Série N°4 (TD)
(Complexité et Preuve des algorithmes)

Exercice 1 (Notation Landau et Complexité)

1. Montrer que :	2. Déterminez l'ordre de grandeur asymptotique des suites suivantes :
a. $n^2=O(10^{-5}n^3)$	a. $30n^2+5n+10$
b. $25n^4-19n^3+13n^2=O(n^4)$	b. $Log(3n)$
c. $2^{n+100}=O(2^n)$	c. $3n^3+7n^2log(n)-9$
d. $n^2=o(10^{-5}n^3)$	d. $Log(n^2)$
e. $30n^4-19n^3+13n^2=\Theta(30n^4)$	e. $n+1/n$
f. $2^{n+100}=\Theta(2^n)$	

Exercice 2 (Complexité des algorithmes itératives)

Déterminez l'expression du temps *T(n)* en fonction de la taille des données *n*, puis en déduisez la complexité temporelle dans le pire des cas pour chaque portion de code suivante.

Algorithme 1

```
Pour i de 1 à n faire
        s ← 0
   Pour j de 1 à i faire
         s ← s+j
   FinPour
   T[i] ← s
FinPour
```

Algorithme 2

```
h ← 0
Pour i de 1 à n faire
  Pour j de 1 à i faire
    Pour k de j à n faire
        h ← h+1
    FinPour
  FinPour
FinPour
```

Exercice 3 (Complexité des algorithmes récursives)

Déterminez la complexité des fonctions récursives suivantes :

Fonction 1

```
fonction F(n:entier):entier
début
   Si (n=0) Alors
     retourner 1
   Sinon
     retourner (n*F(n-1))
Fin
```

Fonction 2

```
fonction G(n:entier):entier
début
   Si (n=0)Alors
     retourner 2
   Sinon
     retourner(G(n-1)*G(n-1))
Fin
```

Exercice 4 (Preuve par invariant de boucle)

On considère les fonctions suivantes :

Fonction 1

```
fonction Fact(n:entier):entier
variables f,i:entier
début
   f ← 1
   i ← 1
   Tanque(i<=n) faire
        f←f*i
        i←i+1
   FinTanque
   retourner f
Fin
```

Fonction 2

```
fonction Somme(n:entier):entier
Variables s, i: entier

début
   s ← 0
   pour i de 1 à n pas de 1 faire
       s ← s + i
   FinPour

retourner s
Fin
```

1-Donner une preuve formelle de terminaison de deux fonctions.

2-Montrer que :

a. $\mathcal{P}(k)$: $f_k = k!$; $0 \leq k \leq n$ est un invariant de boucle pour la fonction 1 ($\boldsymbol{f_k}$: la valeur stockée dans la variable $\boldsymbol{f}$ à l'issue de l'itération $\boldsymbol{k}$ de la boucle TantQue).

b. $\mathcal{R}(k)$: $S_k = \sum_{i=0}^{k} i$; $0 \leq k \leq n$ est un invariant de boucle pour la fonction 2 ($\boldsymbol{S_k}$: la valeur stockée dans la variable $\boldsymbol{S}$ à l'issue de l'itération $\boldsymbol{k}$ de la boucle Pour).

3- Quelle sont les valeurs renvoyées par les fonctions 1 et 2.

Exercice 5 (Preuve des algorithmes récursives)

Quelles sont les valeurs renvoyées par les fonctions suivantes ? Justifiez votre réponse.

Fonction 1:

```
Fonction mystere(n:entier):entier
Début
   Si (n=0) alors
    retourner 0
   Sinon
    retourner (Mystere(n-1)+ n)
   FinSi
Fin
```

Fonction 2:

```
Fonction star(a:entier, n:entier):entier
Début
   Si (n = 0) alors
       retourner 1
   Sinon
       retourner a*star(a,n - 1)
   FinSi
Fin
```

Exercice 6 (Exercice d'approfondissement)

On considère les portions d'algorithme ci-après :

Algorithme 1:

```
K ← 1                                [c1]
Pour i de 1 à n faire                [c2]
   j← 1                              [c3]
    Répeter
       j← j*3                        [c4]
    Jusqu'à(j>=n)                    [c5]
    k← k*2                           [c6]
FinPour
Retourner(K/2)                       [c7]
```

Algorithme 2 :

```
p ← 0                                [c1]
Pour i de 1 à n*n+1 faire            [c2]
   j← i                              [c3]
    Tantque(j<>0)faire               [c4]
       p ← p+1                       [c5]
       j ← j div 2                   [c6]
     FinTantque
FinPour
Retourner(p)                         [c7]
```

1- Montrer que la complexité temporelle dans le pire des cas est :
   a. Pour l'algorithme 1 : $O(n \log_3(n))$
   b. Pour l'algorithme 2 : $O(n^2 \log(n))$

2- Démontrer la terminaison et la correction des deux algorithmes.