

Problema 1 Miscel·lània (3,33 punts)

1. Quina implementació de la classe `Taula<T>` triaries per cadascun dels casos següents? Indica per cadascun, quina seria la clau i quin el valor.
 - (a) El sistema informàtic de gestió d'un hospital necessita saber amb tota rapidesa les dades del pacient que es troba en una determinada habitació.
 - (b) El sistema d'ajut d'un famós processador de textos està estructurat de forma que quan l'usuari entri les primeres lletres d'alguna comanda del processador, el sistema retorni ràpidament una llista de totes les comandes que comencen per aquelles lletres.
 - (c) Una cadena de supermercats ha implantat un sistema informàtic que permet la lectura del codi de barres de cada producte que es troba en el supermercat. En el mateix moment de la lectura, cal registrar a la nota del client el nom i preu del producte i, també cal restar una unitat de les existències d'aquell producte a la base de dades del supermercat.
 - (d) Els mossos d'esquadra han demanat a una empresa de software que els dissenyi un sistema que els permeti saber immediatament, a la vista d'una matrícula, totes les dades del cotxe i del seu propietari.
2. Considerem un *hash* tancat amb $B = 7$. Defineix una família de funcions de redispersió que provoqui un problema d'apinyament primari. Il·lustra aquest problema amb la inserció de 4 claus (per simplicitat, podeu suposar que les claus són numèriques). Defineix, seguidament, una altra família de funcions de redispersió que elimini l'apinyament primari però que mantingui el secundari. Il·lustra tots dos fets amb la inserció de les **quatre mateixes claus** que en el cas anterior.
3. Enumera dues condicions necessàries per a que un hash tancat tingui un cost $O(1)$.

Problema 2 Llista genèrica i polimòrfica (3,34 punts)

Proposa una implementació per a la classe `LlistaP<T>` que compleixi els següents requeriments:

- Ha de seguir escrupulosament l'especificació de la classe `Llista<T>` presentada a l'assignatura.
- Implementació doblement enllaçada, utilitzant memòria dinàmica.
- Una llista l definida com `LlistaP<T> l`; haurà de ser capaç de contenir, simultàniament, elements de la classe `T` i de qualsevol subclasse de `T`.

Concretament, es **demana**:

1. Fer la representació de la classe `LlistaP<T>` i de les classes `Node<T>` i `IteradorLlistaP<T>` (per representació entenem els atributs de la part privada d'aquestes classes).
2. Implementar l'operació de la classe `LlistaP<T>`:
`void inserir(T& x, IteradorLlistaP<T>& it);`
3. Implementar les classes `ObjecteCasa` i `Roba` de manera que es pugui crear una llista polimòrfica d'objectes de la casa amb els requeriments presentats.
4. Hi hauria alguna dificultat en la implementació de l'operació
`void consulta(T& x, IteradorLlista<T>& it)?`
Donar una proposta de solució d'aquesta dificultat canviant la capçalera d'aquesta operació.

5. A més a més de tot el que conté l'especificació de la classe `Llista<T>`, què hauria d'indicar molt clarament l'especificació de la classe `LlistaP<T>`?

No cal fer cap especificació (cal seguir la presentada a classe).

Problema 3 Iteradors sobre arbres binaris en inordre (3,33 punts)

Implementa la classe `IteradorABinIn<T>` que permet fer recorreguts en inordre sobre un arbre binari implementat en memòria dinàmica amb apuntadors al fill esquerre, al fill dret i **al pare**.

Es demana:

1. Representació (i.e. atributs de la part privada) de les classes `Abin<T>`, `IteradorABinIn<T>` i `Node<T>`.
2. Implementació de les operacions de la classe `IteradorABinIn<T>` següents:
 - `void situarInici();`
 - `IteradorABinIn<T> operator++();` (`++it;`)
 - `bool fi();`

Aquestes operacions hauran de seguir estrictament l'especificació donada a classe.

Intenta trobar una implementació elegant pel cas en què l'iterador es refereixi a un element més enllà del darrer element de la llista de nodes (i.e. una implementació que no exigeixi la consideració de casos particulars en el disseny de les operacions esmentades anteriorment).