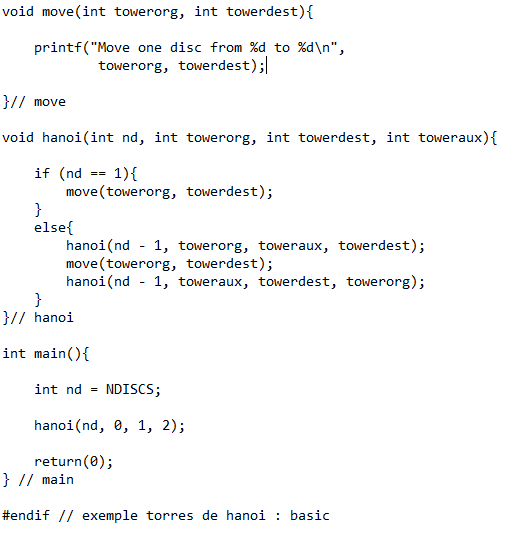
**Comprensió de la Recursivitat**

**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

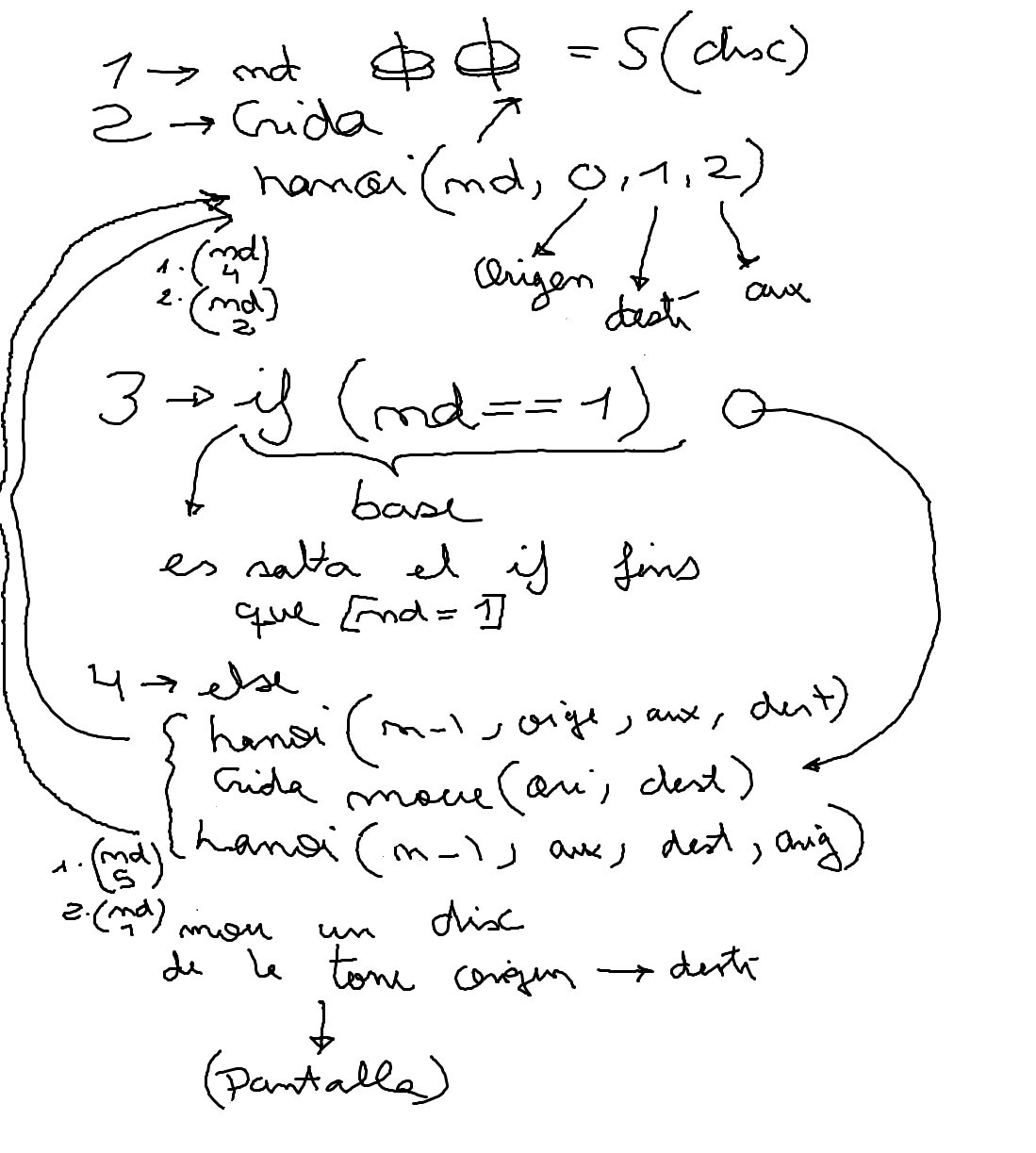
...

