=2	12:6=2 r=0
12:7=1 r=5	12:8=1 r=4	12:9=1 r=3	12:10=1 r=2	12:11=1 r=1	12:12=1 r=0

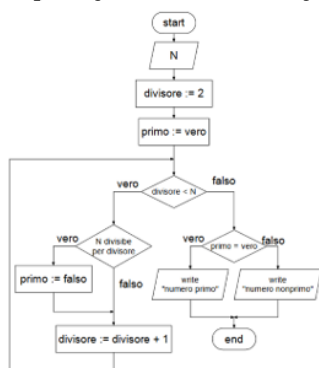
$D(12) = \{1, 2, 3, 4, 6, 12\}$

Dall'analisi delle operazioni si vede come considerando i divisori di 12 a un certo punto divisori e quozienti si invertono. A quel punto non sarebbe necessario proseguire.

Il limite massimo dei divisori di un numero n è dato, infatti, dal valore di $n/2$. Non è infatti necessario provare se tutti i numeri da 2 a n se sono divisori del numero ma è sufficiente controllare i numeri da 2 a $n/2$ siano o meno divisori del numero dato.

E' nello stesso modo sufficiente fare la verifica da 2 a $\sqrt{n}$. Per ogni divisore di n , minore della sua radice quadrata, ce n'è uno maggiore di essa e i due sono strettamente collegati ed è inutile ricercare divisori di n maggiori della sua radice quadrata.

L'algoritmo per valutare se un numero è primo si può rappresentare nel modo seguente.



```
start
read N
divisore := 2
primo := vero
while (divisore < N)
  if (N divisibile per divisore) then
    primo := falso
  end if
  divisore := divisore + 1
end while
if (primo = vero) then
  write "numero primo"
else
  write "numero non primo"
end if
end
```

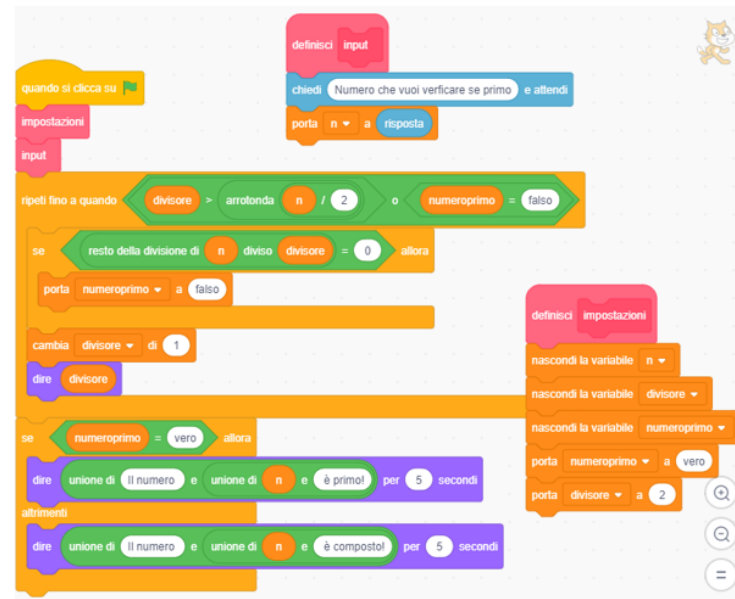
<http://www.di.unito.it/~rossano/DIDATTICA/MDAMS-0405/mdams-17.pdf>

Codice Python

```
n=int(raw_input("Inserisci un numero naturale n maggiore di 1: "))
primo=True # ipotizziamo che N sia primo
for i in range(2,n):
    if n%i==0:
        primo=False
    if primo: # se N è primo
        print n," il numero è primo"
else:
    print n," il numero è composto"
```



È possibile utilizzare un linguaggio di programmazione a blocchi anche per questo scopo e realizzare una procedura con cui allenarsi o con cui trovare i divisori di un numero qualsiasi. L'algoritmo e il relativo diagramma di flusso è traducibile nella seguente codifica a blocchi.



