

## Informatique, composition n° 2

Le mercredi 25 janvier 2023

**Exercice 1**

La suite des nombres de Bernoulli notée  $(b_n)_{n \in \mathbb{N}}$  est définie par :

$$b_0 = 1, \quad \forall n \geq 1, \quad b_n = \frac{-1}{n+1} \sum_{k=0}^{n-1} \binom{n+1}{k} b_k.$$

**Leonhard Euler** (1707-1783) a démontré la formule suivante qui exprime les nombres  $\zeta(2k)$ , où  $\zeta$  est la fonction  $x \mapsto \zeta(x) = \sum_{n=1}^{+\infty} \frac{1}{n^x}$ , à l'aide des nombres de Bernoulli :

$$\forall k \in \mathbb{N}^*, \quad \zeta(2k) = \frac{(-1)^{k-1} 2^{2k-1} \pi^{2k} b_{2k}}{(2k)!}.$$

Dans cet exercice (informatique pour tous), on se propose de programmer le calcul des nombres de Bernoulli  $b_n$  afin d'obtenir des valeurs exactes de  $\zeta(2k)$ .

Les algorithmes demandés doivent être écrits en langage Python. On sera très attentif à la rédaction du code notamment à l'indentation.

1. Écrire une fonction `factorielle(n)` qui renvoie la factorielle d'un entier  $n \in \mathbb{N}$ .
2. On considère la fonction Python suivante `binom(n, p)` qui renvoie le coefficient binomial  $\binom{n}{p}$  :

```
def binom(n, p):
    if not (0 <= p <= n):
        return 0
    return factorielle(n) // (factorielle(p) * factorielle(n - p))
```

Combien de multiplications sont effectuées lorsque l'on exécute `binom(30, 10)` ?

Expliquer pourquoi il est possible de réduire ce nombre de multiplications à 20 ?

Quel serait le type du résultat renvoyé si l'on remplaçait la dernière ligne de la fonction `binom` par `return factorielle(n) / (factorielle(p) * factorielle(n - p))` ?

3. Démontrer que, pour  $n \geq p \geq 1$ , on a

$$\binom{n}{p} = \frac{n}{p} \binom{n-1}{p-1}.$$

En déduire une fonction récursive `binom_rec(n, p)` qui renvoie le coefficient binomial  $\binom{n}{p}$ .

4. Démontrer que, pour  $n \geq p \geq 1$ , on a

$$\binom{n}{p} = \binom{n-1}{p-1} + \binom{n-1}{p}$$

En déduire une fonction récursive `binom_rec2(n, p)` qui renvoie le coefficient binomial  $\binom{n}{p}$ .

5. Expliquer, ligne par ligne, ce que fait le code suivant :

```

dico={}
def binom_asc(p,n):
    if 2*p>n :
        p=n-p
    dico[(0,1)]=1
    dico[(1,1)]=1
    for k in range(2,n+1):
        dico[(0,k)]=1
        dico[(k,k)]=1
        for i in range(1,k):
            dico[(i,k)]=dico[(i,k-1)]+dico[(i-1,k-1)]
    return dico[(p,n)]

```

6. Écrire une fonction non récursive `bernoulli(n)` qui renvoie une valeur approchée du nombre rationnel  $b_n$ . On pourra utiliser librement une fonction `binomial(n, p)` qui renvoie le coefficient binomial  $\binom{n}{p}$ .

Par exemple, `bernoulli(10)` renvoie 0,075 757 575 757 575 76 qui est une valeur approchée de  $b_{10} = \frac{5}{66}$ .

7. Proposer un code Python basé sur votre script de la question 4 utilisant le principe de *mémoïsation* et une fonction `binom_rec3(n, p)` qui renvoie le coefficient binomial  $\binom{n}{p}$ .

## Exercice 2

- Proposer une fonction Python `Maxliste`, qui mange une liste de nombres `L` et qui renvoie une liste de deux éléments dont le premier est le maximum de la liste et le second un indice où ce maximum est atteint.
- (a) Expliquez ce que renvoie la fonction suivante où `L` est une liste de nombres entiers compris (au sens large) entre 0 et 1000.

```

def Occurence(L):
    K=[0 for k in range(1001)]
    for i in range(len(L)):
        K[L[i]]=K[L[i]]+1
    return K

```

- (b) Proposer une fonction `TriOcc` qui permet de trier par ordre croissant une liste `L` de nombres entiers compris (au sens large) entre 0 et 1000 qui utilise la fonction `Occurence`.
- Que fait la fonction suivante où `L` est une liste de nombres entiers naturels ? On expliquera avec précision.

```

def function(L):
    for i in range(1,len(L)):
        insertion=L[i]
        j=i
        while j>0 and L[j-1]>insertion :
            L[j]=L[j-1]
            j=j-1
        L[j]=insertion
    return L

```

- Proposer une fonction Python `fusion` qui mange deux listes d'entiers naturels `L1` et `L2` triées par ordre croissant et qui renvoie une liste `L` composées de tous les nombres des listes `L1` et `L2` triés par ordre croissant.