Moure 2-1

Hanoi 3,0,1,2

...

Dibuix pagina següent:



**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.

**6.4 Construcció d’un moral per la comprensió: Executar l’executable hanoiplus que implementa la solució que se us demana en aquest enunciat.**

**Executeu-lo pels casos que vulgueu, se us proporciona l’output del cas de 3, 4, i 5 discs per si no us executa al vostra ordinador. Construir un moral gran que contingui l’arbre de recurisió anterior, però ara afegint l’estat de les torres en cada node de l’arbre i el número de disc que es mou en cada moviment. Això ocupa espai i pot ser gran. Ho podeu fer retallant els dibuixos del fitxer de sortida i posant-los espaiadament en una cartolina o similar (per poder-ho transportar). Feu el moral pel cas que us sembli millor per entendre-ho, 3, 4 o 5 discs. Fer una foto d’aquest moral amb màxima resolució per poder visualitzar-lo correctament a diferents nivells de zoom/detall electrònicament. Entregar aquesta foto com un fitxer separat en la vostra entrega final. Si heu fet més d’un cas aporteu tants fitxers com arbres hagueu muntat. Anomenar aquest fitxer com** **moral-hanoi-n amb l’extensió del format que sigui (jpg/gif/...) i on n indiqui el nombre de discs que correspon el moral. Per cada sub-arbre crear un paper que cobreixi tot el seu sub-arbre (fills) i que contingui l’estat final de les torres de hanoi que és el final de l’últim moviment del sub-arbre. Quan es posi sobre el paper estem eliminant l’execució i detall d’aquest sub-arbre pel seu efecte i per tant emula el retorn recursiu de tot el sub-arbre que oculta. Si disposeu de tots els papers per cada arbre i obriu l’arbre al anar fent crides recursives, i els tapeu amb el resultat per anar retornant de les crides recursives, teniu una eina visual pel seguiment de l’execució recursiva del problema i per consolidar la comprensió. Per cada un d’aquests papers que sigui blanc en una cara i l’estat de sortida en l’altre. Així si no s’ha creat encara aquest sub-arbre en l’execució el podeu fer servir per deixar en blanc l’espai en l’arbre total, anar destapant quan es fan les crides i tornar a tapar amb el resultat amb els retorns de les crides. Feu un seguit de fotos amb els papers ben col·locats per mostrar la seqüencia de l’execució. Entregar aquesta seqüencia de fotos en un fitxer separat ppt per exemple i incloure l’explicació de cada punt que es mostra. Se us demana que la seqüència mostri almenys quatre (4) punts diferents de l’estat de l’execució. Portar el moral en paper a la sessió de pràctiques que es revisa la pre-entrega.**

**6.4.1 Resposta: (En un fitxer a part o fotos i l’explicació aquí.)**

PPT apart, fotos en un zip anomenat com el fitxer del moral que no cap aquí.

Explicacio parcial al ppt.

**6.5 Després d’aquest estudi, creieu que teniu el concepte de recursivitat clar? Indiqueu la impressió de la vostra comprensió i aporteu aquí les preguntes, dubtes o comentaris que tingueu pendents de resoldre per completar la comprensió de la recursivitat. Si creieu que sí expliqueu què és el que més us ha ajudat a entendre-ho i perquè.**

**6.5.1 Resposta:**

La veritat, no ens ha quedat gens clar, es un concepta difícil, esta poc explicat, i tot parteix de uqe això ho entes perfectament.

La veritat, faria falta, com ja es va comentar en la practica passada, una clase de practiques abans de la pre-entrega.

Tot el que hi ha al campus, son ejemples curts, quant tenim que entregar un document que igual roça les mes de 10 pagines...

Es a dir es demana moltisima perfeccio, però els outputs via txt surten mal formatejats, el .exe no funciona a tots els ordinadors... es a dir els recursos per poder entendre la recursivitat estan malament montats i/o presentats.

