Rapport de TP3 AI01
ABOU GHALIA Mostafa et MARRAS Hugo

On note le nombre d'éléments qui compose la chaîne de caractère str h la hauteur de l'arbre et n le nombre d'éléments dans l'arbre

Fonctions demandées:

1. motValide:
   Entrée : chaine de caractère
   Sortie : Entier
   Complexité : **O**(m) et **Ω**(1)
2. creerNoeud:
   Entrée : chaine de caractère
   Sortie : Entier
   Complexité : **Θ**(1)
3. creerNoeudVide:
   Entrée : une chaine de caractère
   Sortie : un noeud
   Complexité : **Θ**(n)
4. min:
   Entrée : deux entier
   Sortie : un entier
   Complexité : **Θ**(1)
5. max:
   Entrée : deux entier
   Sortie : un entier
   Complexité : **Θ**(1)
6. compare:
   Entrée : deux pointeurs de chaine de caractere
   Sortie : un entier
   Complexité : **Θ**(1)
7. ajouterArbre:
   Entrée : un pointeur vers un arbre et un pointeur vers un noeud
   Sortie : un arbre
   Complexité : **Θ**(1)
8. ajout_noeud_rec:
   Entrée : deux pointeurs vers un noeud
   Sortie : rien
   Complexité : **Θ**(h)
9. ajout_noeud:
   Entrée : un pointeur vers un noeud et un pointeur vers une chaine de caractere
   Sortie : un noeud
   Complexité : **Θ**(h)
10. enlever_occurence_rec:
    Entrée : un pointeur vers un noeud et un pointeur vers une chaine de caractere
    Sortie : un noeud
    Complexité : **Θ**(h)

11. enlever_occurence:
    Entrée : un pointeur vers un noeud et un pointeur vers une chaine de caractere
    Sortie : pointeur vers un noeud
    Complexité : **Θ**(h)
12. check_parfait:
    Entrée : un pointeur vers un arbre
    Sortie : un entier
    Complexité : **Θ**(h)
13. enlever_noeud:
    Entrée : un pointeur vers un noeud et un pointeur vers une chaine de caractere
    Sortie : pointeur vers un noeud
    Complexité : **Θ**(n)
14. nettoyage:
    Entrée : un pointeur vers un noeud
    Sortie : pointeur vers un noeud
    Complexité : **Θ**(n)
15. display_arbre:
    Entrée : un pointeur vers un arbre
    Sortie : rien
    Complexité : **Θ**(n)
16. display_noeud:
    Entrée : un pointeur vers un noeud et un pointeur vers une chaine de caractere
    Sortie : rien
    Complexité : **Θ**(n)
17. node_height:
    Entrée : un pointeur vers un noeud
    Sortie : un entier
    Complexité : **Θ**(h)
18. minuscule:
    Entrée : un pointeur vers une chaine de caractere
    Sortie : rien
    Complexité : **Θ**(1)
19. check_equilibre:
    Entrée : un pointeur vers un noeud
    Sortie : un entier
    Complexité : **Θ**(1)
20. pack_list:
    Entrée : un pointeur vers un arbre
    Sortie : une liste
    Complexité : **Θ**(n)

21. pack_list_rec:
    Entrée : un pointeur vers une liste
    Sortie : rien
    Complexité : **Θ**(n)
22. display_list:
    Entrée : un pointeur vers un arbre et un pointeur vers une liste
    Sortie : rien
    Complexité : **Θ**(n)
23. initialize_list:
    Entrée : un pointeur vers une liste
    Sortie : rien
    Complexité : **Θ**(1)
24. insert_end_list:
    Entrée : un pointeur vers une liste, une chaine de caractere et un entier
    Sortie : rien
    Complexité : **Θ**(n)
25. similarite:
    Entrée : un pointeur vers une liste, une chaine de caractere et un entier
    Sortie : float
    Complexité : **Θ**(n^2)
26. supprimerListe:
    Entrée : un pointeur vers une liste,
    Sortie : rien
    Complexité : **Θ**(n)
27. node_print:
    Entrée : un pointeur vers une noeud, deux entiers
    Sortie : rien
    Complexité : **Θ**(n)
28. avl_print:
    Entrée : un pointeur vers un arbre
    Sortie : rien
    Complexité : **Θ**(n)
29. initialize_arbre:
    Entrée : un pointeur vers un arbre
    Sortie : rien
    Complexité : **Θ**(1)