

Exercice I (e3a 2018)

1. en python les matrices sont définies comme listes de vecteurs lignes, l'énoncé nous précise donc que l'on a en argument la transposée d'une matrice (argument ainsi noté tM).

```

1 def matriceHn(tM):
2     n = len(tM)
3     for i in range(n) :
4         for j in range(n) :          # matrice carree
5             if tM[i][j] != -1 or tM[i][j] != 1 : # <=> tM[i][j] not in [-1, 1]
6                 return 0
7     # la matrice ne contient que des -1 et 1
8     for i in range(n) :
9         for j in range(i+1,n) :
10            ps = 0
11            for k in range(n) :      # calcul du produit scalaire
12                ps += tM[i][k]*tM[j][k]
13            if ps != 0 :              # vecteurs non orthogonaux
14                return 0
15    return 1

```

- 2.(a)
- ```

1 def simulU():
2 import random
3 n = 10
4 X = []
5 for i in range(n) : # generation aleatoire des Xi
6 X.append(random.randint(0,25))
7 S = [X[0]]
8 for i in range(1,n) :
9 if X[i] not in S : # attention pour le cout, boucle for cachee
10 S.append(X[i])
11 return len(S)

```
- 

- (b) En considérant la suite de variables aléatoires indépendantes et identiquement distribuées  $(U_n^i)_i$  et en appliquant la loi forte des grands nombres, on a  $\frac{1}{m} \sum_{i=1}^m U_n^i \xrightarrow{p.s.} E[U_n]$ . Ainsi :

---

```

1 def espU():
2 m = 10**5 # on espere m assez grand pour la convergence p.s.
3 Somme = 0
4 for i in range(m) :
5 Somme += simulU()
6 return Somme/m

```

---

## Exercice II (CCP 2017)

On peut proposer les solutions suivantes :

- 1.
- ```

1 A = [[1,2,3],[4,5,6],[7,8,9],[10,11,12]]

```
-
- 2.
- ```

1 len(A) = 4
2 A[1] = [4,5,6]
3 A[2][1] = 8

```
- 
- 3.
- ```

1 def difference(x,y):
2     result = []
3     for i in range(len(x)):
4         result.append( x[i] - y[i] )
5     return result

```
-

```

4. 1 def norme(x):
2     result = abs(x[0])
3     for i in range(1,len(x)):
4         if abs(x[i]) > result:
5             result = abs(x[i])
6     return result

```

```

1 def itere(x,A):
2     p = len(x)
3     result = []
4     for i in range(p):
5         y = 0
6         for j in range(p):
7             y = y + x[j]*A[j][i]
8         result.append(y)
9     return result

```

```

1 def probainvariante(A,eps):
2     p = len(A)
3     mu0 = [ 1/p ]*p
4     mu1 = itere( mu0 , A )
6. 5 while norme ( difference ( mu1 , mu0 ) ) > eps :
6     mu0 = mu1
7     mu1 = itere( mu0 , A )
8     return mu1

```

Exercice III (CCP 2018)

```

1 def factorielle(n):
2     if n<=1 :
1. 3         return 1
4     return n*factorielle(n-1)

```

```

1 N = 1
2 epsilon = 10**(-6)
2. 3 while 1/( (2*N+3)*factorielle(N+1) ) > epsilon
4     N += 1
5     print(N)

```

```

1 def lia(x, i, a):
2     p = 1
3     for k in range(len(x)) :
4         if k!=i :
5             p *= (a-x[k])/(x[i]-x[k])
6     return p

```

```

3. 7
8 def lagrange(x, y, a):
9     P = 0
10    for i in range(len(x)) :
11        P += y[i]*lia(x,i,a)
12    return P

```
