

ENSAE année 2012 - 2013

Exercices non proposés pour un examen

1

1) On dispose d'une liste de mots et d'un texte sans espace. On souhaite remettre les espaces.
(3 points)

```
l = ["par", "parti", "pas", "pars"]  
t = "parpasparsparti"
```

Quelques rappels :

```
s.startswith(d) # retourne True si la chaîne de caractères s  
                # commence par d  
s [i:j]        # extrait la sous-chaîne de caractère entre les positions i et j (exclu)
```

Utiliser un dictionnaire à la place de la liste `l` peut se révéler plus pratique.

2

1) La solution d'un polynôme du second degré $ax^2 + bx + c = 0$ se met sous la forme $x_1 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$ et $x_2 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$. Ecrire une fonction qui prend en paramètres trois réels et qui retourne les deux solutions en minimisant le nombre d'opérations.

Tester la fonction pour les trois valeurs suivantes :

```
a = 10.**i/3  
c = 10.**(-i)/4*3  
b = 1
```

Expliquer le résultat. (2 point)

2) On cherche à calculer une intégrale $\int_0^1 \ln x dx = [x \ln x - x]_0^1 = -1$ de trois manières différentes afin de comparer le temps de calcul. Les trois programmes sont décrits ci-dessous ainsi que ce qu'ils affichent.

```
import time, math
debut = time.clock()

y = 0
N = 1000000

for i in range (1,N+1) :
    s = i*1.0/N
    x = math.log(s)
    y += x

fin = time.clock()
print "integ ",y/N
print "temps ",fin - debut
```

integ -0.999992173306
temps 0.780309832669

```
import time, math
debut = time.clock()

y = 0
N = 1000000
z = math.log(N)
for i in range (1,N+1) :
    x = math.log(i*1.0)
    y += x
y -= z*N
fin = time.clock()
print "integ ",y/N
print "temps ",fin - debut
```

integ -0.999992173306
temps 0.5965760959

```
import time, math
debut = time.clock()
def monlog (i) :
    return math.log(i)
y = 0
N = 1000000
z = math.log(N)
for i in range (1,N+1) :
    x = monlog(i*1.0)
    y += x
y -= z*N
fin = time.clock()
print "integ ",y/N
print "temps ",fin - debut
```

integ -0.999992173306
temps 0.698039702544

On veut répondre à deux questions :

1. Quel est le programme le plus rapide et pourquoi ? (1 point)
2. Que coûte un appel de fonction ? (1 point)

3) On cherche à calculer une intégrale $\int_0^1 \ln x dx = [x \ln x - x]_0^1 = -1$ de trois manières différentes afin de comparer le temps de calcul. Les trois programmes sont décrits ci-dessous ainsi que ce qu'ils affichent.

```
import time, math

debut = time.clock()

y = 0
N = 1000000
z = math.log(N)
for i in range (1,N+1) :
    x = math.log(i*1.0)
    y += x
y -= z*N
fin = time.clock()
print "integ ",y/N
print "temps ",fin - debut
```

integ -0.999992173306
temps 0.6014340817

```
import time, math

debut = time.clock()
def monlog (i) :
    return math.log(i)
y = 0
N = 1000000
z = math.log(N)
for i in range (1,N+1) :
    x = monlog(i*1.0)
    y += x
y -= z*N
fin = time.clock()
print "integ ",y/N
print "temps ",fin - debut
```

integ -0.999992173306
temps 0.68944518564

```
import time, math
memolog = { }
for i in range(1,N+1) :
    memolog [i] = math.log(i)
debut = time.clock()

y = 0
N = 1000000
z = math.log(N)
for i in range (1,N+1) :
    x = memolog[i]
    y += x
y -= z*N
fin = time.clock()
print "integ ",y/N
print "temps ",fin - debut
```

integ -0.999992173306
temps 0.424331883008

On veut répondre à deux questions :

1. Quel est le programme le plus rapide et pourquoi ? (1 point)
2. Que coûte un appel de fonction ? (1 point)

4) On dispose d'une liste de mots et d'un texte sans espace. On souhaite remettre les espaces. (3 points)

```
l = ["par", "parti", "pas", "pars"]
t = "parpasparsparti"
```

Quelques rappels :

```
s.startswith(d) # retourne True si la chaîne de caractères s
                # commence par d
s [i:j]         # extrait la sous-chaîne de caractère entre les positions i et j (exclu)
```

Utiliser un dictionnaire à la place de la liste `l` peut se révéler plus pratique. L'algorithme proposé est le suivant :

1. On cherche un mot dans la liste qui commence la chaîne de caractères `s`.
2. Si on trouve un mot `m`, on passe à la position `p = m` sinon à la position `p = 1`.
3. On retourne à l'étape 1 en considérant la chaîne de caractères à partir de la position `p`.

5) La solution d'un polynôme du second degré $ax^2 + bx + c = 0$ se met sous la forme $x_1 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$ et $x_2 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$. Ecrire une fonction qui prend en paramètres trois réels et qui retourne les deux solutions en minimisant le nombre d'opérations.

Tester la fonction pour les trois coefficients suivants avec i allant de 1 à 20 :

```
a = 10.0**i/3      # ne pas changer 10.0 en 10 pour éviter
c = 10.0**(-i)/4*3 # les problèmes de calculs avec des entiers
b = 1              # ** est l'opérateur puissance
```

Expliquer le résultat. (2 point)