

## Exercice 0 (4,5 points)

Répondez aux différentes questions sur la feuille annexe. Chaque question est sur 0,5 points

## Exercice 1 (6,5 points)

On s'intéresse aux arbres AVL.

A (1 pt.)- On considère l'arbre ci dessous. En rappelant les propriétés des arbres AVL, expliquez pourquoi il s'agit d'un arbre AVL, et décorez le sur la page fournie en annexe, en y portant les déséquilibres.



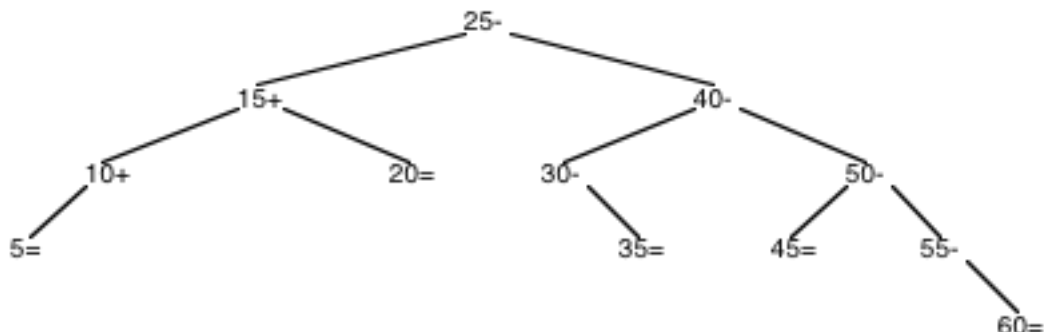
B (0,5 pt)- Rappelez ce qu'est l'opération d'adjonction dans un arbre AVL.

C (2pts.)- En partant de l'arbre AVL ci-dessus, donner les arbres obtenus en ajoutant successivement 50 puis 25 On s'attachera particulièrement à expliquer le raisonnement.

D (0,5pt.)- Donner la liste infixée de l'arbre donné en A, en justifiant votre raisonnement. Quelle propriété a-t-elle Est-ce toujours le cas

E (0,5pt.)- Rappelez ce qu'est l'opération de suppression.

F (1pt.)- Donner l'arbre obtenu par suppression de 20 dans l'arbre ci-dessous, en justifiant votre raisonnement.

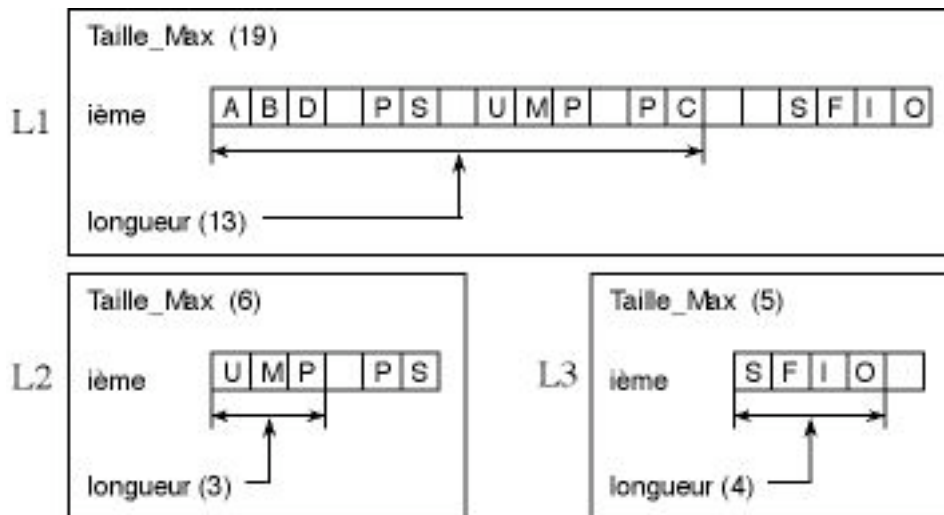


1pt. G- Donner l'arbre obtenu par suppression de 40 dans l'arbre donné dans la question F.

## Exercice 2 (9 points)

Les questions A.1, B.1 et C.1 sont des exemples qui sont là pour vous aider à comprendre ce qu'il faut faire dans les questions A.2, B.2 et C.2 où il faut écrire des programmes dans le langage de programmation de votre choix, compréhensible par le correcteur.

Un texte est une suite quelconque de caractères. Il peut être représenté par une liste contiguë de caractères. On se propose d'implanter quelques opérations spécifiques sur les textes. La spécification d'un paquetage correspondant en Ada est donné en fin d'exercice, pour information, étant entendu que vous pouvez utiliser le langage de programmation de votre choix, en utilisant les opérations sur les listes contiguës. Vous n'avez pas accès à la représentation elle-même. Les dessins des exemples néanmoins y feront référence, car plus explicites. En particulier, on dispose des trois listes suivantes.



A-On étudie l'opération Recherche, qui retourne la première position du texte De dans le texte Dans, s'il est présent et lève l'exception Erreur\_Specification s'il n'y est pas.

A.1 (1pt.)- Donner le résultat de Recherche (De => L2, Dans => L1), et celui de Recherche (De => L3, Dans => L1), en expliquant votre raisonnement.

A.2 (2pts.)- Proposer une implantation de l'opération Recherche. Evaluer la complexité.

