

IC Devoir 1 - Lundi 16 octobre 15h30 à 16h30

Exercice 1 Compléter le script de la fonction suivante qui calcule

$$e1(n) = \sum_{k=0}^n \frac{1}{k!}$$

A la fin du k -ième tour de boucle, la variable f contient le nombre $k!$ tandis que la variable r contient $e1(k)$.

```
def e1(n):
    r=...
    f=...
    for k in range(1,...):
        f=...
        r=...
    return r
```

Exercice 2 Considérons une liste L contenant 5 éléments ; donner les instructions utilisant les méthodes du type list, si possible, pour :

1. ajouter l'élément f à la fin de L
2. ajouter l'élément d au début de L
3. enlever l'élément en position i et le rajouter à la fin
4. insérer l'élément e en position p
5. renverser la liste
6. trier la liste
7. compter le nombre d'occurrences de l'élément a
8. donner la position de la première occurrence de l'élément b , qui est dans L

Exercice 3 Définir la fonction $g : (x, y) \mapsto (x + y, xy)$:

- d'une part avec la définition usuelle def ;
- d'autre part avec l'instruction lambda.

Exercice 4 Identifier la valeur affichée :

```
x=float(input("Donner_un_reel_>0_"))
n=1
while 1/n>=x:
    n=n+1
print(n)
```

Exercice 5 Considérons la suite définie par :

$$\begin{cases} u_0 = 2 & u_1 = -1 \\ \forall n \in \mathbb{N}, & u_{n+2} = 2u_{n+1} + 3u_n \end{cases}$$

Écrire une fonction suite(n) qui retourne le terme de rang n de la suite.

Exercice 6 Considérons la fonction suivante qui prend comme paramètre d'entrée une liste de nombres :

```
def f(L):
    while len(L)>1:
        if L[0]>L[1]:
            L.pop(1)
        else:
            L.pop(0)
    return L[0]
```

1. Reproduire et remplir le tableau suivant qui donne les valeurs au début de chaque boucle pour la liste $[2, -1, 3, 0, 1, 2]$

len(L)	L
6	$[2, -1, 3, 0, 1, 2]$
⋮	⋮
⋮	⋮

2. En déduire ce que fait la fonction.

Exercice 7 Tri par sélection

Prog

1. Donner une fonction pmin(L) qui retourne la position d'un minimum de L .
2. Décrire en quelques mots le tri par sélection (ordre croissant).
3. Proposer une fonction tri(L) qui trie par sélection une liste L .
4. Décrire le déroulement de ce tri appliqué à la liste $[2, -3, -4, 5, 2, 1]$ en reproduisant et complétant le tableau ci-dessous :

boucle	L	pmin(L)	T
X	$[2, -3, -4, 5, 2, 1]$	X	$[\]$
1
⋮	⋮	⋮	⋮



IPT Devoir 1- Proposition de solutions

Solution 1

```
def e1(n):
    r=1
    f=1
    for k in range(1,n+1):
        f=f*k
        r=r+1/f
    return r
```

Solution 2

D'autres instructions sont possibles

```
1 L.append(f) #L=L+[f]
2 L=[d]+L
3 L.append(L.pop(i))
4 L.insert(p,e) #L=L[:p]+[e]+L[p:]
5 L.reverse() #L=L[::-1]
6 L.sort()
7 L.count(a)
8 L.index(b)
```

Solution 3

```
def g(x,y):
    return x+y,x*y
g=lambda x,y:(x+y,x*y)
```

Solution 4

Ce script affiche le plus petit entier naturel non nul, n , tel que

$$\frac{1}{n} < x \Leftrightarrow n > \frac{1}{x}$$

Il calcule $\left\lfloor \frac{1}{x} \right\rfloor + 1$.

Solution 5

```
def suite(n):
    a,b=2,-1
    if n==0:
        return 2
    for i in range(2,n+1):
        a,b=2*b+3*a,b
    return b
```

Solution 6

1. Le suivi des variables donne :

len(L)	L
6	[2, -1, 3, 0, 1, 2]
5	[2, 3, 0, 1, 2]
4	[3, 0, 1, 2]
3	[3, 1, 2]
2	[3, 2]
1	[3]

2. La fonction retourne le maximum de la liste.

Solution 7 Tri par sélection

Position d'un minimum

```
def pmin(L):
    '''
    Entree : L une liste de nombres reels
    Sortie : la position d'un minimum
    '''
    position=0
    for i in range(1,len(L)):
        if L[i]<L[position]:
            position=i
    return position
```

Méthode

→ Le tri par sélection consiste à :

- on cherche le minimum de la liste, on le place dans une liste initialement vide et on le retire de la liste de départ
- puis on recommence, ainsi de suite, jusqu'à ce que la liste de départ soit vidée.

Tri d'une liste

```
def tri(L):
    '''
    Entree : L une liste de nombres reels
    Sortie : T la liste triee
    (methode par selection)
    '''
    LL,T=list(L),[] # copie de L, liste vide
    while len(LL)>0:
        p=pmin(LL)
        T.append(LL.pop(p))
    return T
```

Attention ! L'instruction `LL.pop(p)` renvoie l'élément en position p de la liste `LL` et le retire de la liste. Ainsi, il est possible d'emboîter (de composer) les deux instructions : `T.append(LL.pop(p))`.

► Description du déroulement du tri appliqué à la liste `[2, -3, -4, 5, 2, 1]` :

boucle	L	pmin(L)	T
X	[2, -3, -4, 5, 2, 1]	X	[]
1	[2, -3, 5, 2, 1]	2	[-4]
2	[2, 5, 2, 1]	1	[-4, -3]
3	[2, 5, 2]	3	[-4, -3, 1]
4	[5, 2]	0	[-4, -3, 1, 2]
5	[5]	1	[-4, -3, 1, 2, 2]
5	[]	0	[-4, -3, 1, 2, 2, 5]

La liste ordonnée associée est `[-4, -3, 1, 2, 2, 5]`.