Nella prima si usano due blocchi personalizzati, uno per le impostazioni e uno per l'acquisizione (input) del numero di cui verificare la primalità.

METTITI ALLA PROVA

1.

Per numeri grandi il sistema sta fermo a pensare senza dare alcun cenno all'utente che è al lavoro. Inserisci la visualizzazione del divisore di cui sta eseguendo il test in modo che si veda a video che i lavori proseguono.

2.

Introduci un controllo per l'inserimento di un valore 0 e 1, da considerare con accettabili. Inserisci i blocchi [Azzera cronometro] e [Cronometro] per visualizzare il tempo di esecuzione con numeri via via crescenti (100, 1000, 10000, ...) e annotati.

3.

Inserisci il controllo da 2 a $\sqrt{n}$ e verifica che ottieni gli stessi risultati.

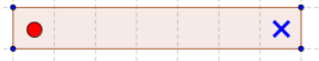
Modifica il codice per ottenere di un numero primo (ha solo i divisori banali) o composto (ha più di due divisori) i suoi divisori. Ricorda che 1 non è primo o composto.



Ciclo definito

Quando è necessario eseguire in una sola volta la valutazione di una condizione e l'incremento di un indice per eseguire delle istruzioni ripetute il ciclo *ripeti n volte* (*for*) è l'ideale. Questo tipo di iterazione è di tipo **definito** e consente di eseguire una porzione di codice un certo numero di volte.

Vediamo un esempio esplicativo utilizzando un labirinto. Per muovere il punto rosso dalla casella in cui si trova a quella indicata dalla croce blu serve spostarsi di 6 passi in avanti. La stessa azione, muoversi in avanti di una casella, viene, quindi, ripetuta per 6 volte. Utilizzando una iterazione si ottimizza questa operazione ripetitiva.

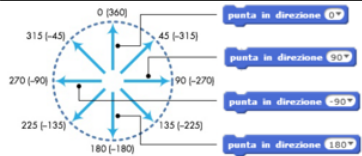


Ciclo definito	Blocchi
Costruisci ora un quadrato usando il blocco ripeti n volte . Non sono tutti strettamente necessari ma possono risultare utili. Costruisci il programma e copia qui a lato la versione funzionante.	quando si clicca su penna su penna giù pulisci punta in direzione 90 fai 100 passi ruota di 90 gradi vai a x: 0 y: 0 ripeti volte

Un quadrato usando un linguaggio di programmazione a blocchi

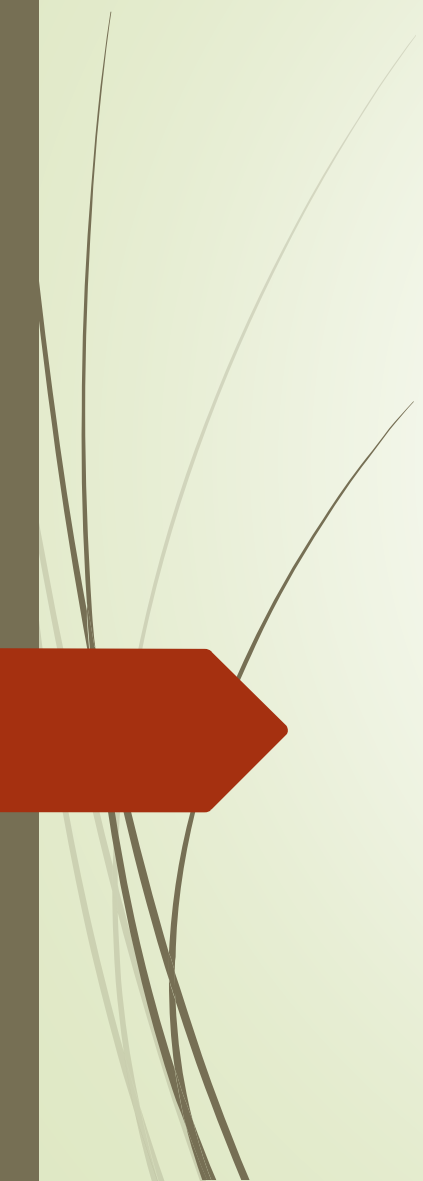
Il quadrato è un **quadrilatero regolare** con quattro lati e quattro angoli, tutti retti, congruenti. Avendo i lati congruenti, è un caso particolare di rombo, avendo i quattro angoli congruenti e retti è un caso particolare di rettangolo e, avendo i lati a due a due paralleli e congruenti, è un caso particolare di parallelogramma.

