

Quesiti da svolgere esclusivamente su carta

Castori sull'ascensore

Difficoltà: facile

Tre castori entrano in un ascensore al nono piano di un grattacielo. Uno di loro deve salire al ventesimo piano, un altro deve scendere al primo piano, il terzo deve salire al sedicesimo piano.

In quale ordine l'ascensore dovrà visitare i piani 20, 1 e 16 per minimizzare il tempo di spostamento? (Si tenga presente che la velocità dell'ascensore è la stessa sia in salita sia in discesa.)

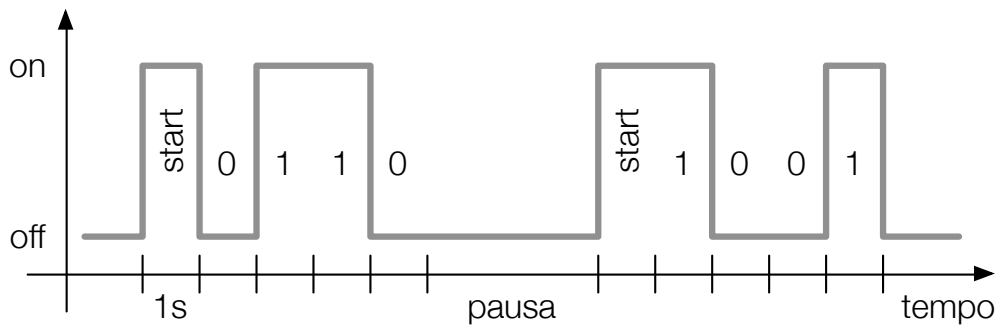
1, 16 e 20, facendo complessivamente 27 spostamenti di piano. (Può essere interessante notare che il tempo di attesa medio per castoro diminuirebbe se l'ascensore facesse il percorso 16, 20, 1, con 30 spostamenti di piano.)

Trasmissione seriale

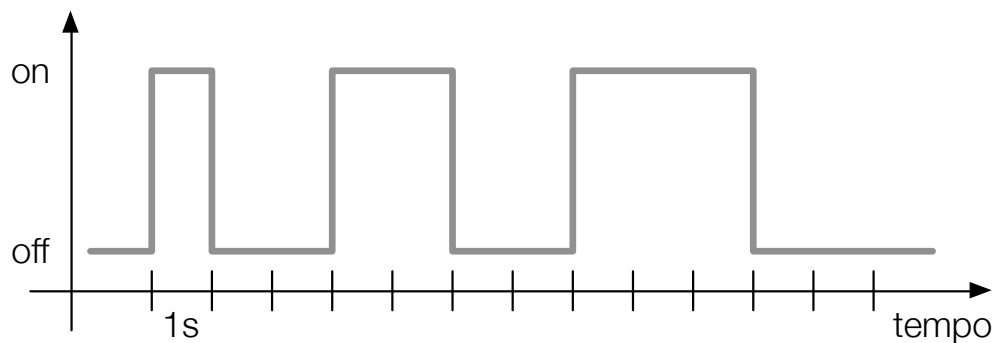
Difficoltà: facile

I castori Alice e Bruno vogliono trasmettersi segnali di notte, usando ciascuno una torcia elettrica, e a questo scopo adottano le cifre binarie: per segnalare 1 convengono di tenere accesa la torcia per un secondo, per segnalare 0 la torcia dovrà essere tenuta spenta per un secondo.

Essi decidono di trasmettersi sequenze di bit; ciascuna sequenza è composta da 4 bit ed è preceduta da un 1, e quindi ha la durata complessiva di 5 secondi. Tra due trasmissioni consecutive deve esserci una pausa di almeno un secondo. Ad esempio, il diagramma qui sotto mostra come possono essere trasmesse le sequenze 0110 e 1001.

