

INFO-H-100 – Informatique – Partie Programmation – Prof. Th. Massart
1^{ère} année du grade de Bachelier en Sciences de l'Ingénieur
Interrogation de janvier

Remarques préliminaires

- On vous demande de répondre à **chaque question sur une feuille séparée**.
- N'oubliez pas d'inscrire votre nom, prénom et numéro de matricule sur chaque feuille.
- Vous disposez de 3 heures et vous ne pouvez pas utiliser de notes.
- La réponse à la question doit comprendre, si approprié, le code *Python* structuré et conforme aux règles de bonne pratique et conventions ainsi que des commentaires pertinents.
- Vous pouvez ajouter des fonctions si cela vous semble nécessaire.
- Sauf mention contraire, vous ne pouvez utiliser aucune fonction de bibliothèques (pas d'import).

Question 1 - Théorie (5 points)

Que donnent les codes suivants ? Justifiez vos réponses et décrivez la situation et son évolution grâce à des diagrammes d'état (comme fait au cours).

```
1. x=3
   x="bonjour"
   y=x
   print x,y
```

```
2. x=range(5)
   y=x
   z=y[1:2]+y[3:4]
   y[3]=5
   print x,y
   z[2]=3.14
   print z
```

```
3. def foo(x):
    x[2]="salut"
    x=[0,2,4]
```

```
    w=[0,1,2]
    foo(w)
    print w
```

```
4. t=[0,1]
   t=[2,3,t]
   print t
   t[0]=t
   print t
```

```
5. liste1=[[1,5,1,5],"Marignan"]
   liste2=liste1[:]
   liste2[0][1]=9
   liste2[1]="guerre"
   print liste1
   print liste2
```

Question 2 - Approximation de $\ln(x)$ (6 points)

On vous demande d'écrire une fonction qui calcule l'approximation de $\ln(x)$, où $x \in R_{>0}$, en utilisant la série suivante :

$$\ln(x) = 2 \left[\left(\frac{x-1}{x+1} \right) + \frac{1}{3} \left(\frac{x-1}{x+1} \right)^3 + \frac{1}{5} \left(\frac{x-1}{x+1} \right)^5 + \dots \right]$$

Les calculs s'arrêteront quand la valeur absolue du dernier terme ajouté est inférieure à **EPS** ou lorsque le nombre de termes considérés atteint la borne **NMAX**. **EPS** et **NMAX** sont considérés comme des constantes définies globalement.

Veillez à faire particulièrement attention à l'efficacité de vos calculs pour cette question.

Question 3 - Triangle de Pascal (8 points)

Le triangle de Pascal est formé de la façon suivante :

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
...
```

Si $b_i^{(n)}$ représente le $i^{\text{ème}}$ élément de la $n^{\text{ème}}$ ligne du triangle (en commençant à compter en 0), le polynôme $(x+a)^n$ peut être obtenu de la façon suivante (Bien entendu, $b_i^{(n)}$ est le coefficient binomial $\binom{n}{i}$) :

$$(x+a)^n = \sum_{i=0}^n b_i^{(n)} x^{n-i} a^i$$

Soient des polynômes en x , représentés sous la forme d'une liste p de taille $n+1$ où chaque $p[i]$ donne le coefficient d'indice i du polynôme de degré n :

$$p^{(n)}(x) = \sum_{i=0}^n a_i x^i \iff p[i] = a_i$$

Le but de l'exercice est d'écrire une fonction qui, pour un n et un a donnés, renvoie le polynôme $(x+a)^n$.

On vous demande d'écrire 3 fonctions.

1. Une fonction `ligne_suivante(ligne_courante)` qui reçoit en paramètre une ligne du triangle de Pascal (sous la forme d'une liste d'entiers) et qui calcule la ligne suivante.
2. Une fonction `ligne(n)` qui reçoit en paramètre un entier n et qui renvoie la ligne correspondante du triangle de Pascal, sous la forme d'une liste d'entiers. On numérote les lignes à partir de 0, ce qui permet d'avoir une correspondance entre les coefficients de $(x+a)^n$ et la ligne n du triangle.
3. Une fonction `pascal(a,n)` qui renvoie $(x+a)^n$ dans la représentation définie ci-dessus.

Question 4 - Tri de mains de cartes (6 points)

On représente une carte par un tuple (`valeur`, `couleur`) où `valeur` est un entier de 1 à 13 (les 3 dernières valeurs représentent respectivement le valet, la dame et le roi) et où `couleur` est une chaîne de caractères parmi "coeur", "pique", "carreau", "trèfle".

Une main est représentée par une liste de cartes.

On vous demande d'écrire une fonction qui trie, de façon croissante, une liste de mains en fonction du nombre de cartes de carreaux qu'elles contiennent. Pour cela, vous pouvez utiliser soit le tri par insertion, soit le tri par sélection (**veillez à préciser sur votre copie quel tri vous avez utilisé**).

Veillez à faire particulièrement attention à votre découpe en fonctions.