Modo sequenziale	Blocchi
Costruisci un quadrato usando più blocchi in sequenza tra quelli indicati a destra. Non sono tutti strettamente necessari ma possono risultare utili. Costruisci il programma e copia qui a lato la versione funzionante.	quando si clicca su penna su penna giù pulisci punta in direzione 90 fai 100 passi ruota di 90 gradi vai a x: 0 y: 0 NB Spazio di lavoro.



Convenzione per indicare in maniera assoluta i parametri di rotazione.





Programmare con la geometria dinamica

GeoGebra

- Uso per la geometria dinamica fin dalla classe prima
- Si possono realizzare vere e proprie righe di codice e ottenere risultati interessanti
- Utilizzo schede ad hoc per gli allievi
- Ogni attività ha una sua scheda alunno per il lavoro di classe e la ripetizione a casa, con possibilità di inviare il lavoro al docente anche per correzione e suggerimenti

➤ ESEMPIO

- Le diagonali di un poligono

Le diagonali dei poligoni - Scheda di lavoro propedeutica

Il segmento che unisce due vertici non consecutivi di un poligono si chiama **diagonale**.

Traccia in ogni poligono le diagonali e completa la tabella che segue.



Numero di lati	Nome poligono	Numero di vertici	Numero di diagonali
3	Triangolo		
4	Quadrilatero		
5			
6			
7			
8			
9			
10			
11			
12			
20	Icosagono		
1000	Chilagono		

Proponi una formula per trovare il numero totale delle diagonali y di un poligono di $x > 2$ vertici.



Calcolo delle diagonali di un poligono qualsiasi.

Seleziona lo strumento per un nuovo punto (•) e individua sul piano due punti A e B. Nascondi le relative etichette.

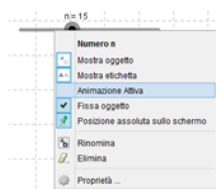
Inserisci uno slider (⇄) che sia riferito a un numero compreso in un intervallo da 0 a 30 unità, incremento 1, e denominato con la lettera latina minuscola *n*. A video compare lo strumento slider ($n \in \mathbb{N}, 0 \leq n \leq 30$). Spostando il cursore cambia il valore associato alla variabile che porta il nome dello slider.

Associeremo a questa variabile numerica al numero di lati di un poligono regolare.

Seleziona il comando Poligono regolare (•). Indica come lato il segmento dato dai due punti A e B e come numero di lati il valore associato alla variabile numerica *n*.

Attribuendo alla variabile *n*, associata allo slider, l'opzione Animazione attiva si ottiene in successione la creazione dei poligoni regolari fino al valore massimo fissato per *n* (nel nostro caso con $n = 30$ si ottiene un poligono regolare di 30 lati e nessun poligono per $n = 0$, $n = 1$ e $n = 2$).

A questo punto è possibile provare la realizzazione per i soli poligoni.



A questo punto serve disegnare le diagonali del poligono (poli1), creando le diagonali come segmento (Segmento[Punto, Punto]) a partire da ogni vertice (Vertice[Poligono, Numero n]) da 1 a $n - 1$ (ciclo for (ripeti) che è indicato con Variabile *i*, Numero *a*, Numero *b*).

Per disegnare le diagonali utilizzo una lista di segmenti definiti dalle coordinate degli estremi. La lista è data dai segmenti che rappresentano le diagonali e che si dipartono da ogni vertice verso tutti gli altri. La lista viene creata tramite due comandi Successione[] nidificati. Il comando Successione[Espressione, Variabile *i*, Numero *a*, Numero *b*] crea una lista di oggetti utilizzando una espressione indicata dall'utente e un indice variabile (*i*) tra due numeri (*a* e *b*).

