

Examen final - mini révision en classe

1. Listes chainées

Dites ce que les méthodes suivantes affichent à la console :

A)

```
public static void traceListe () {  
  
    Maillon<Integer> m1 = new Maillon<Integer>  
                        (2, new Maillon<Integer>(4));  
    Maillon<Integer> m2 = new Maillon<Integer>(8, m1.suivant());  
    Maillon<Integer> m3 = new Maillon<Integer>(5);  
    m1.modifierSuivant(new Maillon<Integer>(7));  
    m2.suivant().modifierSuivant(m1.suivant());  
    Maillon<Integer> m4 = m2.suivant();  
    m4.suivant().modifierSuivant(m4);  
    m4.suivant().modifierInfo(6);  
    m3.modifierSuivant(m2.suivant().suivant());  
    m2.modifierSuivant(m3);  
    m1.modifierSuivant(m2);  
    m4.modifierSuivant(null);  
  
    while (m1 != null) {  
        System.out.print(m1.info() + " ");  
        m1 = m1.suivant();  
    }  
}
```

Réponse : _____

B)

```
public static void traceListe2() {  
  
    Maillon<Integer> m = new Maillon<>(5);  
    Maillon<Integer> m2 = m;  
    Maillon<Integer> m4 = new Maillon<>(9, new Maillon<>(8));  
    Maillon<Integer> m5 = m4.suivant();  
    Maillon<Integer> m3;  
  
    for (int i = 1; i < 5; i++) {  
        m2.modifierSuivant(new Maillon<>(i));  
        m2 = m2.suivant();  
    }  
  
    m3 = m.suivant();  
    m.modifierSuivant(m2);  
    m2 = m3.suivant();  
    m.suivant().modifierSuivant(m2.suivant());  
    m2.suivant().modifierSuivant(m2);  
    m3.modifierInfo(6);  
    m2.modifierSuivant(m3);  
    m3.modifierSuivant(m);  
    m5.modifierSuivant(new Maillon<>(7));  
    m5.suivant().modifierSuivant(m3);  
    m4 = new Maillon<>(10, m4);  
    m2.modifierSuivant(new Maillon<>(1));  
  
    while (m4 != null) {  
        System.out.print(m4.info() + " ");  
        m4 = m4.suivant();  
    }  
}
```

Réponse : _____

2. Techniques de recherche

Soit le tableau suivant :

```
[ 8, 11, 12, 15, 22, 31, 42, 44, 47, 50, 51, 56, 58, 59, 63, 65, 70, 72, 81, 87 ]
```

Donnez les éléments consultés lors de la recherche du nombre 13, en effectuant une :

- Recherche séq. non ordonnée :
- Recherche séq. ordonnée :
- Recherche binaire :

3. ABR

A) Dessinez l'arbre binaire de recherche construit par la suite d'instructions suivantes :

```
TdaAbr<Character> abr =
    new ArbreBinaireRecherche<Character>();
abr.inserer('F');
abr.inserer('D');
abr.inserer('I');
abr.inserer('E');
abr.inserer('G');
abr.inserer('J');
abr.inserer('A');
abr.inserer('B');
abr.inserer('C');
abr.inserer('H');
```

B) Dans l'arbre binaire construit en A), donnez la suite d'éléments obtenue en faisant le parcours :

- préfixe :
- infixé :
- postfixe :

C) Codez les méthodes suivantes qui retournent une chaîne de caractères contenant les éléments du parcours préfixe, infixé et postfixe.

```
public static String contenuPrefixe(Noeud<Character> racine) {
    //À FAIRE
}

public static String contenuInfixe(Noeud<Character> racine) {
    //À FAIRE
}

public static String contenuPostfixe(Noeud<Character> racine) {
    //À FAIRE
}
```

D) Codez la méthode récursive suivante :

```
/**  
 * Compte le nombre de feuilles de l'arbre donne par racine.  
 * @param racine la racine de l'arbre dont on veut calculer le nombre de  
 *               feuilles.  
 * @return le nombre de feuilles de l'arbre dont la racine est donne en  
 *         parametre.  
 */  
public static int nbrFeuilles (Noeud racine) {  
    //À FAIRE  
}
```