Finalment dir que el que hem pogut entendré ho varem treure al seminari anterior i no gracies al codi.

De tota manera com s’ha especificat a la primera linea, el concepta de recursivitat no ha quedat gents clar.

Tenim clar que es nececita una condicio inicial, per que s’entri dins del bucle de recursivitat, i quant la condicio ja no es cumpleixi, el programa finalitzara les ultimes instruccions i saltara fora de la funcio de recursivitat.

Pero tret de això que es teoricament el concepte, no tenim molt clar per que es te que fer així i no amb altres bucles o quant fer servir la recursivitat.

**Disseny (de l’Ampliació) del Programa**

**6.6 Quina diferència hi ha entre només treure una línia de text tal i com fa el codi bàsic proporcionat comparat en generar el dibuix complert que es demana que faci el programa haniiplus? És a dir explicar i justificar els canvis que heu de fer en el programa per incrementar la funcionalitat des del punt de vista del disseny (i sense haver-ho implementat encara). Revisar totes les funcionalitats que es demana afegir (ja que hi ha més que recursivitat) en aquesta pràctica.**

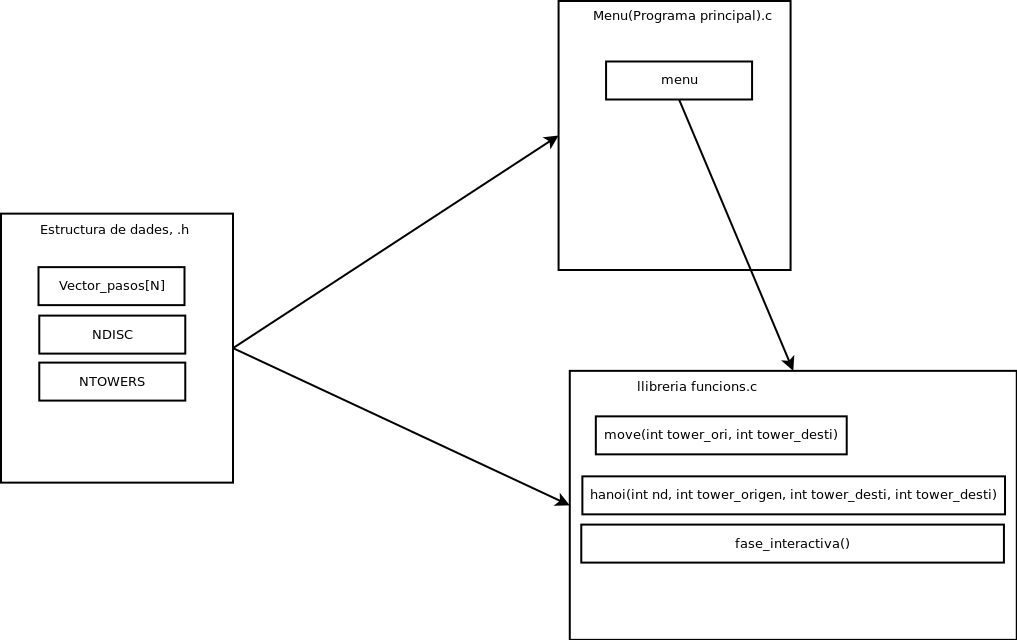
**6.6.1 Resposta:**

**6.7 Quina estructura de dades i variables principals cal definir? Explicar la necessitat de cada estructura i variable i el perquè de com i on es defineixen i declaren. (per la pre-entrega suposar memòria estàtica i per l’entrega final aportar les declaracions necessàries amb memòria dinàmica).**

**6.7.1 Resposta:**

**6.8 Aportar el dibuix del mapa conceptual del disseny (primer esborrany abans de començar la implementació) i explicar-lo**

**6.8.1 Resposta:**

El mapa conceptual e 3 fitxers, un fitxer anomenat «header.h» contindra totes les declaracion de variables i l’estructura de dades del programa, en principi nececitem 2 variables el numero de discos i el numero de torres, i ames a mes, un vector que contindra totes les posicions de les diferents peces ja sigui torre\_origen, torre\_final etc...

Un segon fitxer que te relacionada la capçalera es el de llibreries i funcions, es aquell fitxer que contindra totes les funcions que no siguin escollir de un menú, es a dir, moure, la funcio hanoi, i algunes mes.

Finalment tindrem el fitxer principal amb el main, es un fitxer petit, que només te 1 funcio menú, i que de alla crida les diferents funcions.