analisi torrei di HanoI di G.C.

# Analisi:

* All’apertura del programma, si chiede all’utente il tipo di torre di Hanoi che voglia, se quella numerica (con l’elenco dei passaggi) o quella grafica.
* NEL CASO NUMERICO (Opzione 1 ed eventualmente 2):
  + Si chiedono all’utente quanti cilindri voglia.
  + Si chiede all’utente se vuole delle pause personalizzate (procede sul click di un carattere) oppure il delay automatico, che procede fino alla fine da solo.
  + Se l’utente vuole un delay, si chiede l’inserimento.
  + La Graphics la penso dopo a causa di problemi tecnici con le librerie.
  + Il ragionamento principale implica una funzione con in ingresso il numero di dischi e il nome delle 3 colonne.
  + Se il disco che viene spostato è il numero uno, la funzione ricorsiva termina.
  + Nel caso invece non sia uno, si scambiano le posizioni, si decrementa di 1 il numero del cilindro che si sta spostando e si richiama la funzione.
* NEL CASO GRAFICO (Opzione 3):
  + Si richiedono in ingresso gli stesso dati della variante precedente, tranne il tipo di pausa (si può scegliere solamente il delay.
  + La funziona ha un funzionamento simile, sempre ricorsivo.
  + Alcune differenze principali sono:
    - Ad ogni ciclo si assegna al vettore A il valore zero, perché si presume che il suo disco in cima sia stato spostato al vettore B, che quindi assume come valore in cima il valore num.
    - Si mostra una grafica dopo ogni spostamento.
    - Ogni volta che l’intera grafica è stata mostrata, c’è la pausa.
  + Una volta arrivato alla fine c’è una pausa più lunga, dopo la pausa saranno di nuovo mostrate le opzioni.
* Fine.

# Note sulla grafica:

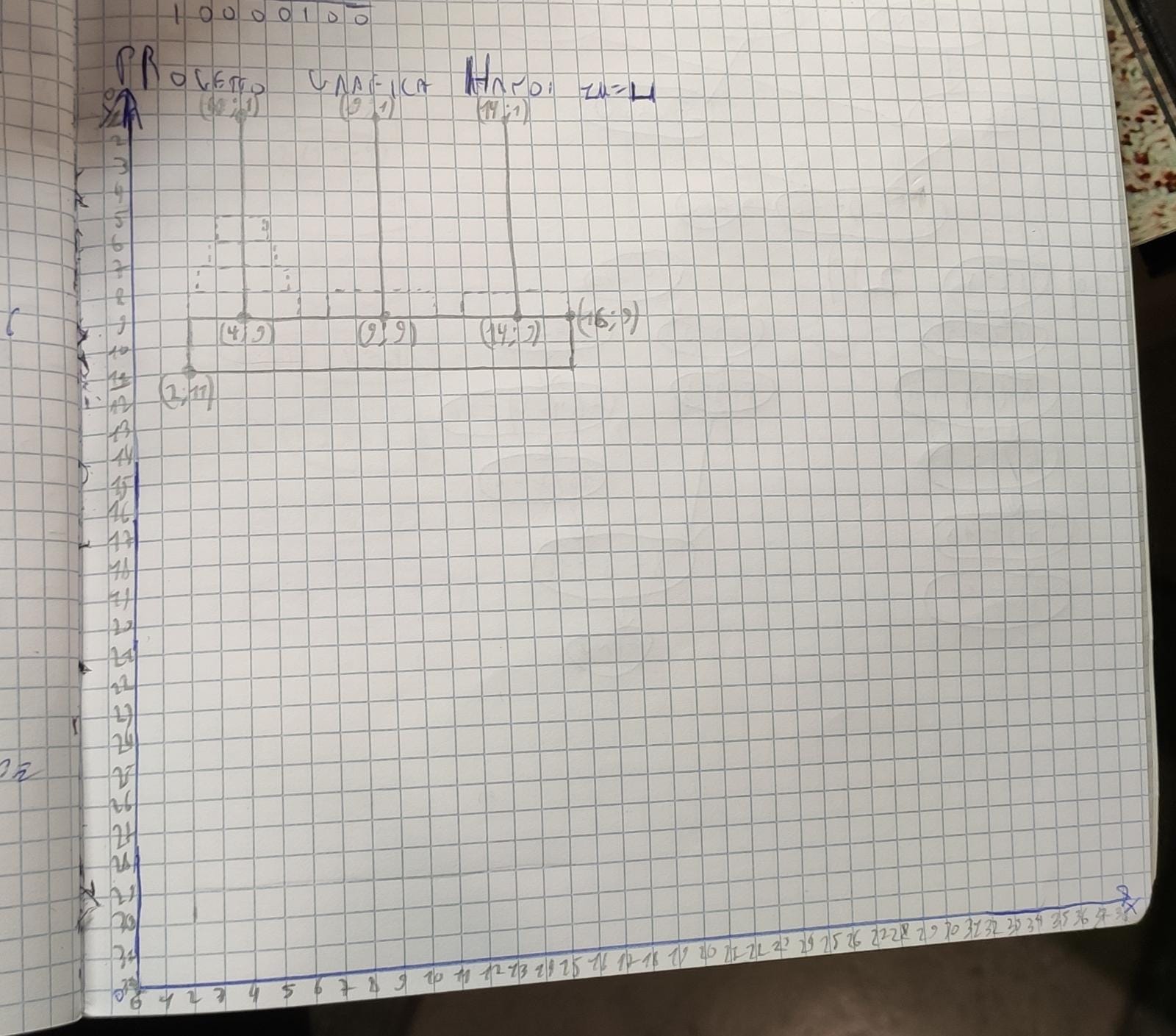
Funzionante solamente su CodeBlocks al momento della scrittura.  
La grafica è modificabile per grandezza e colori, le coordinate vengono viste in “unità”, ogni unità ha una misura fissata dal codice, e in base al suo valore sarà mostrato un grafico più o meno grande.  
Il numero di cilindri è illimitato fino ai limiti della stessa libreria grafica (si rompe con valore oltre il 100 per esempio), e i limiti dei vettori, questi cambieranno la dimensione in modo dinamico tramite espressioni matematiche di tipo vettoriale.

