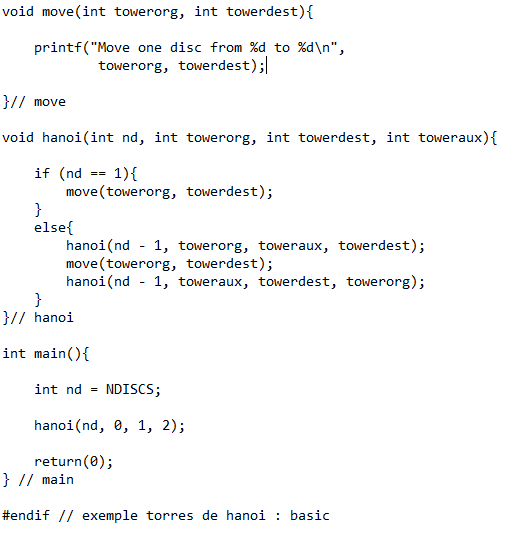
**Comprensió de la Recursivitat**

**6.1 Expliqueu amb les vostres paraules l’algorisme recursiu que proporciona la solució del joc, detallant quina és la base de recursió, quina és la regla recursiva i perquè funciona. Afegir dibuixos que us semblin necessaris per fer entenedora l’explicació i el concepte.**

**6.1.1 Resposta:**



El que fa bàsicament, es cridar a la funció Hanoi passant-li uns valors per a la posició de origen, posició de destí, posició auxiliar.

Els disc ens venen definits poden ser 3/4/5.

Si es el últim disc, movem la posició origen per la posició de destí de la ultima peça, si no tenim el últim disc, entra en una condició, que ens torna a cridar a la funció Hanoi, amb n-1 discs, enviant com a paràmetres, els mateixos que em passat, intercanviant la posició final per la auxiliar.

Després ell mateix torna a intercanviar les peces que tenia en la posició de origen per la de destí.

Una vegada a acabat torna a cridar a la funció Hanoi, amb el valor n-1 es a dir tornem a restar un disc, aquí enviem primer la posició auxiliar, segon la posició de destí i finalment la posició de origen.

I com que la funció torna a cridar 2 vegades a la funció Hanoi, aquestes crides la tornaran a cridar 2 vegades mes, i així successivament, fins que arribem a nd=1, que es la condició que posem i significa que estem intentant col·locar el últim disc.

El programa ensenya per pantalla cada un dels moviments quant crida a la funció move.

Amb valors

5,0,1,2

Hanoi 4,0,2,1

4,0,2,1

Hanoi 3,0,1,2

...

Moure 0-2

Hanoi 3,2,1,0

...

