

Exercices de révisions

1 Exponentiation rapide recursive

Pour calculer une puissance entière de x , on peut utiliser les propriétés :

$$x^n = \begin{cases} 1 & \text{si } n = 0 \\ x & \text{si } n = 1 \\ (x^2)^p & \text{si } n = 2p \text{ est pair} \\ x (x^2)^p & \text{si } n = 2p + 1 \text{ est impair} \end{cases}$$

On en déduit une fonction recursive de calcul de puissance :

```
def power(x, n):
    if n == 0:
        return(1)
    elif n == 1:
        return(x)
    y = power(x * x, n // 2)
    if n % 2 == 0:
        return(y)
    else:
        return(x * y)
```

Q1 : Rédiger les assertions nécessaires au bon fonctionnement de cette fonction.

Q2 : Décrire l'évolution de l'état de la mémoire lors de l'appel `power(2., 5)` (on pourra rédiger la réponse à cette question sous la forme d'une pile d'appel récursif).

2 Suite de Fibonacci

On considère le programme suivant :

```
def funct(n) :
    if n <=1 :
        return(1)
    else:
        return(funct(n-1) + funct(n - 2))

print(funct(5))
```

Q1 : Quel est le résultat de ce programme ?

Q2 : Combien de fois la fonction est-elle appelée ?