Fiche

python™

Exercice 1 – Somme des éléments d'une liste Soit la fonction suivante.

```
def somme(L,x):
    """ La fonction prend en entrée une liste L de flottants ou d'entiers,
    et retourne la somme de ses éléments. """
    s=0
    for i in range(len(L)):
        #Inv(i): s est la somme des i premiers éléments de la liste.
        s+=L[i]
        #Inv(i+1): s est la somme des i+1 premiers éléments de la liste.
    return s
```

Question 1 *Montrer la terminaison et la correction de la fonction* `somme`

Exercice 2 – Maximum d'une liste Soit la fonction suivante.

```
def maximum(L):
    """ La fonction prend en entrée une liste L non vide de flottants ou d'entiers,
    et retourne le maximum de ses éléments. """
    m=L[0]
    for i in range(1,len(L)):
        #Inv(i): m est le plus grand élément de L[0:i].
        if L[i]>m:
            m=L[i]
        #Inv(i+1): m est le plus grand élément de L[0:i+1].
    return m
```

Question 1 *Montrer la terminaison et la correction de la fonction* `maximum`

Exercice 3 – Recherche d'un élément dans une liste Soit la fonction suivante.

```
def recherche(L,x):
    """ La fonction prend en entrée une liste L et un élément x,
    et retourne True si x est dans L, False sinon."""
    for i in range(len(L)):
        if L[i]==x:
            return True
    return False
```

Question 1 *Montrer la correction de la fonction* `recherche`.

Exercice 4 – Recherche ddichotomique dans une liste triée Soit la fonction suivante.

```
def recherche_dicho(L,x):
    """ La fonction prend en entrée une liste L triée dans l'ordre croissant et un élément x,
    et retourne True si x est dans L, False sinon."""
    g=0
    d=len(L)
```

```
    while g<d:
        #Inv: x ne se trouve ni dans L[0:g] ni dans L[d:len(L)].
        m=(g+d)//2
        if L[m]==x:
            return True
        elif L[m]<x:
            g=m+1
        else:
            d=m
        #Inv: x ne se trouve ni dans L[0:g] ni dans L[d:len(L)].
    return False
```

Question 1 *Montrer la correction de la fonction* `recherche_dicho`.

Exercice 5 – Factorielle

$n!$ On donne l'algorithme suivant.

```
for i in range(1,n+1):
    # en entrant dans le ième tour de boucle, p = (i-1)!
    p=p*i
    # en sortant du ième tour de boucle, p = i!
```

Question 1 *Montrer que l'algorithme précédent permet de calculer $n!$.*

Exercice 6 – L'algorithme d'Euclide

`https://lgarcin.github.io/CoursPythonCPGE/preuve.html`

```
def pgcd(a, b):
  while b!= 0:
      a, b = b, a % b
  return a
```

Question 1 *Monter la terminaison de la fonction* `pgcd(a,b)`.

Question 2 *Monter la correction de la fonction* `pgcd(a,b)`.

On donne une seconde version de l'algorithme d'Euclide. Pour cela on effectue la division euclidienne de a par b où $a$et b sont deux entiers strictement positifs. Il s'agit donc de déterminer deux entiers q et r tels que $a = bq + r$ avec $0 \leq r < b$. Voici un algorithme déterminant q et r :

```
q = 0
r = a
while r >= b :
    q = q + 1
    r = r -b
```

On choisit comme invariant de boucle la propriété $a = bq + r$.

Question 3 *Montrer la correction de l'algorithme.*

Exercice 7 – Multiplication

L'objectif est de calculer le produit de deux nombres entiers positifs a et b sans utiliser de multiplication.

```
p = 0
m = 0
while m < a :
    m = m + 1
    p = p + b
```

Question 1 *Montrer la terminaison de l'algorithme.*

Question 2 *Proposer une propriété d'invariance.*

Question 3 *Montrer la correction de l'algorithme.*

Exercice 8 – Exponentiation rapide

```
def expo_rapide(x,n):
    y = x
    z = 1
    m = n
    # Invariant : x*y^m = x^n
    while m>0:
        q,r=m//2,m%2
        if r==1:
            z=z*y
        y=y*y
        m=q
    return z
```

Question 1 *Montrer que la fonction permet de calculer x^n.*