Dobbiamo, quindi, disegnare i segmenti a partire da un vertice verso tutti gli altri (successioni nidificate).

Pseudocodice	Javascript
<pre>Ripeti per (i=1; i<=n-1; i=i+1) Ripeti per (j=i+1; j<=n; j=j+1) Disegna un segmento usando i vertici i e j del poligono Fine Ripeti Fine Ripeti</pre>	<pre>var i; var j; var n=numerolati; for (i = 1; i <= n-1; i++) { for (j = i+1; j <= n; j++) { Segmento[Vertice[poli1, i], Vertice[poli1, j]]; } }</pre>



La parte più interna del codice è quella che crea i segmenti (comando Segmento[Punto, Punto]) partendo da un vertice (Vertice[Poligono, Numero n]) verso tutti gli altri.

L'espressione da inserire è quindi la seguente.

Successione[Successione[Segmento[Vertice[poli1, i], Vertice[poli1, j]], j, i + 1, n], i, 1, n - 1]

Ecco il risultato ottenibile.

A questo punto potremmo visualizzare il numero di lati, vertici e di diagonali.

Per stabilire il numero di diagonali è possibile utilizzare la formula:

$$n_{\text{Diagonali}} = \frac{n \cdot (n - 3)}{2}$$

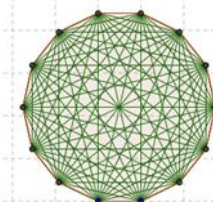
Definiamo, dalla linea di inserimento, due variabili numeriche che riportino il calcolo del numero delle diagonali e la misura dell'angolo interno.

$$n_{\text{AngoloInterno}} = 180(n - 2) / n$$

$$n_{\text{Diagonali}} = (n \cdot (n - 3)) / 2$$

È possibile a questo punto inserire un testo (ABC) dinamico a video.

Ponete la velocità dell'animazione nella proprietà dello slider a 0.1 e attivate l'animazione.

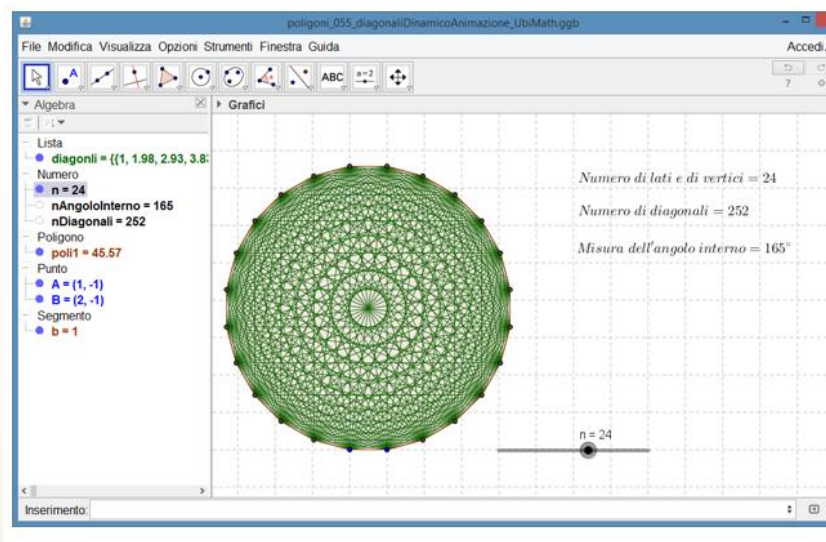


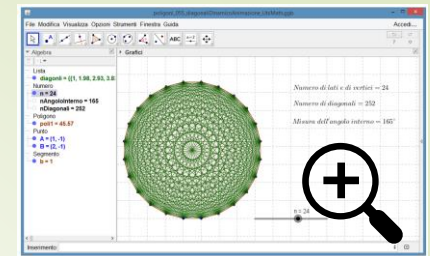
$n = 9$

Numero di lati e di vertici = 9

Numero di diagonali = 27

Misura dell'angolo interno = 140°





Per disegnare le diagonali utilizzo una lista di segmenti definiti dalle coordinate degli estremi. La lista è data dai segmenti che rappresentano le diagonali e che si dipartano da ogni vertice verso tutti gli altri. La lista viene creata tramite due comandi *Successione[]* nidificati. Il comando *Successione[Espressione, Variabile i, Numero a, Numero b]* crea una lista di oggetti utilizzando una espressione indicata dall'utente e un indice variabile (*i*) tra due numeri (*a* e *b*).