Moure 0-1

Hanoi 4,2,1,0

4,2,1,0

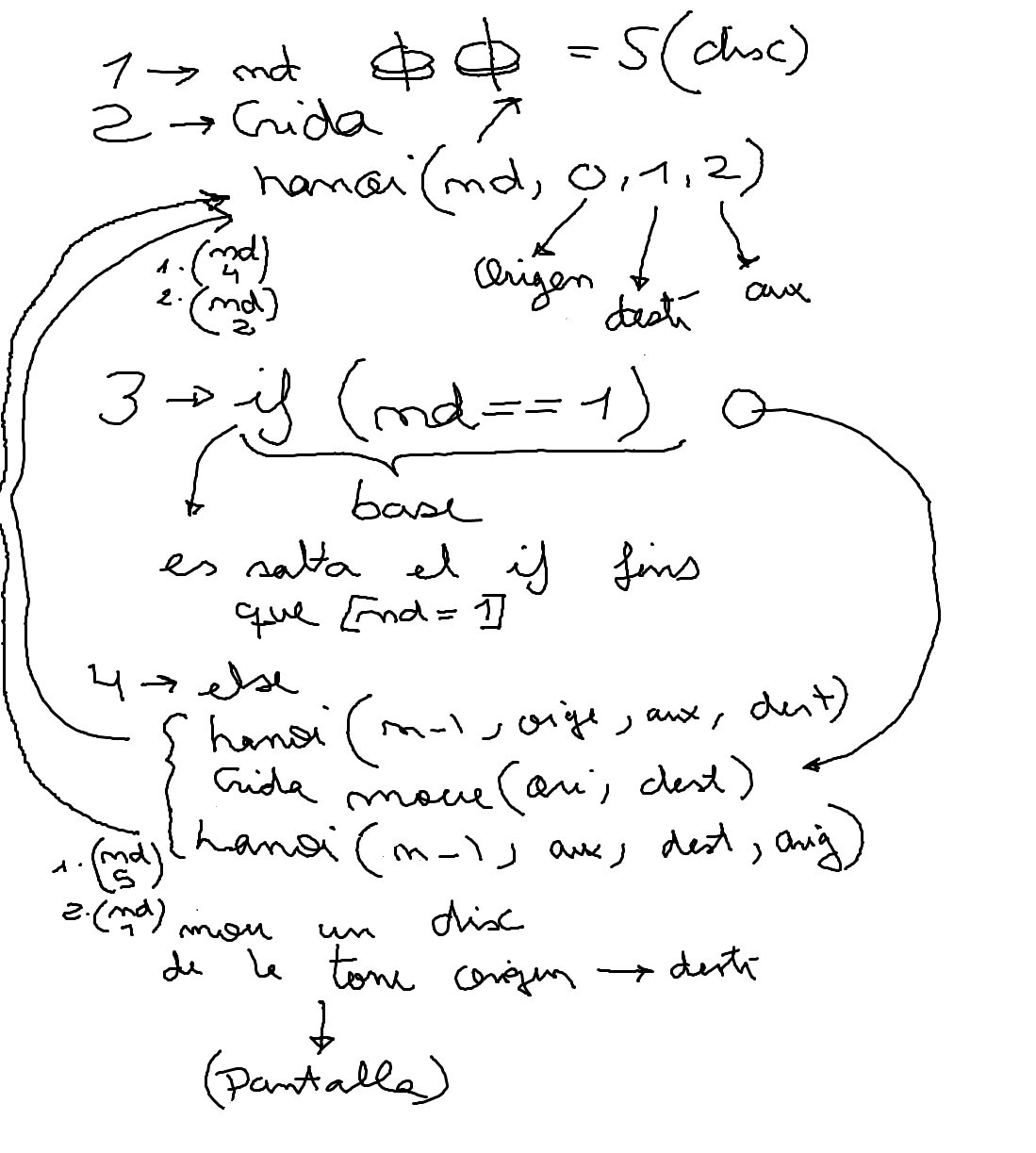
Hanoi 3,2,0,1

...

Moure 2-1

Hanoi 3,0,1,2

...



**6.2 Construir l’arbre d’execució del model de les còpies fet a ma per l’execució de les crides 1) hanoi(3,0,1,2), hanoi(4,0,1,2) i hanoi(5, 0, 1, 2). Posar el número de moviment en cada moviment que apareix a l’arbre, i posar el nivell de profunditat en cada nivell. Comprovar que els moviments i els nivells de profunditat concorden amb els donats pel programa. (Fer foto de cada un dels arbres i incloure en l’entrega els tres fitxers per separat) a) Quina és la torre destí, i nivell de profunditat del primer moviment en cada un dels tres casos? b) L’últim moviment del primer sub-arbre de profunditat 2 en el cas de 3 discs és el 3. Quina sub-torre ha mogut aquest sub-arbre al acabar aquest moviment? c) Quin és l’últim moviment del primer sub-arbre de profunditat 2 en el cas de 4 discs i quina sub-torre ha mogut? d) Quin és el primer moviment de l’últim sub-arbre de profunditat 2 en el cas de 4 discs i a quin nivell de profunditat s’efectua. Executa el codi bàsic amb debugger i para l’execució en aquest nivell i fes una captura del callstack i inclou-la aquí. Comprova i comenta els valors dels paràmetres de les diferents crides recursives del debugger en relació el camí del teu arbre recursiu fet a ma.**

**6.2.1 Resposta:**

**6.3 Com s’ho fa el programa bàsic per recordar quin moviment ha de fer i ser coherent amb els moviments anteriors i posteriors si no guarda informació ni dels discs ni de l’estat de les torres? Raona la resposta. Podem saber quin disc es mou en cada moviment? Si és que si explica com podem fer que el programa bàsic ens ho mostri. Si és que no indica com s’ha de fer per poder imprimir amb els moviments quin disc es mou.**

**6.3.1 Resposta:**

Perque, la condicio de recursivitat es que le nombre de discos arribi a 1, es a dir que volem que ND = 1, i dins de la recursivitat, n’em decreixent el ND, cada vegada que es torna a cridar la funcio.

El programa no recorda quin disc es te que moure, però cada vegada que es crida a la funcio move(\*\*\*\*\*\*), el sistema ensenya per pantalla quin disc esta movent i ho imprimeix, de tal forma que, si ho anesim veient pass per pass, podriem veure com mou disc a disc.

Ja que cada vegada que crida la funcio move(\*\*\*\*) esta movent 1 disc, de la torre\_origen a la torre\_destí.