I. Aide-mémoire CC-INP

tableau à une dimension

```
>>> L=[1,2,3] (liste)
>>> v=array([1,2,3]) (vecteur)
```

accéder à un élément

`>>> v[0]` renvoie 1

ajouter un élément

`>>> L.append(5)` uniquement sur les listes

tableau à deux dimensions (matrice) :

```
>>> M=array(([1,2,3],[3,4,5]))
```

accéder à un élément

`>>> M[1,2]` ou

`>>> M[1][2]` donne 5

Extraire une portion de tableau (2 premières colonnes)

```
>>> M[ :,0 :2]
```

tableau de 0 (2 lignes, 3 colonnes)

```
>>> zeros((2,3),float)
```

séquence équirépartie quelconque de 0 à 10.1 (exclus) par pas de 0.1

```
>>> arange(0,10.1,0.1)
```

Définir une chaîne de caractères

```
>>> mot="Python"
```

Taille d'une chaine

```
>>> len(mot)
```

Extraire des caractères

```
>>> mot[2 :7]
```

Boucle For

```
for i in range(10) :
        print( i )
```

Condition If

```
if ( i >3) :
        print( i )
else :
        print(" hello ")
```

Boucle While

```
i=0
while i < 5 :
        print( i )
        i=i+1
```

Définir une fonction

qui possède un argument et renvoie 2 résultats

```
def fonction (param) :
            res1=param
            res2=param*param
            return (res1,res2)
```

Exercice 1 *ATS 2019*

Partie C – Algorithmique

Les deux questions de cette partie sont indépendantes.

1. On souhaite obtenir un encadrement du réel α, solution de l'équation $\mathrm{sh}(\alpha) - 1 = 0$, en appliquant un procédé de dichotomie. Recopier en la complétant la fonction *Python* suivante, qui prend en argument un réel strictement positif $\varepsilon > 0$, et renvoie deux réels a et b vérifiant $a \leq \alpha \leq b$ et $b - a \leq \varepsilon$.

```
import numpy as np

def dicho(eps)
    a = ...
    b = ...
    while ...... :
      c = (a+b)/2
      if ...... :
           a = c
      else b = c
 return([a,b])
```

Note : la fonction sh *s'écrit* `np.sinh` *en Python.*

2. En utilisant la fonction précédente sur machine, on trouve $0,881$ comme valeur approchée de α. On admet de plus que la suite $(I_n)_{n\in\mathbb{N}}$ vérifie

$$I_0 = \alpha, \qquad \text{et pour tout } n \in \mathbb{N}, \quad I_{n+1} = \frac{\sqrt{2}}{2n+2} - \left(\frac{2n+1}{2n+2}\right) I_n.$$

Écrire une fonction, en *Python* ou bien en pseudo-code, qui prend en entrée un entier naturel n et renvoie une valeur approchée de I_n.

Exercice 2 *e3a PC 2018*

On désigne par $\mathbb{N}^*$ l'ensemble des entiers naturels non nuls. Pour tous k, n dans $\mathbb{N}^*$, on note

$$S_k(n) = \sum_{i=1}^{n} i^k.$$

On pourra utiliser le fait que : $S_2(n) = n(n+1)(2n+1)/6$.

1. Soit k dans $\mathbb{N}^*$. Justifier que la suite $\left(\frac{1}{n^{k+1}} S_k(n)\right)_{n\in\mathbb{N}^*}$ converge vers $\frac{1}{k+1}$.

Si X est une variable aléatoire à valeurs das une partie finie de $\mathbb{N}$, on note $E(X)$ son espérance et $V(X)$ sa variance. Soit N un entier naturel supérieur ou égal à 2.

