

1 Petit échauffement ...

1.1 Egalité

Ecrire la fonction `egal(L1, L2)` qui renvoie 1 si les deux listes sont égales et 0 sinon. Par exemple:

```
egal([1,2,3], [1,2,3]) = 1
egal([1,2], [1,2,3]) = 0
```

1.2 Palindrome

Ecrire la fonction `palindrome(L)` qui renvoie 1 si L est un palindrome et 0 sinon. Par exemple:

```
palindrome([1,2,1]) = 1
palindrome([1,2,3]) = 0
```

2 Faire le tri !

2.1 Réduire la liste

Ecrire la fonction `reduire(L, N, M)` qui réduit la liste aux éléments indicés de N à M inclu. Par exemple:

```
reduire([3,4,5,6], 2, 4) = [4, 5, 6]
```

2.2 Trier

Ecrire la fonction `trier(L)` qui effectue un tri par fusion. Le tri va d'abord couper la liste en deux, trier les deux parties récursivement et enfin les fusionner. Par exemple:

```
trier([3,1,2]) = [1, 2, 3]
```

3 Définissons les ensembles !

3.1 Etre un ensemble

Ecrire la fonction `etre_ensemble(L)` qui renvoie 1 si la liste est triée et qu'elle ne contient aucun duplicata et renvoie 0 sinon. Par exemple:

```
etre_ensemble([1, 2, 3]) = 1
etre_ensemble([3, 2, 1]) = 0
etre_ensemble([1, 2, 2]) = 0
```

3.2 Union d'ensembles

Ecrire la fonction `union(L1, L2)` qui renvoie l'union de l'ensemble L1 et L2.

Par exemple:

```
union([1, 2, 3], [4]) = [1, 2, 3, 4]
```

3.3 Intersection d'ensembles

Ecrire la fonction `inter(L1, L2)` qui renvoie l'intersection de l'ensemble L1 et L2. Par exemple:

```
inter([1, 2, 3], [4]) = []  
inter([1, 2, 3], [3]) = [3]
```

3.4 Différence d'ensembles

Ecrire la fonction `diff(L1, L2)` qui renvoie la différence de l'ensemble L1 et L2. Par exemple:

```
diff([1, 2, 3], [4]) = [1, 2, 3]  
diff([1, 2, 3], [3]) = [1, 2]
```

4 Question de transition ...

Ecrire la fonction `parcourir(A)` qui renvoie la liste des noeuds de l'arbre (en notation postfixée). Par exemple:

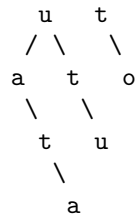
```
      +  
     / \  
A =  -  1  
     / \  
    10 3
```

```
parcourir(A) = ['+', '-', 10, 3, 1]
```

5 Besoin d'un dictionnaire ?

Le but de cet exercice est d'implémenter un dictionnaire représenté sous forme d'arbre binaire. Un mot est encodé par un chemin dans l'arbre. Par exemple, si notre dictionnaire contient les mots **toto**, **tata** et **tutu**:

```
  A =  t  
       \  
      o  
     /\
```



5.1 Chercher un mot

Ecrire la fonction `chercher(A, L)` qui renvoie 1 si le mot est présent dans `A` et 0 sinon. Par exemple:

```
chercher(A, ['c', 'o', 'u', 'c', 'o', 'u']) = 0
chercher(A, ['t', 'o', 't', 'o']) = 1
```

5.2 Insérer un mot

Ecrire la fonction `insérer(A, L)` qui va entrer le mot `L` dans le dictionnaire. (Il peut être utile d'écrire une fonction auxiliaire qui crée un arbre à partir d'un mot). Par exemple:

```
insérer(A, ['t', 'u', 't', 'o']) = A'
avec
```