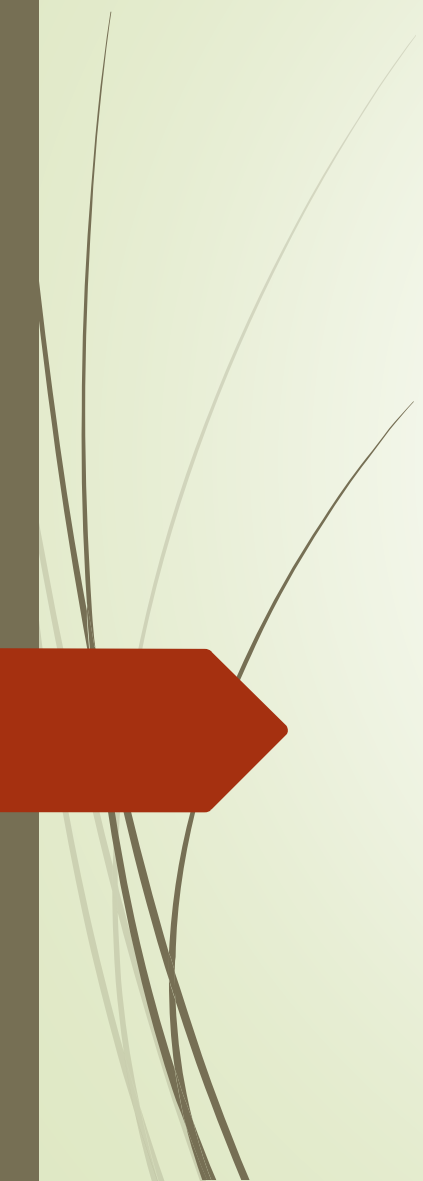
Dobbiamo, quindi, disegnare i segmenti a partire da un vertice verso tutti gli altri (successioni nidificate).

Pseudocodice	Javascript
<pre> Ripeti per (i=1;i<=n-1;i=i+1) Ripeti per (j=i+1;j<=n;j=j+1) Disegna un segmento usando i vertici i e j del poligono Fine Ripeti Fine Ripeti </pre>	<pre> var i; var j; var n=numerolati; for (i = 1; i <= n-1; i++) { for (j = i+1; j <= n; j++) { Segmento[Vertice[poli1, i], Vertice[poli1, j]]; } } </pre>

La parte più interna del codice è quella che crea i segmenti (comando *Segmento[Punto, Punto]*) partendo da un vertice (*Vertice[Poligono, Numero n]*) verso tutti gli altri.

L'espressione da inserire è quindi la seguente.

Successione[Successione[Segmento[Vertice[poli1,i],Vertice[poli1,j]],j,i + 1,n],i,1,n - 1]



Valutazione

vengono inseriti nelle verifiche sia quesiti delle gare di matematica sia legati a problemi di avvio al pensiero computazionale con punteggio aggiuntivo

Esempio 1

Il signor Euclide gioca con due numeri interi positivi e ha inventato la seguente procedura.

