Segon examen parcial de PRO2 Model A Durada: 2h 50m 19/01/2018

Cognoms **Nom** **DNI**

Problema 1 (6 punts)

En aquest problema heu implementar alguns mètodes públics de la classe *Llista*, la implementació de la qual heu vist a classe de teoria. Mostrem a continuació la representació del tipus *Llista*, que utilitza nodes *doblement encadenats* amb punters a l'element següent (`seg`) i l'element anterior (`ant`). Aquesta implementació de la classe *Llista* conté els atributs següents: (1) `longitud`, de tipus enter; (2) `primer_node`, un punter a `Node` que apunta al node que representa el primer element de la llista; (3) `ultim_node`, un punter a `Node` que apunta al node que representa l'últim element de la llista; i (4) `act`, un punter a `Node` que apunta al node que representa l'element actual de la llista, anomenat *el punt d'interès* de la llista.

```
template <class T> class Llista {
  private:
    struct Node {
        T info;
        Node* seg;
        Node* ant;
    };
    int longitud;
    Node* primer_node;
    Node* ultim_node;
    Node* act;      ...   // especificació i implementació d'operacions privades
  public:
     ...                  // especificació i implementació d'operacions públiques
};
```

En les vostres respostes a aquest problema no podeu utilitzar cap mètode privat o públic de la classe `Llista` que heu vist a classe de teoria (per exemple, `copia_node_llista`, `esborra_node_llista`, `afegir`, `eliminar`, `concat`, `inici`, `fi`, `avanca`, `retrocedeix`, `actual` o `modifica_actual`). Si utilitzeu algun mètode privat o públic auxiliar en la vostra resposta a algun apartat, heu d'especificar-lo –escrivint-ne clarament la capçalera, la precondició i la postcondició– i implementar-lo en aquesta resposta.

1.1 Definiu el mètode públic `push_back(x)`, que afegeix l'element x al final de la llista paràmetre implícit. Tingueu en compte que el paràmetre implícit pot ser una llista buida o no buida. Per exemple, si x és 6 i a és la llista $\{1,5,3,4,2\}$, després de la crida `a.push_back(x)`, a ha de ser la llista $\{1,5,3,4,2,6\}$. [2 punts]

```
void push_back(const T& x) {
/* Pre: El paràmetre implícit és igual a la llista {e_1,...,e_n}. */
/* Post: El paràmetre implícit és igual a la llista {e_1,...,e_n,x}. */
```

1.2 Definiu el mètode públic `interseccio_ordenada`, que modifica el paràmetre implícit de manera que en contingui la intersecció amb la llista c2. Observeu que aquesta operació ha d'alliberar la memòria de tots els nodes que es descartin del paràmetre implícit en calcular-ne la intersecció amb c2. Per exemple, si a és la llista $\{1,3,5,7,8,9,10\}$ i b és la llista $\{-1,1,2,3,7,8,9\}$, després de la crida `a.interseccio_ordenada(b)`, a ha de ser la llista $\{1,3,7,8,9\}$ i el seu punt d'interès ha d'apuntar a 1. [4 punts]

void *interseccio_ordenada* (**const** *Llista* & *c2*) {
/* Pre: El paràmetre implícit és igual a C1. C1 i c2 estan ordenades en ordre creixent i no contenen elements repetits. */
/* Post: El paràmetre implícit conté els elements de C1 que pertanyen a la intersecció de C1 i c2 en el mateix ordre en què estaven en C1. El punt d'interès del paràmetre implícit apunta al seu inici. */

Segon examen parcial de PRO2 Model A Durada: 2h 50m 19/01/2018

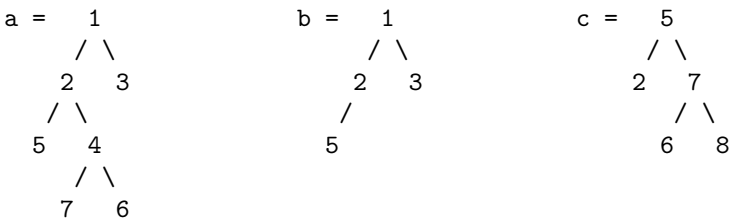
Cognoms **Nom** **DNI**

Problema 2 (4 punts)

Implementeu eficientment el mètode públic `poda_subarbre`, especificat a continuació.

```
bool poda_subarbre(int x);
/* Pre: El paràmetre implícit és un arbre binari A de enters. Els valors dels nodes de A
   són tots diferents. */
/* Post: Si x és el valor d'algun node de A, el resultat és cert i el paràmetre implícit és
   el resultat d'eliminar de A el node amb valor x i tots els seus descendents; altrament,
   el resultat és fals i el paràmetre implícit no varia (és a dir, és A). */
```

Per exemple, si t és igual a l'arbre a de la figura i x és 4, després de la crida `t.poda_subarbre(x)`, t ha de ser l'arbre b de la figura i el resultat cert. De la mateixa manera, si s és l'arbre c de la figura i z és 3, després de la crida `s.poda_subarbre(z)`, s no varia, és a dir, s ha de ser l'arbre c de la figura i el resultat fals.



Donem a continuació la definició del tipus `Arbre`, que heu d'utilitzar per resoldre aquest problema.

```
template <class T> class Arbre {
  private:
    struct Node_arbre {
      T info;
      Node_arbre* segE;
      Node_arbre* segD;
    };
    Node_arbre* primer_node;
    ...                 // especificació i implementació d'operacions privades
  public:
    ...                 // especificació i implementació d'operacions públiques
};
```

Si utilitzeu algun mètode privat o públic de la classe `Arbre` que heu vist a classe de teoria (per exemple, `copia_node_arbre`, `esborra_node_arbre`, `a_buit`, `es_buit`, `arrel`, `plantar` o `fills`) en la vostra resposta, heu d'especificar-lo –escrivint-ne clarament la capçalera, la precondició i la postcondició– i implementar-lo en aquesta resposta.

Concretament, es demana implementar eficientment el mètode públic `poda_subarbre` fent servir diversos mètodes privats auxiliars que treballin directament amb dades de tipus `Node_arbre` i de tipus punter a `Node_arbre`. Heu de

- Escriure la capçalera, la precondició i la postcondició dels mètodes auxiliars.
- Implementar els mètodes auxiliars.
- Implementar el mètode públic `poda_subarbre` utilitzant un dels mètodes auxiliars.

Observeu que el mètode `poda_subarbre` ha d'alliberar la memòria de tots els nodes que s'eliminin del paràmetre implícit.

bool *poda_subarbre*(**int** *x*) {
/* Pre: El paràmetre implícit és un arbre binari A de enters. Els valors dels nodes de A són tots diferents. */
/* Post: Si x és el valor d'algun node de A, el resultat és cert i el paràmetre implícit és el resultat d'eliminar de A el node amb valor x i tots els seus descendents; altrament, el resultat és fals i el paràmetre implícit no varia (és a dir, és A). */

Mètodes auxiliars: especificació i implementació.