B-On étudie l'opération Supprimer\_Les\_Espaces\_Inutiles, qui supprime les espaces qui suivent immédiatement un autre espace dans le texte Dans.

B.1 (0,5pt.)- Donner le résultat de Supprimer\_Les\_Espaces\_Inutiles (L1), en expliquant votre raisonnement.

B.2 (2pts.) - Proposer une implantation de l'opération `Supprimer_Les_Espaces_Inutiles`, Evaluer la complexité.

C- On étudie l'opération `Substituer`, qui remplace la première occurrence du texte `La_Chaine` dans le texte `Dans` par le texte `Par`, s'il est présent et lève l'exception `Erreur_Specification` s'il n'y est pas.

C.1 (1pt.) - Donner le résultat de `Substituer (La_Chaine => L2, Par => L3, Dans => L1)`, en expliquant votre raisonnement.

C.2 (2,5pts.) - Proposer une implantation de l'opération `Substituer`. Evaluer la complexité.

### *Spécification du paquetage en Ada.*

```
with Listes_Contigues;
package Les_Textes is
  type Texte (Taille_Max: Positive) is private;
  function Recherche (De, Dans: Texte) return Positive;
  procedure Substituer (La_Chaine: in Texte;
                       Par: in Texte;
                       Dans: in out Texte);
  procedure Supprimer_Les_Espaces_Inutiles
    (Dans: in out Texte);
private
  package L_C is new Listes_Contigues (Character);
  type Texte (Taille_Max: Positive) is
    new L_C.Liste (Taille_Max);
end Les_Textes;
```

# Annexe à rendre

Numéro de copie: \_\_\_\_\_

Utiliser cette feuille pour répondre aux questions ci-dessous.

1 Décorez l'arbre AVL suivant en y portant les déséquilibres (Exercice 1 Q.A).



2 Dans un B-arbre d'ordre 10, qui contient 100 000 valeurs, à combien estimez-vous le nombre d'accès disque pour rechercher un élément ?

- 10000     50     17     14     5

3 Dans un B-arbre d'ordre m, qui contient N valeurs, si on applique une recherche dichotomique sur les nœuds, combien y aura-t-il de comparaisons ?

- $N/m$       $m \log_m N$       $\log_2 N$       $\log_2 N/m$       $\log_m N$

4 Si on trie une liste contiguë, qui contient 100 000 valeurs, par un tri par tas, sur une machine dont le temps de base d'une comparaison ou d'un transfert est de l'ordre de la dizaine de microsecondes, à combien estimez-vous le temps de tri ?

- 10 jours     1 jour     1 heure     1 minute     1 seconde

5 Si on trie une liste contiguë, qui contient 100 000 valeurs, par un tri par la méthode de la bulle, sur une machine dont le temps de base d'une comparaison ou d'un transfert est de l'ordre de la dizaine de microsecondes, à combien estimez-vous le temps de tri ?

- 10 jours     1 jour     1 heure     1 minute     1 seconde

6 Quel est l'ordre de grandeur de la complexité au pire de l'opération qui teste si deux arbres binaires de recherche contiennent les mêmes éléments, en supposant qu'ils en contiennent au plus n ?

- $\log_2 n$       $n*(n-1)/2$       $n * \log_2 n$      n      $n^2$

7 Si on définit  $\text{debut}(x) = \text{cons}(\text{premier}(x), \text{listevide})$ , parmi les cinq égalités suivantes, quelle est celle qui est obligatoirement vraie ?

- concaténer(début(x), x)=concaténer(x, fin(x))
- début(fin(x))=x
- concaténer(début(x), fin(x))=x
- début(fin(x))=fin(début(x))
- fin(début(x))=x

8  Supposons une liste construite à l'aide des types suivants:

```

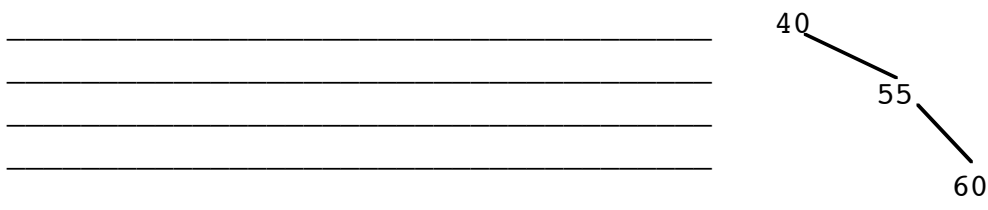
type Place; type Pt_Place is access Place;
type Place is record
  Contenu: Element;
  Suivant: Pt_Place;
end record;

```

Quel groupe d'instructions implante l'opération *insérer p après q*, où q repère un élément de la liste et p repère l'élément à insérer?

- Q.Suivant := P.Suivant; P.Suivant := Q;
- P.Suivant := Q; Q.Suivant := P.Suivant;
- P.Suivant := Q.Suivant; Q.Suivant := P;
- Q.Suivant := P; P.Suivant := Q.Suivant;
- Q.Suivant := P.Suivant; P.Suivant := Q.Suivant;

9  Ajouter 45 dans l'arbre binaire de recherche suivant, et expliquer.



10  On dispose d'un fichier de hachage qui contient 100000 éléments. Chaque bloc disque peut contenir 40 éléments. La fonction de hachage a pour résultat un nombre entre 1 et 1000, équitablement répartis. A combien estimez-vous le nombre d'accès disque pour rechercher un élément

- 1000
- 40
- 10
- 3
- 1