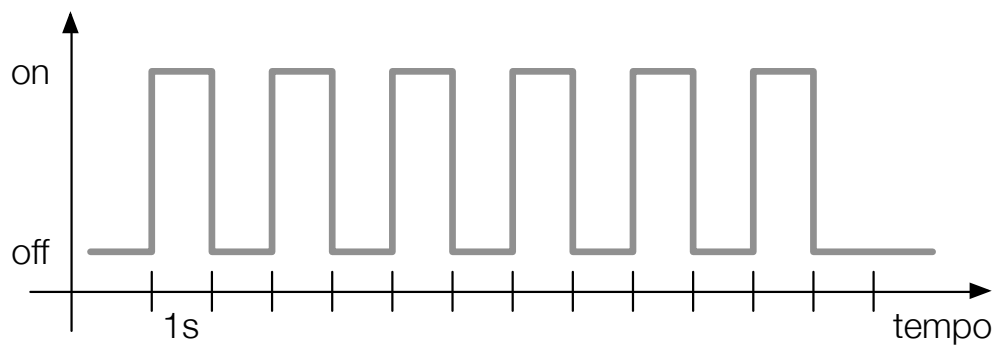


Quale sequenza o sequenze sono trasmesse secondo il seguente diagramma?



0011 e 1100.

E secondo il seguente diagramma?



0101 e, di nuovo, 0101.

Gossip a scuola

Difficoltà: media

Non passa giorno che qualcuno non arrivi a scuola con qualche fresca notizia da riferire agli amici! Agli alunni piace diffondere le ultime novità, e molti di essi, non appena ne vengono a conoscenza, le riferiscono a qualcuno in particolare, come descritto qua sotto:

- Angela lo riferisce a Daniela e Beniamino;
- Daniela lo riferisce a Michele;
- Michele lo riferisce a Pietro;
- Stefano lo riferisce a Michele, Angela e Pietro;
- Beniamino lo riferisce a Stefano, Anna e Nicola;
- Pietro non lo riferisce a nessuno;
- Nicola lo riferisce a Gianna;
- Gianna lo riferisce a Pietro;
- Anna lo riferisce a Angela e Elena.

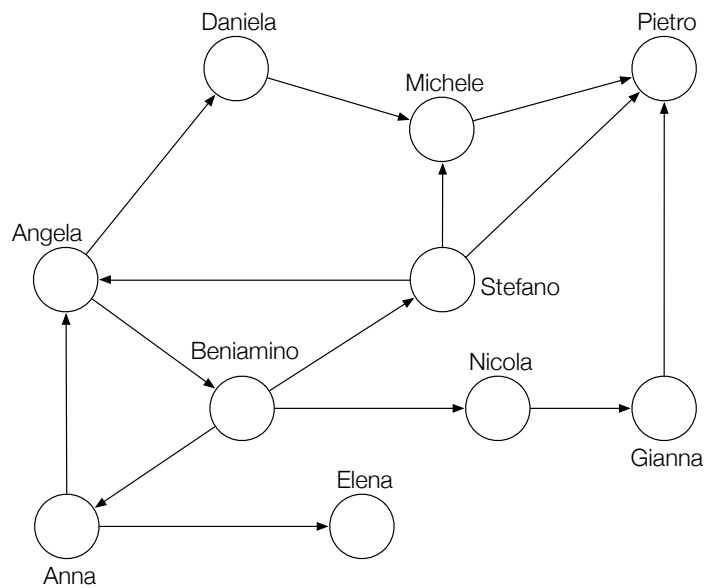
Ad esempio, quando ad Angela è riferita una notizia, da Anna o da Stefano o da qualcun altro all'esterno della scuola, ella, a sua volta, la riporta subito sia a Beniamino sia a Daniela. Stamane, quando è giunta a scuola, Anna aveva un'interessante novità, che non ha esitato a diffondere. . . Più tardi, al termine delle lezioni, sono stati appurati i seguenti fatti:

- Elena e Pietro hanno ricevuto la notizia;
- Gianna e Stefano, pur essendo entrambi presenti a scuola, non l'hanno ricevuta;
- stamane a scuola uno solo era assente.

Chi era assente? (Motivare la risposta!)

