

# UQÀM Université du Québec à Montréal

## Programmation II (inf-2120)

Été 2013

### Examen final

23 juillet 2013

#### CONSIGNES

· Les règlements de l'UQAM concernant le plagiat seront strictement appliqués.

· Il est important de bien expliquer vos choix s'il y a lieu.

· Aucune documentation permise.

· La durée de l'examen est de 3 heures.

· Vous pouvez utiliser les versos comme brouillon ou comme espace supplémentaire.

· Il est interdit de dégrafer le questionnaire.

· Les téléphones cellulaires, calculatrices, ordinateurs, palm, baladeurs, iPods, etc. sont interdits.

#1 \_\_\_\_\_ / 17

#2 \_\_\_\_\_ / 20

#3 \_\_\_\_\_ / 18

#4 \_\_\_\_\_ / 12

\_\_\_\_\_  
TOTAL

\_\_\_\_\_ / 67

#### IDENTIFICATION

NOM : \_\_\_\_\_

PRÉNOM : \_\_\_\_\_

CODE PERMANENT : \_\_\_\_\_

SIGNATURE : \_\_\_\_\_

GROUPE : \_\_\_\_\_

PROFESSEUR : \_\_\_\_\_

commentaire :

**Numéro 1. (17 pts)**

**Objectif(s) :**

- Application des connaissances.
- Tris

Soit le tableau suivant :

11	8	6	3	10	1	9	5	7	2	14	4	15	13	12
----	---	---	---	----	---	---	---	---	---	----	---	----	----	----

Pour les sous-questions a), b) et c), donnez l'état (les valeurs) du tableau après chacun des quatre premiers tours de la boucle principale pour chacun des tris mentionnés (ordre croissant).

a) (5 pts) Insertion

[illegible]

**b) (5 pts) Sélection**

[illegible]

c) (5 pts) Bulle

[illegible]

**d) (2 pts)** Appliquez l'algorithme de **partition** utilisé par Quicksort (ou tri rapide) sur le tableau suivant. Considérez le premier élément du tableau comme étant le pivot.

[illegible]

---

**Numéro 2. (20 pts)**

**Objectif(s) :**

- Synthèse.
- Interface graphique awt.

Expliquez le rôle de chacune des composantes suivantes (en un très court texte) :

**a) (4 pts)** Component (Ex : Button, Container, Label, ...)

**b) (4 pts)** Container (Ex : Panel, Window, ...)

**c) (4 pts)** Event (Ex : WindowEvent, ActionEvent, ...)

**d) (4 pts)** Layout (Ex : GridLayout, BorderLayout, ...)

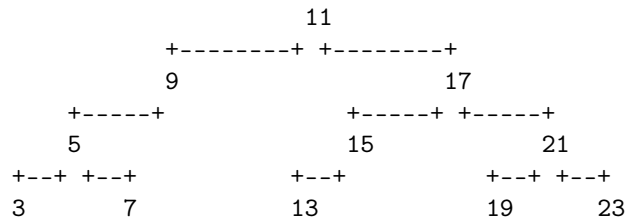
**e) (4 pts)** Listener (Ex : WindowListener, ActionListener, ...)

**Numéro 3. (18 pts)**

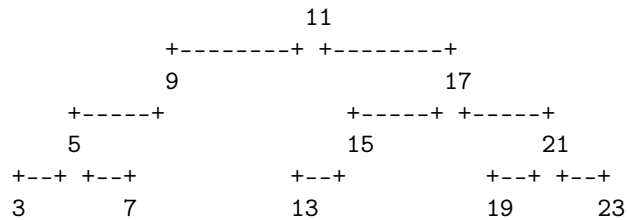
Objectif(s) :

- Application des connaissances.
- Arbre binaire de recherche.

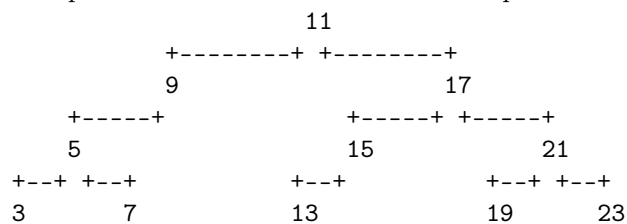
**a) (6 pts)** Ajouter la valeur 14 dans l'arbre binaire de recherche suivant :



**b) (6 pts)** Supprimez la valeur 13 dans l'arbre binaire de recherche suivant, tracez l'arbre résultant au complet. Au besoin utilisez le successeur pour faire le remplacement.



**c) (6 pts)** Supprimez la valeur 11 dans l'arbre binaire de recherche suivant, tracez l'arbre résultant au complet. Au besoin utilisez le successeur pour faire le remplacement.



**Numéro 4. (12 pts)**

Objectif(s) :

- Application des connaissances.
- Arbre binaire de recherche.
- Écriture de code.

Soit la déclaration d'une classe d'arbre binaire de recherche :

```
public class ABR<K extends Comparable<? super K>, V> {
    private class Noeud< Kn extends Comparable<? super Kn>, Vn > {
        public Kn clef;
        public Vn valeur;
        public Noeud<Kn,Vn> gauche;
        public Noeud<Kn,Vn> droite;
    }

    private Noeud<K,V> racine = null;
}
```

Le niveau d'un noeud dans un arbre est calculé comme suit : Le niveau de la racine est de 1. Le niveau d'un noeud est le niveau de son parent + 1. Écrivez le code d'une méthode de la classe **ABR** qui étant donné une clef calcule le niveau du noeud contenant cette clef. Considérez qu'il n'y a pas deux noeuds qui ont la même clef dans cette arbre. Si la clef n'est pas présente alors la méthode retourne la valeur 0.

```
public int niveau( K clef ) {
```