Récursivité : parcours d'arbres binaires

Questions : On suppose que les fonctions `infix` et `postfix` affichent la suite des nœuds d'un arbre binaire (`[racine,gauche,droite]`) respectivement dans un ordre infixé et postfixé.

```python
def infix(t) :
  if t != [] :
    infix(t[1])
    print(t[0],end=' ')
    infix(t[2])
  else :
    print(0,end=' ')
  return
```

```python
def postfix(t) :
  if t != [] :
    postfix(t[1])
    postfix(t[2])
    print(t[0],end=' ')
  else :
    print(0,end=' ')
  return
```

Qu'affichent les appels suivants ?

Réponses :

1. `>>> postfix([1, [], [2, [], [3, [5, [], []], [4, [], []]]]])`
   `0 0 0 0 5 0 0 4 3 2 1`
2. `>>> postfix([1, [3, [5, [], []], []], [2, [], [4, [], []]]])`
   `0 0 5 0 3 0 0 0 4 2 1`
3. `>>> infix([6, [4, [2, [], []], []], [3, [], [1, [], []]]])`
   `0 2 0 4 0 6 0 3 0 1 0`
4. `>>> postfix([2, [4, [], []], [1, [], [6, [], [3, [], []]]]])`
   `0 0 4 0 0 0 0 3 6 1 2`
5. `>>> postfix([1, [2, [4, [], []], [3, [5, [], []], []]], []])`
   `0 0 4 0 0 5 0 3 2 0 1`
6. `>>> postfix([1, [3, [5, [], []], [4, [], []]], [2, [], []]])`
   `0 0 5 0 0 4 3 0 0 2 1`
7. `>>> postfix([2, [4, [], [6, [], []]], [1, [3, [], []], []]])`
   `0 0 0 6 4 0 0 3 0 1 2`
8. `>>> infix([1, [2, [4, [], []], [3, [5, [], []], []]], []])`
   `0 4 0 2 0 5 0 3 0 1 0`
9. `>>> infix([2, [4, [], [6, [], []]], [1, [3, [], []], []]])`
   `0 4 0 6 0 2 0 3 0 1 0`
10. `>>> infix([5, [3, [1, [], []], []], [4, [], [2, [], []]]])`
    `0 1 0 3 0 5 0 4 0 2 0`
11. `>>> infix([1, [], [2, [], [3, [5, [], []], [4, [], []]]]])`
    `0 1 0 2 0 5 0 3 0 4 0`
12. `>>> postfix([2, [4, [], []], [1, [], [6, [], [3, [], []]]]])`
    `0 0 4 0 0 0 0 3 6 1 2`
13. `>>> postfix([1, [3, [], [5, [], []]], [2, [4, [], []], []]])`
    `0 0 0 5 3 0 0 4 0 2 1`

14. `>>> infix([2, [4, [], []], [1, [], [6, [], [3, [], []]]]])`
    `0 4 0 2 0 1 0 6 0 3 0`
15. `>>> infix([1, [3, [], [5, [], []]], [2, [4, [], []], []]])`
    `0 3 0 5 0 1 0 4 0 2 0`
16. `>>> postfix([2, [4, [6, [], []], []], [1, [], [3, [], []]]])`
    `0 0 6 0 4 0 0 0 3 1 2`
17. `>>> infix([2, [4, [], []], [1, [], [6, [], [3, [], []]]]])`
    `0 4 0 2 0 1 0 6 0 3 0`
18. `>>> postfix([5, [3, [1, [], []], []], [4, [], [2, [], []]]])`
    `0 0 1 0 3 0 0 0 2 4 5`
19. `>>> infix([1, [3, [5, [], []], []], [2, [], [4, [], []]]])`
    `0 5 0 3 0 1 0 2 0 4 0`
20. `>>> infix([2, [], [1, [4, [], []], [6, [], [3, [], []]]]])`
    `0 2 0 4 0 1 0 6 0 3 0`
21. `>>> infix([1, [3, [5, [], []], [4, [], []]], [2, [], []]])`
    `0 5 0 3 0 4 0 1 0 2 0`
22. `>>> postfix([6, [4, [2, [], []], []], [3, [], [1, [], []]]])`
    `0 0 2 0 4 0 0 0 1 3 6`
23. `>>> postfix([2, [], [1, [4, [], []], [6, [], [3, [], []]]]])`
    `0 0 0 4 0 0 0 3 6 1 2`
24. `>>> infix([2, [4, [6, [], []], []], [1, [], [3, [], []]]])`
    `0 6 0 4 0 2 0 1 0 3 0`