Beniamino era assente. Infatti, Gianna ha come unico informatore Nicola, mentre l'unico informatore di Stefano è Beniamino. Ma se fosse stato assente Nicola, allora a Stefano la notizia sarebbe giunta da Beniamino (il quale l'avrebbe avuta da Angela). Ovviamente, poiché Beniamino era assente, la notizia non può essere arrivata neppure a Nicola.

Per trovare la risposta, può essere utile rappresentare la situazione con un grafo:



Mini-robot

Difficoltà: difficile

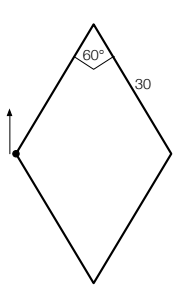
Abbiamo un mini-robot in grado di eseguire questi comandi:

- V : avanza di un passo;
- $S(g)$: ruota a sinistra di un angolo di g gradi;
- $D(g)$: ruota a destra di un angolo di g gradi.

Più comandi in sequenza sono separati dal simbolo $+$, mentre la ripetizione per n volte di un comando o una sequenza C di comandi può essere concisamente indicata da $n * (C)$. Ad esempio:

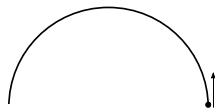
- la sequenza $V + S(30) + 5 * (V) + D(45)$ significa: (dalla posizione in cui ti trovi e nel verso in cui guardi) avanza di un passo, poi girati di 30 gradi a sinistra, poi avanza di 5 passi, infine girati di 45 gradi a destra;
- il comando $3 * (10 * (V) + D(60))$ significa: ripeti per 3 volte queste due azioni: avanza di 10 passi e poi girati di 60 gradi a destra.

Se il mini-robot si trova rivolto verso nord nella posizione indicata dal pallino nero nella figura qui sotto, quali comandi gli faranno percorrere la traiettoria romboidale, dove il lato del rombo è percorso in 30 passi? Attenzione: alla fine il mini-robot dovrà ritrovarsi nella stessa precisa posizione di partenza (rivolto a nord).



$D(30) + 30 * V + D(120) + 30 * V + D(60) + 30 * V + D(120) + 30 * V + D(30)$
oppure, più sinteticamente:
 $2 * (D(30) + 30 * V + D(120) + 30 * V + D(30))$

Se il mini-robot riceve il comando $180*(V+S(1))$, percorrerà una traiettoria “semicircolare”, come indicato nella figura qui sotto, e alla fine sarà rivolto verso sud.



Quali comandi faranno invece percorrere al mini-robot la traiettoria sotto indicata?

A diagram showing a complex trajectory starting from a point on the left and ending at a point on the right. The path consists of a curve that starts by curving downwards and to the right, then levels out, and finally curves downwards and to the right again. At the end point, there is a small vertical arrow pointing downwards, indicating the robot's orientation at the end of the path.

$90 * (V + S(1)) + 90 * (V + D(1))$

Quesiti da svolgere con l'aiuto del computer

Scacchiera

Ai Kanguri piace molto giocare a scacchi. Ma siccome sono assai indaffarati, spesso devono interrompere le loro partite e registrare in qualche modo la posizione dei pezzi.

I pezzi sono gli stessi degli scacchi tradizionali: Re, Donna, Torre, Alfiere, Cavallo e Pedone, ma le regole di gioco dei Kanguri sono alquanto diverse e non ci occorre conoscerle, salvo quella fondamentale: ci deve sempre essere *uno e un solo Re bianco e uno nero!*

Per registrare le posizioni sono in uso due codifiche alquanto diverse: la **TEX** e la **MAX**. A voi scoprirne i dettagli sperimentando col *programma Scacchiera!*

Il bottone "Genera" sotto alla scacchiera genera una nuova posizione con le due codifiche corrispondenti. È possibile spostare un pezzo a caso, cliccando sul bottone omonimo. Il programma mostrerà i due diagrammi e le codifiche corrispondenti.

Quando riterrete di aver compreso tutto sulle codifiche, potrete rispondere a queste domande.

1. Che cosa rappresenta una C maiuscola nella codifica MAX?

La C rappresenta un Cavallo bianco.

