2BCPST	DS n°4 : CONCOURS BLANC (durée 3h)	mardi 7 janvier 2025

Ce problème est consacré à l'étude et la modélisation d'une file d'attente à un ou plusieurs guichets. On y étudiera deux modèles régulant différemment l'arrivée aléatoire des usagers. Dans la partie A le temps est continu, dans la partie B il est discrétisé. A la fin du sujet se trouve une notice avec les commandes en Python pouvant être utilisées.

A) : le modèle continu

Partie I : les premiers clients dans le cas de guichets multiples

Dans cette partie il y a r guichets. On appelle durée du service d'une personne au guichet le temps passé par l'employé au guichet à la servir, une fois son attente dans la file achevée.
n personnes (où $n \geq r+1$) sont rangées dans une file d'attente lorsque les r guichets ouvrent simultanément à l'instant $t=0$. Les r premières personnes se répartissent sur les différents guichets.
Pour $1 \leq i \leq r$ on note X_i la durée de service du i-ème client de la file d'attente. On suppose que les variables aléatoires X_i pour $1 \leq i \leq r$ sont mutuellement indépendantes et suivent toutes la même loi exponentielle de paramètre $\mu > 0$.

1) a) Soit $a > 0$. Soit X une variable aléatoire suivant la loi uniforme sur le segment $]0,1[$.

Déterminer la loi de la variable aléatoire $Y = -\frac{1}{a}\ln(X)$.

b) Ecrire en Python une fonction **expo** prenant en entrée un réel $a > 0$ et qui renvoie une valeur simulée d'une variable aléatoire suivant la loi exponentielle de paramètre a.
Cette fonction pourra être utilisée pour traiter les autres questions d'informatique de la **partie A**.

2) Soit Y_r le temps d'attente de la $(r+1)$-ième personne avant que l'un des r guichets se libère.

a) Exprimer Y_r en fonction des variables aléatoires $X_1, X_2, ..., X_r$.

b) Déterminer $P(Y_r > x)$ pour tout réel positif x.

c) Montrer que Y_r suit une loi exponentielle dont on précisera le paramètre.

3) Soit Z_r le temps nécessaire pour que chacun des r premiers clients soit servi.

a) Déterminer la fonction de répartition de Z_r.

b) En déduire que Z_r est à densité et donner une densité de Z_r.

c) En considérant une variable aléatoire suivant une loi exponentielle de paramètre $a > 0$, donner sans calcul la valeur de l'intégrale $\int_0^{+\infty} axe^{-ax}dx$.

d) Montrer que Z_r admet une espérance donnée par : $E(Z_r) = \int_0^{+\infty} x\mu r e^{-\mu x}\left(1-e^{-\mu x}\right)^{r-1} dx$.

Partie II : le cas d'un guichet unique

On étudie dans cette partie la file d'attente qui se forme à un seul guichet.
On suppose qu'à un instant initial $t=0$ il n'y a aucun client. On pose $M_0=0$ et pour tout $i\in\mathbb{N}^*$ on note M_i l'instant d'arrivée du i-ème client. Pour tout $i\in\mathbb{N}^*$ on note $A_i=M_i-M_{i-1}$. Ainsi A_i correspondant à l'intervalle de temps entre le $(i-1)$-ème et le i-ème client si $i\geq 2$, et $A_1=M_1$.
On suppose que les variables aléatoires $A_1, A_2, ..., A_n, ...$ sont indépendantes et suivent toutes la même loi exponentielle de paramètre λ strictement positif.
On appelle encore durée du service d'une personne au guichet le temps passé par l'employé au guichet à la servir, une fois son attente dans la file achevée.
On suppose ici que les variables aléatoires indiquant les durées de service au guichet des différentes personnes suivent la même loi exponentielle de paramètre $\mu>0$.
On admettra que les durées de service de chaque personne sont des variables aléatoires qui forment, avec les variables aléatoires $(A_n)_{n\in\mathbb{N}^*}$ une famille de variables aléatoires mutuellement indépendantes.