E) Dites en vos propres mots ce que fait la méthode suivante :

```
public static int mystere (Noeud<Integer> racine, int i) {  
    int res;  
    if (racine != null) {  
        if (!racine.info().equals(i)) {  
            if (racine.info().compareTo(i) < 0) {  
                res = mystere(racine.droite(), i);  
            } else {  
                res = mystere(racine.gauche(), i);  
            }  
            if (res != -1) {  
                res = res + 1;  
            }  
        } else {  
            res = 0;  
        }  
    } else {  
        res = -1;  
    }  
    return res;  
}
```

4. Tris

En regardant la première passe (en bleu) des tris suivants, dites de quel tri il s'agit (sélection, insertion, à bulles ou Quicksort) et complétez la deuxième passe.

A) Tri :

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
50	11	81	77	22	31	42	9	23	54	3	34	72	59	10	12	70	35	18
11	50	77	22	31	42	9	23	54	3	34	72	59	10	12	70	35	18	81

B) Tri :

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
50	11	81	77	22	31	42	9	23	54	3	34	72	59	10	12	70	35	18
3	11	81	77	22	31	42	9	23	54	50	34	72	59	10	12	70	35	18

C) Tri :

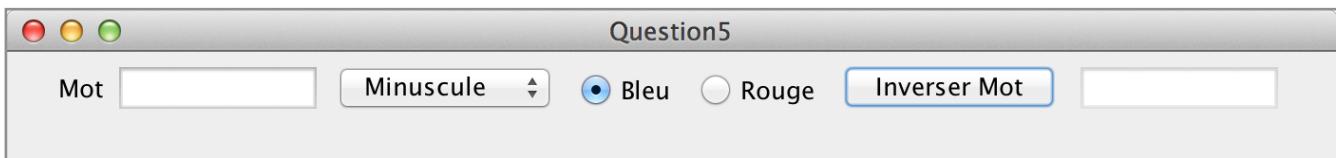
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
50	11	81	77	22	31	42	9	23	54	3	34	72	59	10	12	70	35	18
10	11	18	35	22	31	42	9	23	12	3	34	50	59	72	54	70	77	81

D) Tri :

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
50	11	81	77	22	31	42	9	23	54	3	34	72	59	10	12	70	35	18
11	50	81	77	22	31	42	9	23	54	3	34	72	59	10	12	70	35	18

5. Interface graphique

Créez la fenêtre suivante (taille 700 x 80 pixels) et ajoutez les fonctionnalités décrites ci-dessous.



Le champ Mot permet à l'utilisateur d'entrer une chaîne de caractères et est vide à l'initialisation de l'application. Par défaut, les caractères entrés dans ce champ sont bleus (Color.BLUE).

La liste déroulante contient les valeurs "Minuscule" et "Majuscule" et la valeur sélectionnée par défaut est "Minuscule". À la sélection d'une valeur dans cette liste, la chaîne dans le champ Mot se transforme en minuscules ou en majuscules, selon le choix fait.

Les boutons radio permettent de sélectionner la couleur des caractères entrés dans le champ Mot. Par défaut, le bouton sélectionné est Bleu. Au clic d'un de ces boutons, la couleur du texte dans le champ Mot change (s'il y a lieu) selon la couleur sélectionnée. Seulement un des deux boutons peut être sélectionné.

Le bouton Inverser Mot permet d'inverser le mot entré dans le champ Mot.

Au clic de ce bouton :

Si le champ Mot est vide, le champ à la droite du bouton Inverser Mot est effacé (s'il contenait quelque chose) et le message d'erreur suivant est affiché :



Si le champ contient une chaîne de caractères non vide, celle-ci est affichée, à l'envers, dans le champ à la droite du bouton Inverser Mot (en noir).

Le champ à droite du bouton Inverser Mot est actif, mais ne doit pas pouvoir être modifié.