2. Che cosa rappresenta una a minuscola nella codifica MAX?

La a rappresenta un Alfiere nero.

3. Che cosa rappresenta una T maiuscola nella codifica TEX?

La T rappresenta una Torre.

4. Che cosa rappresenta una c minuscola nella codifica TEX?

Una c minuscola nella codifica TEX rappresenta la terza colonna da sinistra.

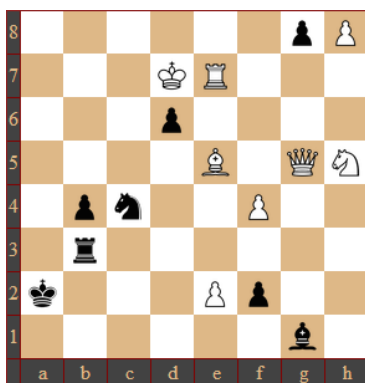
5. Se la codifica TEX di una posizione è `BRe2Df5c3Tc8Ab6/NRa2Dd6Tg5Cc5`, in quale casella si trova la Donna nera?

La Donna nera si trova in d6.

6. Se la codifica MAX di una posizione è `8/3C4/4D3/1c6/6T1/5Rd1/3r4/8`, in quale casella si trova il Re bianco?

Il Re bianco si trova in f3.

7. Scrivete la codifica MAX corrispondente alla posizione qui sotto.



`6pP/3RT3/3p4/4A1DC/1pc2P2/1t6/r3Pp2/6a1`

Nella codifica MAX vengono descritte le righe una alla volta, dall'alto verso il basso, separate dal simbolo '/'. Per ogni riga si elencano i pezzi presenti da sinistra a destra indicandoli con le iniziali (maiuscole per i bianchi, minuscole per i neri), inserendo dei numeri (da 1 a 8) se tra un pezzo e il successivo ci sono delle caselle vuote (il numero indica il numero di caselle vuote comprese tra i due pezzi).

8. Disegnate la scacchiera corrispondente alla codifica TEX
BRb5Db3Ta8Ae5h1Cf4Pe7b2/NRd6Dg8g3Af6a4d2Cc4e2Pg2.

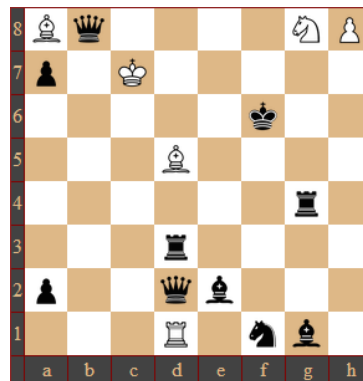


Nella codifica TEX sono elencate prima le posizioni dei pezzi bianchi (B), poi quelle dei pezzi neri (N), separate dal simbolo '/'. Per ogni colore, i pezzi presenti sulla scacchiera sono elencati nell'ordine standard (Re, Donna, Torre, ...): per ciascuno è riportata l'iniziale maiuscola, seguita dalle coordinate in cui si trova il pezzo; in caso di pezzi ripetuti, si elencano tutte le coordinate i cui si trovano quei pezzi.

9. Scrivete la codifica TEX corrispondente alla codifica MAX Ad4CP/p1R5/5r2/3A4/6t1/3t4/p2da3/T1ca1.

BRc7Td1Aa8d5Cg8Ph8/NRf6Db8d2Tg4d3Ae2g1Cf1Pa7a2

Per trovare la soluzione può essere utile disegnare la posizione rappresentata:



Clickomania

Vi ricordate le regole di *Clickomania* che avete ricevuto ieri sera?

Vostro compito è realizzare un programma per giocare a Clickomania o, piú precisamente, un programma che aggiorni la parete di gioco in seguito a ogni mossa (click) del giocatore, utilizzando l'*ambiente di programmazione Blockly*.

Blockly

Blockly consente di *costruire dei programmi* a partire da blocchi colorati predefiniti, che si possono incastrare come in un puzzle.

