

Contrôle continu

Exercice 1 Listes récursives

Question 1.1 Écrire une fonction **Nbelements** qui, étant donné une liste d'entiers **L** passé en paramètre, renvoie le nombre d'éléments de la liste d'entiers **L**.

1. Donner le ou les cas d'arrêt
2. Expliquer le cas général.
3. Écrire la fonction **Nbelements(L)**
4. Dérouler votre fonction **Nbelement** sachant que $L = \langle 2, 2, 4, 4 \rangle$.

Question 1.2 Écrire une fonction **appartenance** qui renvoie vrai si une information **val** donnée se trouve dans une liste ordonnée (par ordre croissant) **L** passée en paramètre et faux sinon.

1. Donner le ou les cas d'arrêt
2. Expliquer le cas général.
3. Écrire la fonction **appartenance(val, L)**
4. Dérouler votre fonction **appartenance(val, L)** sachant que $L = \langle 2, 2, 4, 4, 7 \rangle$ et $val = 6$.

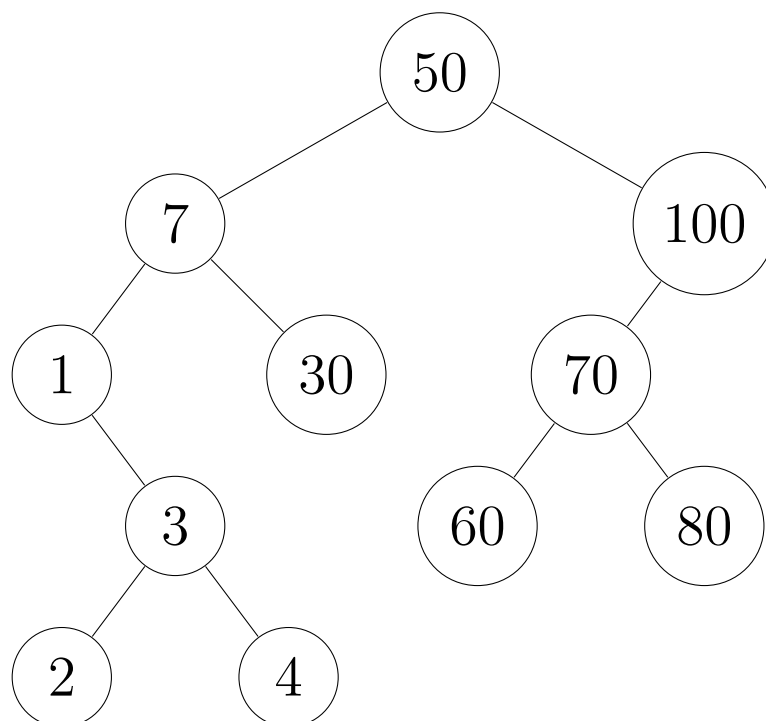
Question 1.3 Écrire une fonction **NbSup** qui, étant donné une liste d'entiers **L** d'entiers triée par ordre croissant, et un entier **val** passés en paramètre, renvoie le nombre d'information enregistrée la liste **L inférieures ou égales** à **val**. Nous supposons que la liste ne peut contenir des valeurs identiques.

Si $L = \langle 2, 2, 4, 4, 5, 6, 7 \rangle$, la fonction **NbSup(L, 4)** devra retourner 4.

1. Donner le ou les cas d'arrêt
2. Expliquer le cas général.
3. Écrire la fonction **NbSup(L, val)**.
4. Dérouler votre fonction **NbSup** sachant que $L = \langle 2, 2, 4, 4, 5, 6, 7 \rangle$ et $val = 4$.

Exercice 2 Arbres binaires

Soit l'arbre binaire suivant avec 50 comme racine :



Question 2.1 Exprimer

1. le père, le ou les fils du sommet 50
2. le père, le ou les fils du sommet 7

Question 2.2 Donner tous les feuilles de l'arbre.

Question 2.3 Donner la taille et la hauteur de l'arbre.

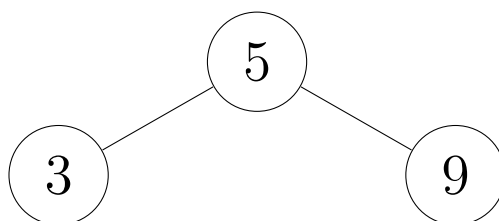
Question 2.4 Est-il un arbre binaire de recherche? Justifier votre réponse.

Exercice 3 Arbres binaires de recherche

L'objectif de cet exercice est d'afficher les valeurs contenues par les sommets de l'arbre binaire de recherche de façon croissante. Nous utiliserons la procédure **Ecrire(info)** qui affiche l'information **info**.

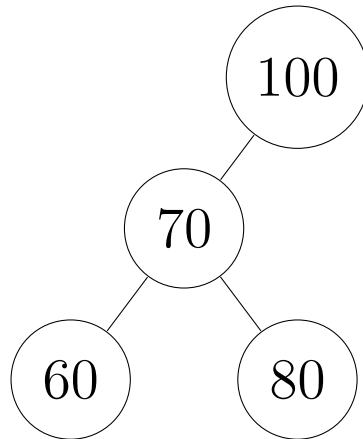
Question 3.1 Donner l'affichage que retournerait la procédure **afficheOrdre(unArbre)**.

1. si **unArbre** est vide;
2. si **unArbre** est feuille;
3. si **unArbre** est l'arbre de la figure dessous :



Question 3.2 Écrire la procédure **afficheOrdre(unArbre)**.

Question 3.3 Dérouler votre fonction sur l'exercice de la figure suivante :

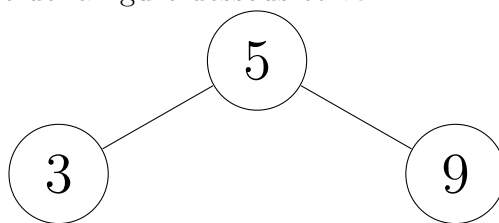


Exercice 4 Encore des arbres binaires de recherche

L'objectif de cet exercice est d'écrire une fonction **SommeElement** qui, étant donné un arbre binaire de recherche **A** et une valeur **val** passé en paramètre, retourne la somme des valeurs contenues dans l'arbre ayant une valeur inférieure à **val**. Nous supposons que l'arbre ne contient pas deux valeurs identiques.

Question 4.1 Donner la valeur que retournerait la fonction **SommeElement(unArbre, val)**.

1. si **unArbre** est vide ;
2. si **unArbre** est feuille ;
3. si **unArbre** est l'arbre de la figure dessous et **val** = 5.
4. si **unArbre** est l'arbre de la figure dessous et **val** = 1.



Question 4.2 Écrire une fonction **SommeElement(unArbre, val)**