```
main1 = [(1,"carreau"),(2,"carreau"),(3,"carreau")]
main2 = [(1,"pique"),(2,"coeur"),(3,"carreau")]
main3 = [(1,"carreau"),(2,"coeur"),(3,"carreau")]
mains = [main1,main2,main3]
```

```
triSelection(mains)
```

Par exemple, le morceau de code ci-dessus modifie la liste `mains` de la façon suivante :

```
[
 [(1, 'pique'), (2, 'coeur'), (3, 'carreau')],
 [(1, 'carreau'), (2, 'coeur'), (3, 'carreau')],
 [(1, 'carreau'), (2, 'carreau'), (3, 'carreau')]
]
```

BON TRAVAIL !

INFO-H-100 – Informatique – Partie Programmation – Prof. Th. Massart
1^{ère} année du grade de Bachelier en Sciences de l'Ingénieur
Correction de l'interrogation de janvier

Question 2 - Approximation de $\ln(x)$

EPS = 1E-6

NMAX = 1000

```
def ln(x):
    fact = (x - 1.0)/(x + 1)
    fact2 = fact * fact
    den = 1
    nbTermes = 1
    terme = fact
    somme = fact
    while abs(terme) >= EPS and nbTermes < NMAX:
        nbTermes += 1
        den += 2
        fact *= fact2
        terme = fact / den
        somme += terme
    return 2 * somme
```

Question 3 - Triangle de Pascal

```
def ligne_suivante(ligne_courante):
    lc = ligne_courante + [0]
    ls = [1]
    for i in range(len(lc)-1):
        ls.append(lc[i] + lc[i+1])
    return ls
def ligne(n):
    li = [1]
    for i in range(n):
        li = ligne_suivante(li)
    return li
def pascal(a, n):
    coeffs = ligne(n)
    f = 1
    for i in range(n,-1,-1):
        coeffs[i] *= f
        f *= a
    return coeffs
```

Question 4 - Tri de mains de cartes

```
def compteCarreaux(liste):
    somme = 0
    for elem in liste:
        if elem[1] == "carreau":
            somme+=1
    return somme

def plusPetit(main1,main2):
    return compteCarreaux(main1)<compteCarreaux(main2)

def posMinFrom(ls, i):
    """La position du min de ls en commençant en i.
    Pre: i dans ls
    """
    res = i
    while i < len(ls):
        if plusPetit(ls[i],ls[res]):
            res = i
        i += 1
    return res

def echange(ls, i1, i2):
    ls[i1], ls[i2] = ls[i2], ls[i1]

def triSelection(ls):
    for i in range(len(ls) - 1):
        pos = posMinFrom(ls, i)
        echange(ls, i, pos)

def triInsertion(ls):
    for i in range(1, len(ls)):
        val = ls[i]
        j = i
        while j > 0 and plusPetit(val, ls[j-1]):
            ls[j] = ls[j-1]
            j -= 1
        ls[j] = val

main1 = [(1,"carreau"),(2,"carreau"),(3,"carreau")]
main2 = [(1,"pique"),(2,"coeur"),(3,"carreau")]
main3 = [(1,"carreau"),(2,"coeur"),(3,"carreau")]

mains = [main1,main2,main3]
mains_bis = mains[:]

triSelection(mains)
triInsertion(mains_bis)

print mains
print mains_bis
```