Non c'è un solo programma che permette di giocare a Clickomania: i blocchi a disposizione possono essere combinati a formare tanti diversi programmi, tutti funzionanti. Tuttavia ciascun blocco ha un *costo* e il vostro punteggio sarà tanto maggiore quanto minore sarà la somma dei costi dei blocchi usati.

Una volta costruito un programma, si può iniziare il gioco cliccando sul bottone "Gioca". A questo punto, ad ogni click del giocatore (voi stessi, mentre testate il vostro programma!!), la parete verrà aggiornata in base alle istruzioni del programma costruito.

Blockly consente inoltre di vedere gli effetti dell'esecuzione del programma costruito, attraverso un'animazione, direttamente sulla parete di gioco, e permette di salvare fino a 4 programmi costruiti.

Interfaccia

Lanciate il programma *Clickomania*: vedrete una pagina composta di tre colonne.

- A sinistra trovate vari elementi.
 - La parete di mattoni.
 - Un cursore con ai lati una tartaruga e una lepre, per aumentare/diminuire la velocità dell'animazione.
 - Il bottone "Gioca" che consente di iniziare una partita e che si trasforma in "Reset" durante l'esecuzione del programma; cliccando su "Reset" è possibile iniziare una nuova partita.
 - Le icone relative a salvataggio/visualizzazione/caricamento dei programmi (o versioni di uno stesso programma): cliccando sul primo simbolo si può salvare il programma attualmente in costruzione, soffermandosi col mouse sul secondo simbolo si può visualizzare il programma salvato, cliccando sul terzo simbolo si può caricare il programma salvato in precedenza.
- Al centro trovate un menu di categorie tra cui scegliere i blocchi colorati; soffermandosi col mouse su un blocco si può leggere una descrizione di cosa fa il blocco.
- A destra trovate uno spazio bianco in cui potete trascinare i blocchi scelti dal menu, per costruire le istruzioni del vostro programma. Nell'angolo in basso a destra c'è il simbolo di un cestino nel quale possono essere trascinati i blocchi che non servono piú.

Durante l'aggiornamento della parete in seguito ad una mossa (click), i blocchi in esecuzione saranno via via evidenziati. Per fare degli esperimenti, tra una mossa e l'altra potete modificare il programma; però come soluzione del quesito accetteremo soltanto un programma che non debba essere modificato a seconda della mossa!

Blocchi a disposizione

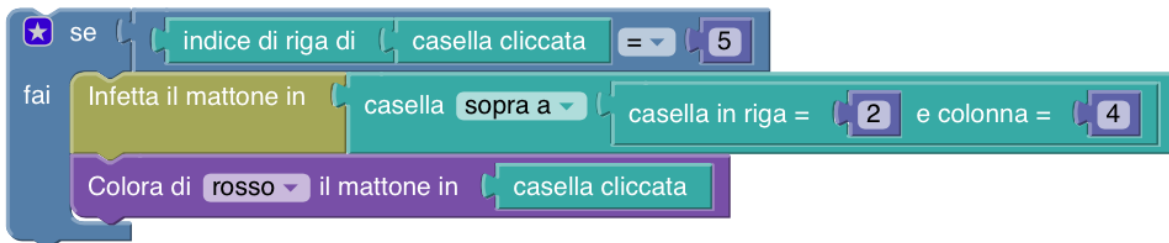
I blocchi disponibili sono raggruppati nelle seguenti categorie (alcuni blocchi sono presenti in più di una categoria): Aggiornamento parete, Caselle, Colori, Infezione, Logica, Cicli, Matematica, Variabili. Ciascun blocco ha un costo che si può visualizzare soffermandosi col mouse sopra il blocco stesso. L'elenco completo dei blocchi con i loro costi è disponibile nel foglio allegato.

Esempi

Il programma più semplice che gioca a Clickomania è il seguente. Ad ogni click, vengono rimossi tutti i mattoni che fanno parte dell'area di contagio attorno alla casella cliccata, e gli spazi vuoti che si vengono a creare vengono eliminati dalla caduta dei mattoni che si trovano sopra.