1) a) Pour tout $n\in\mathbb{N}^*$ exprimer la variable aléatoire M_n en fonction des variables $A_1, ..., A_n$.

b) En déduire l'espérance et la variance de M_n pour tout $n\in\mathbb{N}^*$.

c) A l'aide de l'inégalité de Bienaymé-Tchebychev montrer pour tout $n\in\mathbb{N}^*$ l'inégalité :

$$P\left(\frac{n}{2\lambda}\leq M_n\leq\frac{3n}{2\lambda}\right)\geq 1-\frac{4}{n}.$$

d) Ecrire en Python une fonction **M** prenant en entrée un entier naturel non nul n et un paramètre *lam* strictement positif et renvoyant une valeur simulée de la variable aléatoire M_n.

e) Ecrire en Python une fonction **EstimProba** prenant en entrée un entier naturel non nul n, un paramètre réel *lam* strictement positif et un entier naturel non nul *nbsim* qui renvoie une estimation de la probabilité $P\left(\frac{n}{2\lambda}\leq M_n\leq\frac{3n}{2\lambda}\right)$ obtenue à l'aide de *nbsim* simulations.

f) En exécutant la fonction **EstimProba** l'instruction print(**EstimProba**(40,1,10000)) renvoie une valeur de 0,98 à 10^{-2} près. Commenter ce résultat.

2) Soient A une variable aléatoire suivant la loi exponentielle de paramètre λ et B une variable aléatoire suivant la loi exponentielle de paramètre μ.

a) Ecrire en Python une fonction permettant d'estimer la probabilité $P(A<B)$ à partir des valeurs des paramètres λ et μ.

b) Déterminer la loi de la variable aléatoire C définie par $C=-B$.

c) On rappelle que si U et V sont des variables à densité indépendantes admettant pour densité respectivement f et g, alors $U+V$ admet pour densité la fonction h définie par :

$$\forall x\in\mathbb{R},\ h(x)=\int_{-\infty}^{+\infty}f(t)g(x-t)dt.$$

A l'aide de ce rappel, déterminer une densité de la variable aléatoire $A-B$. On vérifiera que l'on obtient $h(x)=\frac{\lambda\mu}{\lambda+\mu}e^{\mu x}$ pour tout $x\leq 0$, et on donnera la valeur de $h(x)$ lorsque $x>0$.

d) En déduire : $P(A<B)=\frac{\lambda}{\lambda+\mu}$.

e) Comparer $P(A<B)$ et $\frac{1}{2}$ en fonction des paramètres λ et μ.

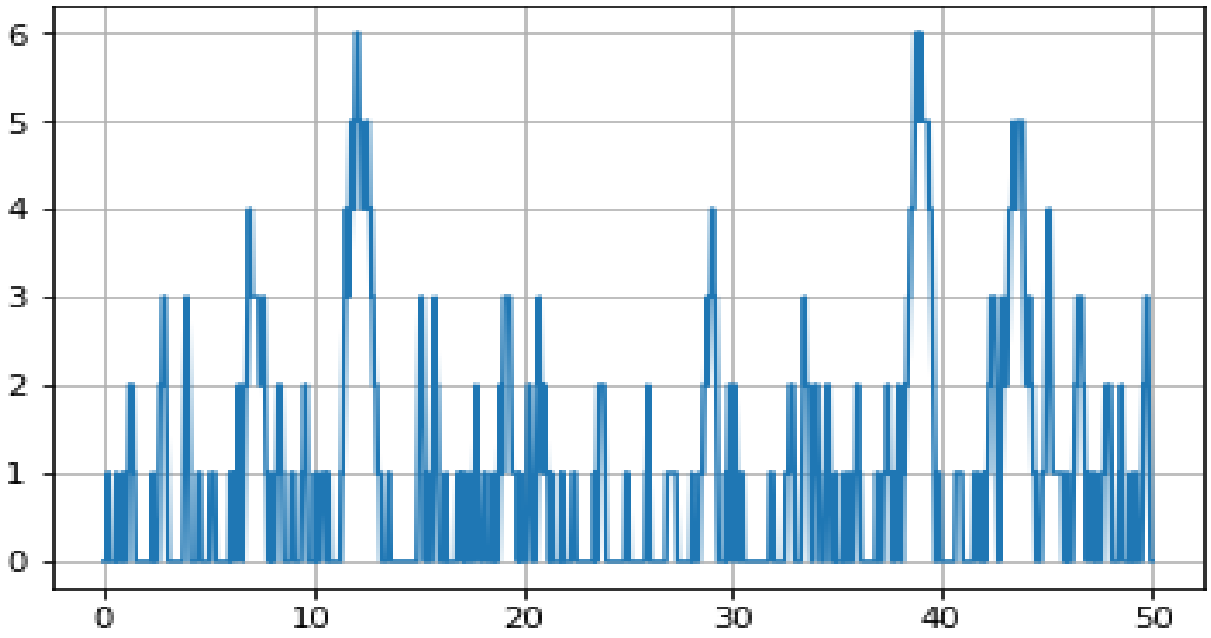
3) a) Ecrire en Python une fonction **arrivees** prenant en entrée un réel $T>0$ et un paramètre *lam* >0 qui renvoie la liste des instants d'arrivée simulés des clients éventuels se présentant entre les instants 0 et T (on renverra la liste vide s'il ne se présente aucun client entre les instants 0 et T, et on note *lam* le réel λ).