2. Soit X une variable aléatoire à valeurs dans $[\![1, N]\!]$. Démontrer que

$$E(X) = \sum_{i=1}^{N} P(X \geq i).$$

On dispose d'une boîte dans laquelle sont placés des jetons indiscernables au toucher numérotés de 1 à N. Soit k un entier naturel ≥ 2. On tire k fois de suite un jeton dans cette boîte. On note son numéro et on le remet dans la boîte. Les tirages sont indépendants les uns des autres. On note X_i la variable aléatoire qui prend comme valeur le numéro du jeton du $i^{\text{ème}}$ tirage, pour i dans $[\![1, N]\!]$. On suppose que la loi de X_i est uniforme sur $[\![1, N]\!]$. On note U_k et V_k les variables aléatoires :

$$U_k = \min(X_1, \ldots, X_k) \text{ et } V_k = \max(X_1, \ldots, X_k).$$

3. Exprimer $E(X_1)$, $E(X_1^2)$ et $V(X_1)$ en fonction de N.
4. On se propose de simuler en Python les variables V_k pour $N = 10$.
   (a) Ecrire une fonction `simulX` qui renvoie une liste de longueur 100 de réalisations des variables $X_1, \ldots, X_{100}$.
   On pourra utiliser la fonction : random.randint.
   L'instruction random.randint(1,10) fournit un nombre entier aléatoire dans $[\![1, 10]\!]$ uniformément.
   (b) En déduire une fonction `REALIV` qui renvoie une liste de longueur 100 de réalisations des variables $V_1, \ldots, V_{100}$.
5. Soit k dans $\mathbb{N}^*$ supérieur à 2.
   (a) Soit $i \in [\![1, N]\!]$. Justifier que :
   $$P(U_k \geq i) = \left(\frac{N-i+1}{N}\right)^k.$$
   (b) On appelle plusieurs fois la fonction `REALIV` de la question 4b. On constate qu'à chaque fois, le résultat obtenu est une liste qui se termine par un grand nombre de 10. Justifier mathématiquement ce résultat.
   (c) Exprimer $E(U_k)$ en fonction de N à l'aide de la fonction S_k introduite au début de l'exercice. Donner un équivalent de $E(U_k)$ lorsque N tend vers $+\infty$.
6. (a) On introduit les variables $Y_i = N + 1 - X_i$, pour i dans $[\![1, N]\!]$. Justifier que les variables $(Y_1, \ldots, Y_N)$ sont indépendantes et de même loi. Préciser cette loi.
   (b) En déduire $E(V_k)$ et $V(V_k)$ en fonction de $E(U_k)$ et $V(U_k)$.
7. On considère le couple de variables aléatoires (U_2, V_2).
   (a) Exprimer $U_2 + V_2$ et $U_2 V_2$ en fonction de X_1 et X_2.

(b) En déduire $V(U_2+V_2)$ et $E(U_2V_2)$ en fonction de N.
On peut en déduire par un calcul la covariance de U_2 et V_2, notée $\mathrm{Cov}(U_2,V_2)$. On admet sa valeur :

$$\mathrm{Cov}(U_2,V_2)=\frac{(N^2-1)^2}{36N^2}.$$

(c) Exprimer $V(U_2)$ et $V(V_2)$ en fonction de N.

(d) On note $\rho_2(N)$ le coefficient de corrélation de U_2 et V_2. Exprimer $\rho_2(N)$ en fonction de N.

(e) Que peut-on dire de la suite $(\rho_2(N))_{N\in\mathbb{N}\setminus\{0,1,2\}}$ lorsque N tend vers $+\infty$?

8. (a) Si X est une variable aléatoire à valeurs dans une partie finie de $\mathbb{N}$, démontrer que $E(X^2)=\sum_{i=1}^{N}(2i-1)P(X\geq i)$.

(b) Exprimer $E(U_k^2)$ en fonction de N à l'aide des fonctions S_k et S_{k+1} introduites au début de l'exercice.

(c) Exprimer $V(U_k)$ en fonction de N à l'aide des fonctions S_k et S_{k+1} introduites au début de l'exercice.

(d) Donner un équivalent de $V(U_k)$ lorsque N tend vers $+\infty$.