Rimuovi i mattoni nell'area di contagio attorno a **casella cliccata** e fai cadere tutti i blocchi

Vediamo ora qualche esempio un po' più complicato (non è detto che siano di qualche utilità!!):



In questo caso, se la casella cliccata si trova nella riga di indice 5, allora viene infettato il mattone nella casella di coordinate (1, 4) e viene colorato di rosso il mattone nella casella cliccata.



In questo caso, viene considerata la casella più in alto tra quelle contenenti un mattone infetto, assieme alla casella cliccata; se le due caselle contengono un mattone dello stesso colore, allora verrà rimosso il mattone che si trova nella casella di coordinate (5, 4) e cadrà l'intera pila di mattoni posizionata a partire dalla casella di coordinate (4, 4).



In questo caso, se ci sono più di tre mattoni infetti, ne verranno rimossi tre a caso; se ce ne sono tre o meno, verranno rimossi tutti.

Paleografia informatica

Questa prova va svolta su carta. Se volete, potete usare il Web per fare delle ricerche.

Essendo nato nel 4016 aveva ormai quasi quarant'anni, e da tutti era considerato un archeologo esperto, uno dei luminari nel suo campo, la *paleografia informatica*. Eppure non aveva mai partecipato prima di allora a una campagna di scavi che lo emozionasse tanto: sentiva che quello che stavano scoprendo avrebbe gettato una nuova luce sul passato dell'umanità e sui modi misteriosi con cui gli uomini, piú di due millenni prima, interagivano con i computer. Ben prima dell'avvento della programmazione olistica, prima del trasferimento osmotico della conoscenza, prima della costruzione della PCA (Pura Coscienza Algoritmica), gli umani usavano i computer in modo estremamente rudimentale, programmandone il funzionamento direttamente, e lo facevano attraverso misteriosi linguaggi (li chiamavano “**linguaggi di programmazione**”, appunto) il cui significato e la cui sintassi erano ancora in gran parte avvolti nel mistero.

Stava pensando a tutto questo quando, con la mano tremante, si avviava ad esaminare il contenuto della memoria USB trovata il giorno prima, alla fine di una lunga giornata sul campo, sotto il sole cocente dell'antica Silicon Valley, in California: stavano scavando a Mountain View, in quella che era stata la sede di un opificio informatico nato all'inizio del terzo millennio e chiamato GOGOL. Era in quell'area che, il giorno prima, avevano recuperato questo piccolo oggetto metallico. L'aveva riconosciuto subito, ne aveva già visti centinaia: era una memoria USB, forse posseduta da qualche dipendente di Gogol che la usava per trasportare i programmi da un computer all'altro, o magari per lavorarci da casa, durante la notte. Quella memoria conteneva con ogni probabilità del codice sorgente di qualche *programma informatico*.

Ora l'oggetto metallico era nel suo laboratorio; maneggiandolo con grande cautela lo aveva inserito nel lettore speciale (a prova di polvere e di contaminazione) che lui stesso aveva costruito negli anni e che gli permetteva di analizzare il contenuto di memorie di quel periodo. Dopo qualche istante, nello schermo olografico, iniziarono a comparire i piccoli involti che rappresentavano i “*file*” (cosí, nell'antichità, erano chiamati i contenitori dei codici sorgente dei programmi). Era bellissimo: c'erano decine di programmi tutti a sua disposizione, e aprendoli avrebbe potuto scoprire che cosa facevano e forse capire meglio i linguaggi in cui erano scritti e di cui ancora nessuno aveva compreso quasi nulla.

Prese in mano il pacchetto olografico di uno dei *file*, e ne svolse i lembi per aprirlo e vederne il contenuto. Quello che segue fu ciò che apparve, nella penombra della stanza.

Sapete identificare a) il linguaggio di programmazione e b) il contesto storico o informatico in cui sono stati creati i seguenti frammenti di codice?

1. Di questo reperto ormai si legge solo il nome: `euclide.py`. In che linguaggio di programmazione sarà stato scritto?

Linguaggio di programmazione: Python. I *file* contenenti i programmi python generalmente hanno un nome che termina con il suffisso (detto *estensione*, calco dell'inglese *extension*) `.py`.