b) Expliquer ce que fait la fonction suivante et le rôle des différentes variables :

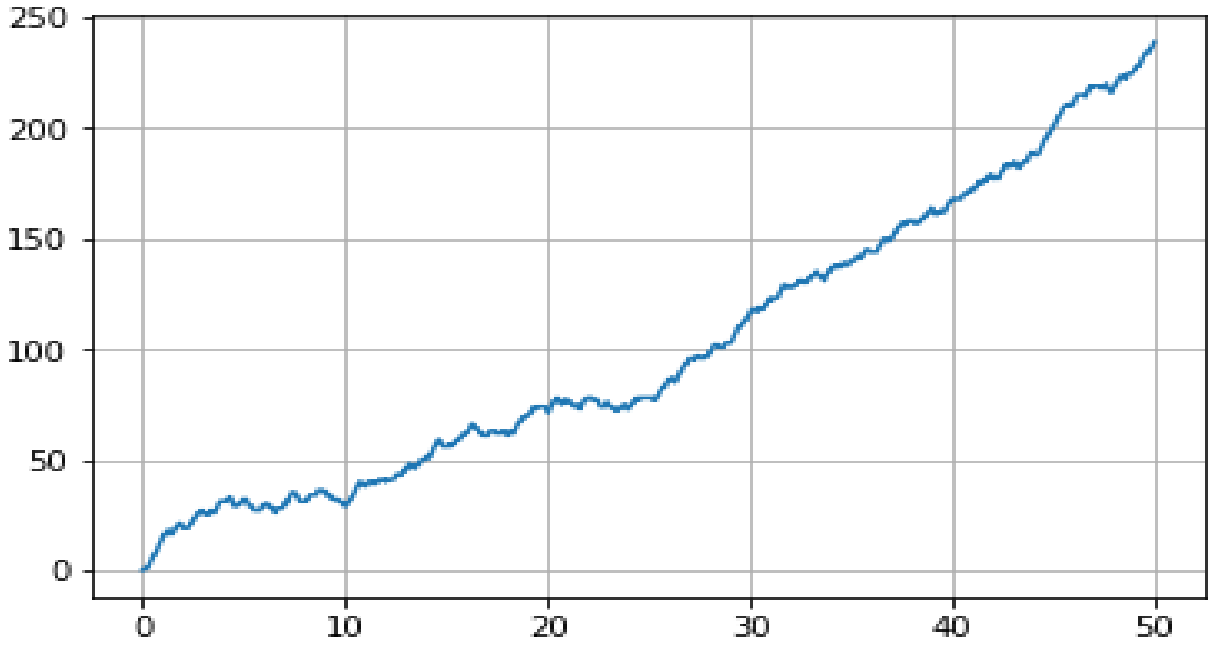
```
def départs (  T, arrivees, mu) :
    t = 0     #  par la suite  t  représente l'instant de départ du client précédent
    L = [ ]
    for a in arrivees :
        s = expo (mu)
        if  a > t :
            t = a + s
        else :
            t = t + s
        if   t <  T :
           L.append(t)
    return L
```

c) A l'aide de ces deux fonctions on a tracé les graphes suivants représentant les nombres de clients présents au guichet au cours du temps (entre les instants 0 et 50)

Graphe obtenu avec les valeurs $\lambda=10$ **et** $\mu=5$:



Graphe obtenu avec les valeurs $\lambda = 5$ **et** $\mu = 10$:



En s'appuyant sur le résultat de la question II) 2) e) commenter ces deux graphiques.

