Arbres

On utilisera les deux types suivants :

```
type 'a ab = Vide | N of 'a * 'a ab * 'a ab;;
type ('a,'b) abs = F of 'a | NI of 'b * ('a,'b) abs * ('a,'b) abs;;
```

qui représentent respectivement un arbre binaire et un arbre binaire strict.

Exercice 1 :

1. Écrire des fonctions `complet_ab` et `complet_abs` prenant en argument un entier n et un élément e et renvoyant un arbre de hauteur n complet dont les nœuds ont tous l'étiquette e, suivant le type utilisé

   Combien de nœuds a un tel arbre ?
2. Écrire une fonction `complet2_abs` prenant en argument un entier n renvoyant un arbre de hauteur n complet dont les nœuds sont numérotés de la façon suivante :
   — la racine porte le numéro 1 ;
   — si un nœud interne porte le numéro n, alors son fils gauche porte le numéro $2n$ et son fils droit le numéro $2n+1$.

Exercice 2 :

1. Définir un type qui représente les arbres généraux (non forcément binaires) dont les nœuds sont de même type.
2. Écrire une fonction `nb_feuilles` renvoyant le nombre de feuilles d'un arbre général.
3. Écrire une fonction `hauteur` renvoyant la hauteur d'un arbre général.

Exercice 3 :

1. Écrire une fonction `max_ab` (resp. `max_ab`) prenant en argument un arbre binaire (resp. strict) et renvoyant la valeur maximale de ses nœuds
2. Écrire des fonctions `list_prof_ab` (resp. `list_prof_abs`) prenant en argument un arbre binaire (resp. strict) et une valeur renvoyant la liste des profondeurs des nœuds où cette valeur est atteinte.

Exercice 4 :

1. Écrire une fonction `generation` prenant en argument un arbre binaire et un entier n et qui renvoie la liste des étiquettes des nœuds de profondeur n.
2. En déduire une fonction qui affiche le parcours en profondeur d'un arbre binaire d'entiers. Quelle est sa complexité ?

Exercice 5 : Écrire une fonction `somme_profondeurs` prenant en argument un arbre binaire et renvoyant la somme des profondeur de ses feuilles.

Exercice 6 : Le nombre de Strahler d'un arbre binaire strict est défini ainsi :
— le nombre de Strahler d'une feuille est égal à 1 ;
— si i et j désignent les nombres de Strahler des fils gauche et droit d'un nœud, alors le nombre de Stralher de ce dernier sera égal à $i+1$ si $i=j$ et à $\max(i,j)$ sinon.

1. Définir un type d'arbre binaire strict adapté au calcul du nombre de Strahler.
2. Écrire une fonction `strahler` renvoyant le nombre de Strahler d'un arbre binaire strict.
3. Quels sont les arbres de hauteur n maximisant le nombre de Strahler ?
4. Quels sont les arbres de hauteur n minimisant le nombre de Strahler ? Ces derniers sont dits filiforme.
5. Combien y-a-t-il d'arbres filiformes de hauteur n ?
6. Écrire une fonction `liste_filiforme` qui prend en paramètre un entier n et renvoie la liste des arbres filiformes de hauteur n.
7. Écrire une fonction `filiforme` qui prend en paramètre une liste de caractères $'G'$ et $'D'$ et qui renvoie l'arbre filiforme correspondant, le k-ème terme de la liste correspondant à la position de la feuille à la k-ème génération.
8. Écrire la fonction inverse de `filiforme`.
9. On définit la suite $(A_n)_{n\in\mathbb{N}}$ des arbres de Fibonacci de la façon suivante
   — A_0 et A_1 sont des feuilles ;
   — pour tout $n \geq 2$, A_n a comme fils gauche A_{n-1} et comme fils droit A_{n-2}
   Écrire une fonction `Fibonacci` qui prend en paramètre un entier n et renvoie A_n. Déterminer le nombre de Strahler de A_n en fonction de n.

Exercice 7 :

1. Écrire une fonction `feuilles` qui prend en paramètre un arbre et qui renvoie la liste de ses feuilles dans l'ordre d'un parcours en profondeur préfixe (ou gauche).
2. Quelle est la complexité, en fonction du nombre de feuilles, de cette fonction ?
3. Si ce n'est pas le cas, la modifier pour que la complexité soit linéaire.