

## TD: Preuves d'algorithmes

### Exercice 1

On donne la fonction ci-dessous codée en Python qui retourne  $x^n$  pour  $n$  entier positif et  $x$  un nombre.

```
def puissance(x,n):  
    p = 1  
    i = 0  
    while i < n:  
        p = p * x  
        i = i + 1  
    return p
```

1. Montrer que la propriété  $p = x^i$  est un invariant de boucle.
2. En déduire la correction partielle de cette fonction.
3. Montrer que  $n - i$  est un variant de boucle.
4. En déduire la terminaison de cette fonction.

### Exercice 2

On donne la fonction ci-dessous codée en Python où  $t$  est une liste de nombres.

```
def f(t):  
    s = 0  
    i = 0  
    while i < len(t):  
        s = s + t[i]  
        i = i + 1  
    return s
```

1. Que fait cette fonction? Donner-lui un meilleur nom.
2. Proposer un jeu de tests.
3. Donner un invariant de boucle.
4. Donner un variant de boucle.

### Exercice 3

Cette fonction retourne la valeur du plus grand élément de la liste t.

```
def maximum(t):  
    M = t[0]  
    for i in range(1, len(t)):  
        if t[i] > M:  
            M = t[i]  
    return M
```

1. Donner un invariant de boucle.
2. Justifier que c'est bien un invariant de boucle.
3. En déduire la correction partielle de l'algorithme.

### Exercice 4

La fonction suivante retourne un booléen qui indique si l'élément elem est dans la liste t.

```
def appartient(elem,t):  
    trouvee = False  
    i = 0  
    while i < len(t) and trouvee != True:  
        if t[i] == elem:  
            trouvee = True  
        i = i + 1  
    return trouvee
```

1. Donner un variant de boucle.
2. Justifier la terminaison de cette fonction.
3. Donner un invariant de boucle.