Il valore consigliato come unità di default è 50, massimo 80, oltre potrebbe essere troppo grande, i valori più piccoli invece funzioneranno ma creeranno un grafico più piccolo e brutto da vedere.

Non è stato possibile applicare una pausa sul click di un tasto, dal momento che il getchar per qualche motivo resetta la grafica.

Per creare la grafica, inizialmente vengono disegnati usando le coordinate parametrizzate, la base e le aste.  
Successivamente, tramite cicli for, vengono disegnati i cilindri, appoggiandosi a vettori che memorizzano una “immagine” della situazione attuale delle torri/aste con i cilindri.  
Il codice grafico con la graphics.h rimane simile a quello grafico con i numeri, e quello dei numeri ha una struttura simile a quella classica iniziale di base.

Schema sul quaderno con il piano cartesiano:



## Fix problema system(“cls”) che non cancella la grafica:

Ho pensato a due possibili soluzioni:

* Cancellare con un rettangolo nero la vecchia graphics, dando l’illusione di un reset (questo è stato applicato).
* Andare a capo per un numero di righe opportuno (usato solamente nel vecchio Memory con la graphics).

## Complessità computazionale:

La complessità è esponenziale, ossia O(2^n).

# Descrizione funzionalità programmi:

Ci sono 3 programmi nel menù:

* Hanoi normale:
  + Risolve l’Hanoi comunicando gli scambi effettuati in modo ricorsivo.
* Hanoi grafico testuale/numerico:
  + Risolve l’Hanoi comunicando gli scambi effettuati in modo ricorsivo.
  + Mostra il contenuto in dei vettori delle torri, tramite testo.
* Hanoi grafico con la graphics.h:
  + Risolve l’Hanoi utilizzando un metodo simile a quello precedente, sempre in modo ricorsivo e tramite dei vettori, e stampa con l’ausilio della graphics.h le torri di Hanoi con una funzione apposita.

Ognuno di questi programmi ha dati in ingresso simili ma funzioni diverse, anche se con lo stesso scopo finale ossia trovare una soluzione per la torre di Hanoi.  
Tutti i programmi permettono di scegliere se come colonna di destinazione, usare la seconda o terza asta/colonna.

### note sulle dimensioni della graphics:

Tramite la variabile globale modificabile nel codice “unita” si può cambiare la dimensione della graphics secondo le proprie necessità (magari è troppo grande oppure piccola ed è necessario cambiarlo).

# Regole del gioco:

* I cilindri devono essere in ordine crescente, non si può mettere un cilindro più grande sopra ad uno più piccolo.

## Compito:

Creare Hanoi con la grafica in modo ricorsivo.