B) : le modèle discret

Partie I : étude de suites

Soit $a > 0$. On considère la suite $(u_n)_{n\in\mathbb{N}}$ définie par son premier terme $u_0 = 0$ et par la relation de récurrence : $\forall n \in \mathbb{N}$, $u_{n+1} = e^{a(u_n - 1)}$.

On considère la fonction f définie sur l'intervalle $[0,1]$ par : $\forall x \in [0,1]$, $f(x) = e^{a(x-1)}$.

I) 1) a) Déterminer le sens de variation de la fonction f sur l'intervalle $[0,1]$.

b) Montrer par récurrence : $\forall n \in \mathbb{N}$, $0 \le u_n \le u_{n+1} \le 1$.

c) Montrer que la suite $(u_n)_{n\in\mathbb{N}}$ converge. On note alors l sa limite.

I) 2) On se place dans cette question 2) uniquement dans le cas où $a = 1$.

a) Démontrer : $\forall x \in [0,1[$, $e^{x-1} > x$.

b) En déduire que l'équation $f(x) = x$ admet sur l'intervalle $[0,1]$ une solution unique que l'on précisera.

c) Quelle est la limite de la suite $(u_n)_{n\in\mathbb{N}}$?

I) 3) On se place dans cette question 3) uniquement dans le cas où $a = 2$.

I) 3) a) Démontrer pour tout $x \in]0,1]$ l'équivalence : $\left(f(x) = x\right) \Leftrightarrow \left(\ln x - 2x + 2 = 0\right)$.

b) Etudier sur l'intervalle $]0,1]$ la fonction $h : x \mapsto \ln x - 2x + 2$.

c) En déduire que l'équation $f(x) = x$ admet sur l'intervalle $[0,1]$ exactement deux solutions, le réel 1 et un réel de l'intervalle $\left]0, \frac{1}{2}\right[$ que l'on notera α.

d) Démontrer par récurrence : $\forall n \in \mathbb{N}$, $u_n \in [0, \alpha]$. Quelle est la limite de la suite $(u_n)_{n\in\mathbb{N}}$?

e) Démontrer que l'on a : $\forall x \in [0, \alpha]$, $0 \le f'(x) \le \frac{2}{e}$.

f) A l'aide du théorème des accroissements finis appliqué à la fonction f, démontrer les inégalités :

$$\forall n \in \mathbb{N},\ 0 \le \alpha - u_{n+1} \le \frac{2}{e}(\alpha - u_n).$$

g) En déduire : $\forall n \in \mathbb{N}$, $0 \le \alpha - u_n \le \left(2e^{-1}\right)^n$.

h) (i) Ecrire en Python une fonction **suite** prenant en paramètre un entier naturel n qui renvoie le terme d'indice n de la suite $(u_n)_{n\in\mathbb{N}}$.

(ii) Définir en Python une fonction **approx** prenant en entrée un réel strictement positif **eps** et qui renvoie une valeur approchée de α à **eps** près.

Partie II :étude d'une file d'attente

On se place dans un espace probabilisé $(\Omega, \mathcal{A}, P)$ et on modélise le temps par une succession d'instants $0, 1, 2, ..., n, ...$ On considère un guichet auquel peut se présenter au plus une personne dans l'intervalle de temps compris entre deux instants successifs $[n-1, n[$ pour tout entier naturel non nul n.

On suppose qu'une première personne est au guichet à l'instant 0 et, pour tout entier naturel non nul n, on désigne par B_n la variable aléatoire prenant la valeur 1 si une personne se présente au guichet pendant l'intervalle de temps $[n-1, n[$ et égale à 0 sinon. La personne ainsi arrivée se place au bout de la file d'attente mise en place devant le guichet.

Ces variables aléatoires $B_1, B_2, ..., B_n, ...$ sont supposées indépendantes et suivent toutes la même loi de Bernoulli de paramètre p, avec $0 < p < 1$.

