

Exercice 1. — Donner une chaîne de documentation à la fonction `division_euclidienne` suivante.

```
def division_euclidienne(a, b):  
    q = 0  
    r = a  
    while r >= b:  
        q = q + 1  
        r = r - b  
    return q, r
```

Exercice 2. — Donner une chaîne de documentation pour une fonction `puissance(x, n)` qui calcule x à la puissance n pour deux entiers x et n .

Exercice 3. — Donner un invariant de boucle pour la fonction suivante qui calcule x à la puissance n .

```
def puissance(x, n):  
    r = 1  
    for i in range(n):  
        r = r * x  
    return r
```

Exercice 4. — Écrire des tests pour la fonction `puissance` de l'exercice précédent.

Exercice 5. — Pour la fonction suivante, lui donner un meilleur nom, une chaîne de documentation, un invariant de boucle et des tests.

```
def f(t):  
    s = 0  
    for i in range(len(t)):  
        s += t[i]  
    return s
```

Exercice 6. — On prétend que la fonction suivante teste l'appartenance de la valeur v au tableau t .

```
def appartient(v, t):  
    i = 0  
    while i < len(t) - 1 and t[i] != v:  
        i = i+1  
    return i < len(t)
```

Donner des tests pour cette fonction, et en particulier un test montrant qu'elle est incorrecte.

Exercice 7. — On prétend que la fonction suivante teste l'appartenance de la valeur v au tableau t .

```
def appartient(v, t):  
    for i in range(len(t)):  
        if t[i] == v:  
            trouvee == False  
        else:  
            trouvee == True  
    return trouvee
```

Donner des tests pour cette fonction, et en particulier des tests montrant plusieurs raisons pour laquelle cette fonction est incorrecte.