DS nº 02 – Concours blanc

Les codes en python doivent être commentés et les indentations dans le code doivent être visibles.

I Exercice de cours – Calcul de factorielle

1. Ecrire une fonction `factorielle` qui prend comme entrée n, un entier positif et qui renvoie la valeur de $n!$.
   (On rappelle que, par convention, $0! = 1$.)
2. Montrer que la complexité en temps de l'algorithme précédent est linéaire, c'est-à-dire en $O(n)$.

II Exercice de TP – Recherche du maximum dans une liste de nombres.

1. Ecrire une fonction `maximum(liste)` qui prend comme entrée une liste de nombres non triée et renvoie le maximum de cette liste.

```
>>>L=[2,8,-7,3]
>>>maximum(L)
8
```

2. Ecrire une fonction `positionMax(liste)` qui renvoie le maximum et la position de ce maximum pour une liste de nombres :

```
>>>positionMax(L)
(8,1)
```

III Exercice – Variant de boucle

Soit n et m deux entiers naturels non nuls donnés. On considère l'algorithme suivant :

```
while n!=m :
        if n>m :
                n=n-m
        else:
                m=m-n
```

1. Rappeler la définition d'un variant de boucle.
2. Prouver que la boucle se termine en montrant que $\mathrm{Max}(n, m)$ est un variant de boucle.

IV Exercice – Evaluation d'un polynôme

Soit P un polynôme : $P(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n$.
On représente P par une liste qui contient ses coefficients : $L_p = (a_0, a_1, \cdots, a_n)$.

1. Ecrire une fonction `evalu(Lp,b)` qui prend comme entrée une liste de réels L_p de taille quelconque et un réel b. La fonction renvoie la valeur de $P(b)$.
2. Ecrire une fonction `evalu2(Lp,b)` qui renvoie la valeur de $P(b)$ mais sans utiliser la fonction puissance `**`.
3. Combien fait-on de multiplications dans `evalu2` ?
4. La méthode de Horner permet de calculer $P(b)$ en utilisant moins d'opérations de multiplication. Elle se base sur le constat suivant :

   $$P(b) = (((a_nb + a_{n-1}) \times b + a_{n-2}) \times b + \cdots a_2) \times b + a_1) \times b + a_0$$

   Ecrire une fonction `evaluHorner(Lp,b)` qui renvoie la valeur de $P(b)$ en utilisant le principe précédent.
5. Combien fait-on de multiplications avec la méthode de Horner ?

Correction DS n° 02 – Concours blanc

III Exercice – Variant de boucle

1. $\max(n, m)$ est un entier ;
2. Au départ, n et m sont strictement positifs. A l'étape k, si $n_k > 0$ et $m_k > 0$, alors n_{k+1} et m_{k+1} sont encore strictement positifs. En effet : si $n_k > m_k$, alors $n_{k+1} = n_k - m_k > 0$. Si $n_k < m_k$, alors $m_{k+1} = m_k - n_k > 0$.
   Donc $\max(m, n)$ est positif.
3. $\max(m, n)$ est strictement décroissante. En effet, dans la boucle while, $n_k \neq m_k$. Il y a donc deux possibilités. Si $n_k > m_k$ alors $\max(n_k, m_k) = n_k$. Dans ce cas : $n_{k+1} = n_k - m_k < n_k = \max(n_k, m_k)$. Donc $n_{k+1} < \max(n_k, m_k)$ et $m_{k+1} = m_k < n_k = \max(n_k, m_k)$. Finalement, les deux nombres n_{k+1} et m_{k+1} sont strictement inférieurs à $\max(n_k, m_k)$. Donc : $\max(n_{k+1}, m_{k+1}) < \max(n_k, m_k)$.
   L'autre cas $n_k < m_k$ est similaire.

$\boxed{\text{Donc } \max(n, m) \text{ est un variant de boucle. Le programme s'arrête.}}$

IV Exercice – Evaluation d'un polynôme

1.
```
def evalu(Lp,x0):
    deg=len(Lp)-1
    s=0
    for i in range(deg+1):
        s=s+Lp[i]*x0**i
    return(s)
```

2.
```
def evalu2(Lp,x0):
    deg=len(Lp)-1
    s=0
    x=1
    for i in range(deg+1):
        s=s+Lp[i]*x
        x=x*x0
    return(s)
```

3. On effectue deux multiplications dans la boucle. Donc $\boxed{\text{on effectue } 2(\deg(P)+1) \text{ multiplications.}}$

4.
```
def evaluHorner(Lp,x0):
    deg=len(Lp)-1
    s=Lp[deg]
    for i in range(1,deg+1):
        s=x0*s+Lp[deg-i]
    return(s)
```

5. On effectue un multiplication dans la boucle. Donc $\boxed{\text{on effectue } \deg(P) \text{ multiplications.}}$