On appelle durée du service d'une personne au guichet le temps passé par l'employé au guichet à la servir, une fois son attente dans la file achevée. Plus précisément, si la durée du service de la première personne, arrivée à l'instant 0, est égale à n, le guichet est libre pour la personne suivante à partir de l'instant n.

On admettra que les durées de service de chaque personne sont des variables aléatoires discrètes qui forment, avec les variables aléatoires $(B_n)_{n\in\mathbb{N}^*}$ une famille de variables aléatoires mutuellement indépendantes.

On suppose de plus que les variables aléatoires indiquant les durées de service au guichet des différentes personnes suivent la même loi de Poisson de paramètre $\mu > 0$.

On notera D la durée du service du client initial, présent à l'instant 0.

On convient d'appeler première vague l'ensemble des personnes arrivées au guichet pendant la durée de service du client initial, puis, de façon plus générale, on appelle $(k+1)$-ième vague l'ensemble des personnes arrivées pendant le temps de service des personnes de la k-ième vague. On désigne alors par N_k le nombre de personnes de la k-ième vague, étant entendu que l'on pose $N_k = 0$ s'il n'y a personne dans la k-ième vague. On pose par convention $N_0 = 1$.

II) 1) Simulation en Python de la variable aléatoire N_1.

La fonction **random.poisson (lam)** du module **numpy** simule pour un paramètre d'entrée **lam** strictement positif une variable aléatoire suivant une loi de Poisson de paramètre **lam**.
On suppose avoir importé cette fonction sous le nom de **Poisson**. Le réel μ sera noté **mu**.

a) Définir en Python une fonction **Bernoulli** prenant en entrée un paramètre réel p tel que $0 < p < 1$ et qui simule une variable aléatoire suivant une loi de Bernoulli de paramètre p.

b) Définir en Python une fonction **N1** prenant en entrée deux paramètres réels **mu** et p avec **mu** strictement positif et $0 < p < 1$ et qui renvoie une valeur simulée de la variable aléatoire N_1.

c) Définir en Python une fonction **moyN1** trois paramètres dont deux réels **mu** et p avec **mu** strictement positif et $0 < p < 1$ et un entier naturel non nul *nbsim* qui renvoie la moyenne de *nbsim* simulations indépendantes de la variable aléatoire N_1.

II) 2) Etude de la loi de la variable aléatoire N_1.

a) Etant donné un entier naturel n, déterminer la loi conditionnelle de la variable aléatoire N_1 sachant réalisé l'événement $(D = n)$.

b) Donner la valeur de la probabilité conditionnelle $P(N_1 = k / D = n)$ pour tous les couples d'entiers naturels (n,k) tels que $k > n$.

c) On pose $q = 1 - p$.
Pour tout couple (n,k) tels que $k \leq n$, justifier l'égalité :

$$P\big((N_1 = k) \cap (D = n)\big) = \binom{n}{k} p^k q^{n-k} \frac{\mu^n}{n!} e^{-\mu}.$$

d) En déduire que N_1 suit une loi de Poisson de paramètre μp.

e) En exécutant la fonction **moyN1** créée en **II) 1)** l'instruction print(moyN1(10,0.2,10000)) renvoie pour valeur 2.0322. Commenter ce résultat.

II) 3) Probabilité que la file d'attente s'achève au bout d'un temps fini

a) On pose, pour tout k entier naturel, $p_k = P(N_k = 0)$.
Démontrer que l'événement A : « la file d'attente s'achève au bout d'un temps fini » (autrement dit : il n'y a plus personne au guichet au bout d'un temps fini) est la réunion des événements $(N_k = 0)$ pour $k \in \mathbb{N}^*$.

b) Démontrer que la suite $(p_k)_{k\in\mathbb{N}}$ est croissante, puis qu'elle converge vers un réel $L \leq 1$.

c) Pour tout k entier naturel, on considère l'événement $R_k = (N_{k+1} = 0) \cap (N_k \neq 0)$.

On a donc en particulier : $R_0 = (N_1 = 0)$.

(i) Montrer que : $\forall k \in \mathbb{N}$, $P(R_k) = p_{k+1} - p_k$.

