

# 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.

## IDENTIFICATION

NOM : \_\_\_\_\_  
PRÉNOM : \_\_\_\_\_  
CODE PERMANENT : \_\_\_\_\_  
SIGNATURE : \_\_\_\_\_  
GROUPE : \_\_\_\_\_  
PROFESSEUR : \_\_\_\_\_

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

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

#3 \_\_\_\_\_ / 12

#4 \_\_\_\_\_ / 18

#5 \_\_\_\_\_ / 12

\_\_\_\_\_  
TOTAL

\_\_\_\_\_ / 79

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**

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

**b) (5 pts) Sélection**

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

**c) (5 pts) Bulle**

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

**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.

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

**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, ...)

**Réponse :**

composante affichable et ayant une interaction possible avec l'utilisateur.

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

**Réponse :**

composante pouvant contenir d'autres composantes.

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

**Réponse :**

contient l'information décrivant une interaction.

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

**Réponse :**

décrit le mode de placement des composantes à l'intérieur d'un contenant.

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

**Réponse :**

fonction décrivant la réaction d'une composante à un événement.

---

**Numéro 3. (12 pts)**

Objectif(s) :

- Application des connaissances.
- Récursion

Écrivez le code correspondant à la fonction récursive suivante :

$$f(x, 0) = \frac{1}{x}$$

$$f(x, n) = x + \frac{1}{f(x, n-1)}$$

```
public class Recursion {
    public static double f( double x, int n ) {
        double resultat = 1 / x;

        if( 0 != n ) {
            resultat = x + 1 / f( x, n - 1 );
        }

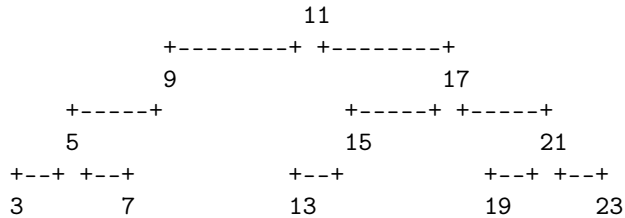
        return resultat;
    }
}
```

**Numéro 4. (18 pts)**

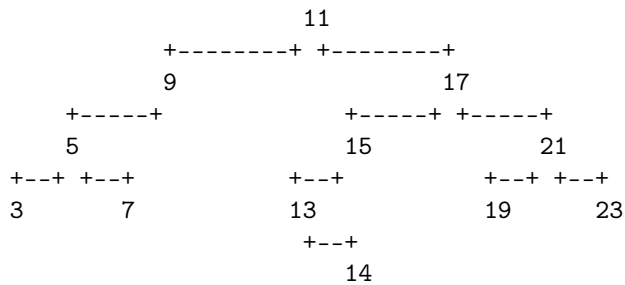
Objectif(s) :

- Application des connaissances.
- Arbre binaire de recherche.

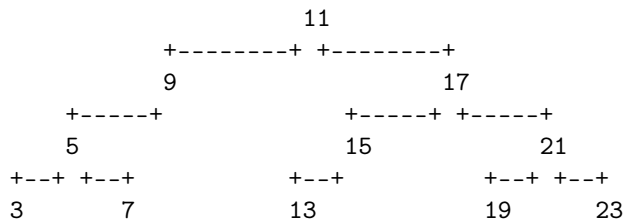
a) (6 pts) Ajouter la valeur 14 dans l'arbre binaire de recherche suivant :



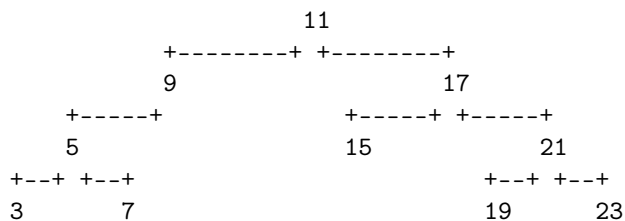
Réponse :



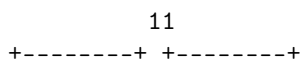
b) (6 pts) Supprimez la valeur 13 dans l'arbre binaire de recherche suivant, tracez l'arbre résultant au complet. Au besoin utilisez le successeur pour faire le remplacement.



Réponse :



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.



```

          9
        +-----+
          5
    +---+ +---+
    3      7

          17
        +-----+ +-----+
          15          21
    +---+          +---+ +---+
    13          19      23
    
```

Réponse :

```

          13
        +-----+ +-----+
          9          17
    +-----+          +-----+
          5          15          21
    +---+ +---+          +---+ +---+
    3      7          19      23
    
```

---

**Numéro 5. (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 semblable à celle vue en classe :

```
public class ABRNoeud< E extends Comparable< E > > {
    protected E valeur;
    protected ABRNoeud< E > gauche;
    protected ABRNoeud< E > droite;
}
```

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 `ABRNoeud` qui étant donné une valeur, calcule le niveau du noeud contenant cette valeur. Considérez qu'il n'y a pas deux noeuds qui ont la même valeur dans cet arbre. Si la valeur n'est pas présente alors la méthode retourne la valeur 0.

```
public int niveau( E cible ) {
    int resultat = 1;
    int direction = cible.compareTo( valeur );

    if( direction < 0 ) {
        resultat = ( gauche == null ) ? 0 : gauche.niveau( cible ) + 1;
    } else if( direction > 0 ) {
        resultat = ( droite == null ) ? 0 : droite.niveau( cible ) + 1;
    }

    return resultat;
}
```