2. C'è un *file* `parlamidamorekangu.ogg`. Con quale fra i seguenti programmi avete maggiori probabilità di interpretarne correttamente il contenuto?

- A. Abiword
- B. Mplayer
- C. Gimp
- D. MS Excel
- E. Notepad

Mplayer (<http://www.mplayerhq.hu>). La codifica OGG Vorbis è usata per file audio e video.

3. Ecco un frammento:

```
main( ) {  
    printf(" hello , _world" );  
}
```

Di che si tratta?

Linguaggio di programmazione: C. Si tratta del famoso esempio — poi usato nelle presentazioni di ogni nuovo linguaggio — che per primi Brian Kernighan and Dennis Ritchie ("K&R") usarono nel loro libro *The C Programming Language* (1978), in cui descrivevano il linguaggio C, da loro inventato.

4. Ecco un altro frammento:

```
to spiral :size :angle
if :size > 100 [stop]
forward :size
right :angle
spiral :size + 2 :angle
end
```

Di che si tratta?

Linguaggio di programmazione: Logo. Logo è un linguaggio inventato da Seymour Papert per insegnare la geometria e la matematica tramite la programmazione. Il programma permette di disegnare "spirali" che paiono molto diverse fra loro, ma invece sottintendono una comune struttura geometrica.

5. Ecco un altro frammento:

```
10 Print Chr$(205.5+RND(1)); : GOTO 10
```

Di che si tratta?

Linguaggio di programmazione: Basic. Il Basic è un linguaggio di programmazione inventato nel 1964 da John Kemeny e Thomas Kurtz per diffondere la conoscenza della programmazione presso un pubblico meno specialistico di quello cui all'epoca veniva insegnato a programmare in Fortran o Cobol. Divenne molto popolare quando fu installato sui primi *personal computer*, in particolare l'Altair (il primo prodotto Microsoft) e l'Apple II. Il programma trovato dal nostro paleografo era un classico dei primi passi nella programmazione del Commodore 64: l'effetto è la stampa di griglie casuali che tassellano tutto lo schermo.

6. Ecco un altro frammento:

```
((lambda (x)
  (list x (list (quote quote) x)))
 (quote
  (lambda (x)
    (list x (list (quote quote) x))))))
```

Di che si tratta?

Linguaggio di programmazione: Lisp. Il Lisp è uno dei primi linguaggi di programmazione (il primo fu il Fortran), definito nel 1958 da John McCarthy. Fra i motivi che lo rendono particolarmente interessante c'è il fatto che la forma in cui i programmi vengono scritti è la stessa che usa l'interprete o il compilatore per realizzarne l'esecuzione. Questa sua natura "riflessiva" lo rende particolarmente adatto per scrivere programmi, come quello ritrovato dal nostro paleografo, che producono sè stessi, i cosiddetti *quine*, dal nome del filosofo Willard Van Orman Quine.

7. Ecco un altro frammento:

```
mortal(X) :-  
    human(X).  
human(socrate).  
?- mortal(socrate).  
yes
```

Di che si tratta?

Linguaggio di programmazione: Prolog. Il Prolog è un linguaggio di programmazione basato sulla logica del primo ordine (quella che contiene solo predicati logici e quantificatori): il programma rappresenta l'inferenza possibile tramite il sillogismo "Tutti gli uomini sono mortali; Socrate è un uomo; ergo: Socrate è mortale."

8. Ecco un altro frammento:

```
tell application "Finder"  
    make new folder at desktop  
end tell
```

Di che si tratta?

Linguaggio di programmazione: Applescript. Applescript è un linguaggio di programmazione creato con lo scopo di controllare le applicazioni al fine automatizzare delle operazioni ripetitive all'interno del sistema operativo Mac OS. Una delle caratteristiche di questo linguaggio è quella di essere facilmente leggibile in quanto i programmi sono scritti utilizzando una formulazione abbastanza simile a quella delle frasi del linguaggio naturale: il programma mostrato crea una nuova cartella all'interno della scrivania.