(ii) Démontrer l'égalité $A = \bigcup_{k\in\mathbb{N}} R_k$, puis en déduire que $P(A) = L$.

d) Justifier par un raisonnement probabiliste pour tout couple d'entiers naturels (j,k) les formules :

$P(N_{k+1} = 0 / N_1 = 1) = P(N_k = 0)$ et $P(N_{k+1} = 0 / N_1 = j) = P(N_k = 0)^j$.

e) En déduire, pour tout k entier naturel, la relation : $p_{k+1} = e^{\mu p (p_k - 1)}$.

f) En utilisant les résultats de la partie **B) I)** , montrer que si $\mu p = 1$ la file d'attente s'achève presque sûrement au bout d'un temps fini. Quelle est la probabilité de A si $\mu p = 2$?

g) On souhaite définir en Python une fonction **testA** prenant en entrée trois paramètres réels *mu*, p et $t_{\max}$, avec mu strictement positif, $0 < p < 1$ et $t_{\max}$ strictement positif, qui simule l'expérience décrite et renvoie 0 si la file d'attente n'est toujours pas achevée à l'instant $t_{\max}$ et 1 sinon.

Compléter le programme suivant pour que la fonction **testA** réponde à la question :

```
def testA (mu, p, tmax) :
    n = 1
    t = 0
    while …… :
        n = n - ……
        duree = Poisson (……)
        for k in range (……) :
            t = t + ……
            n = n + Bernoulli (……)
            if t == tmax :
                if n != …… :
                    return ……
                else:
                    return ……
    return ……
```

h) On définit alors (l'écriture de cette fonction n'est pas demandée) une fonction **moyA** prenant en entrée quatre paramètres réels mu , p , $t_{\max}$ et m , avec mu strictement positif, $0 < p < 1$, $t_{\max}$ strictement positif, et m entier naturel non nul, qui renvoie la moyenne des valeurs prises par la fonction **testA** lors de m simulations indépendantes.

En exécutant les fonctions précédentes on obtient les résultats suivants :

moyA(2, 0.5 , 1000, 10000) = 0.967

moyA(4, 0.5, 1000, 10000) = 0.1974

L'exécution de la fonction **approx** de la partie **A** donne :

approx(2, 0.01) = 0.20318778722911182

Commenter ces résultats.

II) 4) Simulation des variables aléatoires N_k

a) Ecrire en Python une fonction **binomiale** prenant en entrée deux paramètres n et p avec $n \in \mathbb{N}^*$ et $p \in]0,1[$ qui renvoie une valeur simulée d'une variable aléatoire suivant la loi binomiale de paramètres n et p.

b) Compléter la définition de la fonction **N** suivante prenant en entrée trois paramètres k, mu et p, avec $k \in \mathbb{N}^*$, mu strictement positif et $p \in]0,1[$ pour qu'elle renvoie une valeur simulée de la variable aléatoire N_k définie plus haut, le réel μ étant noté *mu* :

```
def N (k, mu, p):
    n = 1
    for j in range( ….. ):
        s = 0
        for i in range( ….. ):
            s = s + Poisson (mu)
        n = binomiale ( ….. , …..)
    return n
```

c) On crée en Python (l'écriture de cette fonction n'est pas demandée) une fonction **moyN0** prenant en entrée quatre paramètres k, mu et p, avec $k \in \mathbb{N}^*$, mu strictement positif, $p \in]0,1[$ et $m \in \mathbb{N}^*$, qui renvoie la moyenne du nombre de fois où l'événement $(N_k = 0)$ est réalisé au cours de m simulations indépendantes de la variable aléatoire N_k.

L'exécution de la fonction précédente et de la fonction **suite** créée en **B) I) 3)** donne :

pour k= 0 moyN0(k,4,0.5,10000)= 0.0 suite(k)= 0
pour k= 1 moyN0(k,4,0.5,10000)= 0.1376 suite(k)= 0.1353352832366127
pour k= 2 moyN0(k,4,0.5,10000)= 0.1826 suite(k)= 0.17740333081914023
pour k= 3 moyN0(k,4,0.5,10000)= 0.1976 suite(k)= 0.1929752496682254
pour k= 4 moyN0(k,4,0.5,10000)= 0.1982 suite(k)= 0.19907980576335968
pour k= 5 moyN0(k,4,0.5,10000)= 0.1999 suite(k)= 0.20152529167524172

Commenter ces résultats.