

Exercices REGEX

Question 1:

Le fichier « 1_access.log » est un fichier de log d'un serveur web. Nous sommes intéressé par les lignes pour lesquelles tout ne s'est pas passé correctement. Le code d'une page servie sans problème est 200. Il nous faut donc toutes les lignes pour lesquels le code n'est PAS 200. De plus, nous ne voulons que les lignes pour lesquelles le client est une machine de l'école. C'est simple à trouver, car les machines de l'école ont une adresse IP qui COMMENCE par « 100 ».

Dans le fichier, pour chaque ligne, l'adresse IP du client est la première info, et le code d'erreur est la pénultième info (vous avez le droit de chercher sur google ce que veut dire « pénultième »).

Le but de la question est de trouver, dans le fichier de log « 1_access.log », le nombre de lignes où tout ne s'est pas passé correctement et où le client est une machine de l'école.

Gardez ce nombre, puis, à grand coup de head et de tail, retrouvez dans le fichier « 1_responses.txt » la ligne correspondante au nombre de lignes trouvé précédemment (par exemple si vous avez trouvé 666 lignes dans le fichier de log, alors affichez la ligne numéro 666). Mettez cette ligne dans votre copie.

N'oubliez pas de décoder la ligne avec la commande « base64 » pour voir si vous ne vous êtes pas trompé.

Question 2:

Un peu de détente, faisons un peu de mot croisés. Ou plutôt, d'expressions régulières croisées.

Le but : remplir cette grille de « mots croisés » version regex. Le but est de remplir chaque case par un chiffre qui répond à la fois à la regex verticale et à la regex horizontale de sa case :

	[[^] 16243]		1 3 5 7 9	
4 2 6 8 3				
[40612]				

Une fois que vous aurez rempli les 4 cases avec 4 chiffres, vous aurez un nombre à 4 chiffres en lisant d'abord les 2 cases du haut de gauche à droite, puis les 2 cases du bas de gauche à droite.

A grand coup de head et de tail, comme pour la question 1 retrouvez dans le fichier « 2_responses.txt » la ligne correspondante au nombre trouvé ainsi.

N'oubliez pas de décoder la ligne avec la commande « base64 » pour voir si vous ne vous êtes pas trompé.

Question 3:

Le fichier « 3_access.log » est un autre fichier de log d'un serveur web. Nous sommes intéressé par les heures des connexions, et plus précisément par les minutes. En fait, il nous faudrait toutes les connexions pour lesquelles les minutes ne sont PAS égales à 24 : par exemple, on prendra les connexions de 1h23, 1h25, mais pas de 1h24. De plus, nous ne voulons que les lignes pour lesquelles le client a une adresse IP qui COMMENCE par « 193 ».

Dans le fichier, pour chaque ligne, l'adresse IP du client est la première info, et la date et heure est la seconde info (après les tirets).

Le but de la question est de trouver, dans le fichier de log « 3_access.log », le nombre de lignes correspondants à ces critères : ayant une adresse IP qui commence par « 193 » et dont les minutes de connexions ne sont pas « 24 ».

Gardez ce nombre, puis, à grand coup de head et de tail, comme pour la question précédente, retrouvez dans le fichier « 3_responses.txt » la ligne correspondante au nombre de lignes trouvé précédemment (par

exemple si vous avez trouvé 666 lignes dans le fichier de log, alors affichez la ligne numéro 666). Mettez cette ligne dans votre copie.

N'oubliez pas de décoder la ligne avec la commande « base64 » pour voir si vous ne vous êtes pas trompé.

Question 4:

Le but : remplir cette grille de « mots croisés » version regex. Le but est de remplir chaque case par un chiffre qui répond à la fois à la regex verticale et à la regex horizontale de sa case :

	[^1342807]	[5-9]
[0-5]		
[40612]		

Une fois que vous aurez rempli les 4 cases avec 4 chiffres, vous aurez un nombre à 4 chiffres en lisant d'abord les 2 cases du haut de gauche à droite, puis les 2 cases du bas de gauche à droite.

A grand coup de head et de tail, comme pour les questions précédentes, retrouvez dans le fichier « 4_responses.txt » la ligne correspondante au nombre trouvé ainsi.

N'oubliez pas de décoder la ligne avec la commande « base64 » pour voir si vous ne vous êtes pas trompé.

Question 5:

Comme pour la question suivante, remplissez cette grille de regex-croisée :

	[^1625380]	1 2 5 7 9
4 2 6 8 3		
[80637]		

Une fois que vous aurez rempli les 4 cases avec 4 chiffres, vous aurez un nombre à 4 chiffres en lisant d'abord les 2 cases du haut de gauche à droite, puis les 2 cases du bas de gauche à droite.

A grand coup de head et de tail, comme pour les questions précédentes, retrouvez dans le fichier « 5_responses.txt » la ligne correspondante au nombre trouvé ainsi.

N'oubliez pas de décoder la ligne avec la commande « base64 » pour voir si vous ne vous êtes pas trompé.