

Exercice 01

L'objectif de cet exercice est d'écrire un programme python pour mesurer les performances d'une file d'attente M/M/1 où les clients arrivent selon une loi de Poisson paramètre λ , et sont servis par un serveur unique selon une loi exponentielle μ .

- Ecrire un programme qui reçoit en entrée le taux d'arrivé des clients λ et le taux de service μ , puis, calcule et affiche les mesures suivantes :
 - Le coefficient d'utilisation du serveur ρ
 - Nombre moyen de clients dans le système (\bar{N})
 - Nombre moyen de clients dans la file (\bar{N}_Q)
 - Nombre moyen de clients en service (\bar{N}_S)
 - Temps d'attente moyen dans le système (\bar{T})
 - Temps d'attente moyen dans la file d'attente (\bar{T}_Q)
 - Temps moyen de service d'un client (\bar{T}_S)

Remarque : Implémenter les mesures précédente sous forme de fonctions python : $\rho()$, $p_0()$, $N()$, N_q , $N_s()$, $T()$, T_q , $T_s()$.

- Tester le programme avec le jeu de test suivants : exécuter votre programme avec les données de test, et comparer les résultats obtenus avec les résultats attendus

Données de test		Résultats attendus							
λ	μ	ρ	P_0	\bar{N}	\bar{N}_Q	\bar{N}_S	\bar{T}	\bar{T}_Q	\bar{T}_S
1	2	0.5	0.5	1.0	0.5	0.5	1.0	0.5	0.5
1.5	2	0.75	0.25	3.0	2.25	0.75	2.0	1.5	0.5
4	5	0,8	0.2	4.0	3.2	0.8	1	0.8	0,2
6	5		Le système n'est pas stationnaire : $\rho = 1.2 > 1$						
4	6	0,667	0.333	2.0	1.333	0.667	0,5	0,333	0,167
2	3	0,667	0.333	2.0	1.333	0.667	0,5	0,333	0,167
3	2	1,5	Le système n'est pas stationnaire : $\rho = 1.5 > 1$						

- Implémenter la fonction $p_n(n)$ qui prend en paramètre un entier $n \geq 0$, puis, calcule et retourne la probabilité $P(N = n)$.
- Ajouter le code pour afficher un graphique des probabilités $P(N \geq n)$, pour $N = 1, 2, \dots, n_{max}$. n_{max} est un entier positif saisi par l'utilisateur.

Exercice 02

L'objectif de cet exercice est d'écrire un programme python pour mesurer les performances d'une file d'attente M/M/S, les clients arrivent selon une loi de Poisson paramètre λ , et sont servis par un nombre de serveurs S unique selon une loi exponentielle de paramètre μ .

1. Ecrire un programme qui reçoit en entrée le taux d'arrivé des clients λ et le taux de service μ , puis, calcule et affiche les mesures suivantes :
 - Le coefficient d'utilisation du système α
 - La probabilité que le système est vide (Aucun client dans le système) : P_0
 - La probabilité d'attente pour un client (la formule d'Erlang C): P_w
 - Nombre moyen de clients dans le système (\bar{N})
 - Nombre moyen de clients dans la file (\bar{N}_Q)
 - Nombre moyen de clients en service (\bar{N}_S)
 - Temps d'attente moyen dans le système (\bar{T})
 - Temps d'attente moyen dans la file d'attente (\bar{T}_Q)
 - Temps moyen de service d'un client (\bar{T}_S)

Remarque : Implémenter les mesures précédente sous forme de fonctions python : `rho()`, `a()`, `p_0()`, `erlang_C()`, `N()`, `Nq`, `Ns()`, `T()`, `Tq`, `Ts()`.

2. Tester le programme avec le jeu de test suivants : exécuter votre programme avec les données de test, et comparer les résultats obtenus avec les résultats attendus

Données de test			Résultats attendus								
λ	μ	S	α	P_0	P_w	\bar{N}	\bar{N}_Q	\bar{N}_S	\bar{T}	\bar{T}_Q	\bar{T}_S
10	4	3	0.833	0.045	0.702	6.011	3.511	2.5	0.601	0.351	0.25
60	18	4	0.833	0.021	0.658	6.622	3.289	3.333	0.11	0.055	0.056
30	20	2	0.75	0.143	0.643	3.429	1.929	1.5	0.114	0.064	0.05
60	25	2	1.2	Le système n'est pas stationnaire : $a = 1.2 >= 1$							
16	4	4	1.0	Le système n'est pas stationnaire : $a = 1.0 >= 1$							

3. Implémenter la fonction `p_n(n)` qui prend en paramètre un entier $n \geq 0$, puis, calcule et retourne la probabilité $P(N = n)$.
4. Ajouter le code pour afficher un graphique des probabilités $P(N \geq n)$, pour $N = 1, 2, \dots, n_{\max}$. n_{\max} est un entier positif saisi par l'utilisateur.