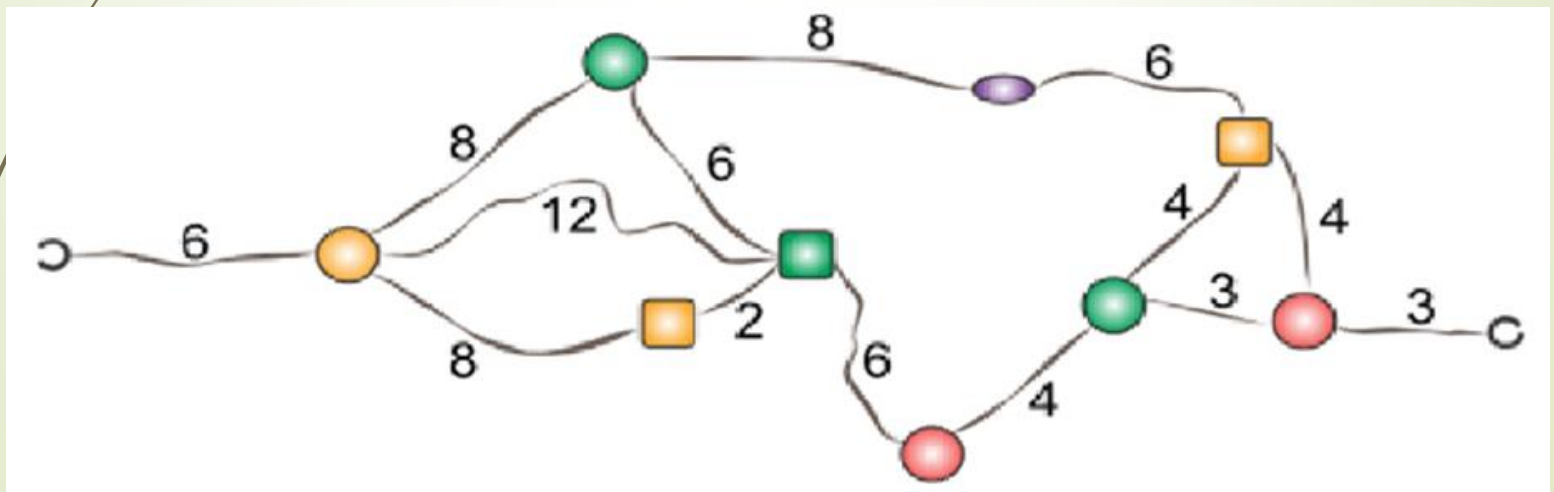
1. Scrive il numero più grande sul foglio A
 2. Scrive il numero più piccolo sul foglio B
 3. Sottrae il numero in B dal numero in A e scrive il risultato su di un foglio C
 4. Se in C c'è scritto 0, il risultato è il numero contenuto in A e la procedura è quindi terminata
 5. Altrimenti ripete dal passo 1 con i numeri contenuti in B e in C.
- ➡ Se parte dai numeri 12 e 15, qual è l'ultimo numero che sarà scritto sul foglio A alla fine della procedura?

Esempio 2

Pamela si è fatta una collanina. Ora che è finita non è sicura che le andrà bene.

I numeri dicono la lunghezza dei fili tra le perline. I ganci per chiudere la collana sono agli estremi destro e sinistro.

Quanto deve essere al massimo il girocollo di Pamela affinché possa indossare la collana?



Esempio 3

L'insegnante ha assegnato alla classe un compito per casa.

Michele considerato che per lo svolgimento impiegherebbe diverso tempo ha realizzato un programma che risponda alla richiesta e intende portarlo a scuola come soluzione.



- Qual è la richiesta del docente?
- Il blocco "suona il tamburo" è rilevante per rispondere al quesito proposto?
- Partendo da $x=0$ fino a $x>99$ quante volte viene ripetuto il ciclo "ripeti fino a quando"?
- Quale risultato restituisce il programma?
- Prospetta una soluzione al quesito posto dal docente che non utilizzi un programma ma che riduca i calcoli da eseguire.

Esempio 4

Marco deve istruire un robot che, partendo da casa, sia in grado di raccogliere tutti i fiori, evitare il bosco e portare quanto raccolto al magazzino per la spedizione, usando il minor numero di comandi possibile.

I comandi interpretabili dal robot sono:

Avanti(n caselle) – Sinistra – Destra – Prendi – Scarica



gap di competenze informatiche



► Dossier - www.ilsole24ore.com/dossier/20191206_educazione-digitale-ACa5ij3