

Exercice 1 : Programmation – Récursivité

1.a. La fonction A est récursive car elle s'appelle elle-même sur la dernière ligne de la fonction.

1.b. Cette fonction récursive peut s'appeler indéfiniment si le résultat de choice est toujours False . Python fixe une profondeur de récursivité maximale de 1000 pour s'assurer que le programme s'arrête, autrement dit, Python lève une erreur si cette limite est atteinte lorsque choice renvoie False plus de 1000 fois .

2.a.

```
def A(n):
    if (n ≤ 0) or choice([True, False]):
        return "a"
    else:
        return "a" + A(n - 1) + "a"
```

2.b. Si nous appelons la fonction A(n) avec $n < 50$, nous serons confrontés à l'un de ces cas :

- Soit elle se terminera par un choix aléatoire "Vrai" renvoyé par choice([True, False]).
- Soit elle se terminera par le maximum de répétitions récursives possibles avec A(n - 1)

jusqu'à A(0).

- 3.
- B(0) → "bab"
 - B(1) → "bab", "bbabb"
 - B(2) → "bab", "baaab", "bbabb", "bbbabbb"

4.a.

```
def regleA(n):
    n = len(chaine)
    if (n ≥ 2):
        return (chaine[0] == "a") and (chaine[n - 1] == "a") and
                regleA(chaine[1:n - 1])
    else:
        return chaine == "a"
```

4.b.

```
def regleB(n) :
    n = len(chaine)
    if (n ≥ 2) :
        return (chaine[0] == "b") and (chaine[n - 1] == "b") and
                (regleA(chaine[1:n - 1]) or regleB(chaine[1:n - 1]))
    else :
        return False
```