

Décrivez par un exemple (suivi pas à pas) ce que fait le script suivant (8 points) :

Pour chaque ligne d'un fichier dont le nom est passé en paramètre (`sys.argv[1]`)

Application d'une expression régulière à partir de chaque caractère de cette ligne (`ligne[p]`) pour extraire **la plus grande chaîne de caractères constituée des répétitions successives de ce caractère** :

```
while p < longueur - 1 :  
    r = re.search("(" + ligne[p] + "+)", ligne[p:])  
    ...  
    p = p+1
```

Si ce caractère est répété successivement plus d'une fois :

écriture dans le fichier résultat de ce caractère suivi du nombre de répétitions entre | :

```
if l > 1 :  
    fdr.write(ligne[p] + "|" + str(l) + "|") # écriture dans un fichier
```

Sinon réécriture du caractère

Affichage du nombre de caractères du fichier initial, du fichier résultat et du gain

Exemple :

abccdeefgghhi

 →

abc 3 de 2 fg 2 h 2 i

 → 14 - 21 = -7 (sans les \n et pour cet ex. le gain est négatif (voir plus loin))

A quoi sert ce script et comment pourriez-vous l'améliorer (donnez juste quelques idées) ? (3 points)

Ce programme est un programme naïf de compression.

- ne factoriser les caractères qu'à partir de 5 (aaaaa → a|5|) → avant ce n'est pas la peine
- opérer la recherche des caractères successifs sur plusieurs lignes
- prendre en compte la répétition de motifs
- ...

Ecriture d'entêtes `#!/usr/bin/python2` ou `perl` dans des fichiers qui n'en possèdent pas (9 points)

```
#!/usr/bin/env python2  
import sys, re  
suffixes = {'py' : '#!/usr/bin/python2\n',  
            'pl' : '#!/usr/bin/perl\n'}  
  
for nom_fichier in sys.argv[1:] :  
    resultat = re.search("(\\...)$", nom_fichier)  
    if resultat and suffixes.has_key(resultat.group(1)) :  
        print nom_fichier, " -> script python ou script perl"  
        fd = open(nom_fichier, "r")  
        contenu_fichier = fd.readlines()  
        fd.close()  
        if contenu_fichier[0] != suffixes[resultat.group(1)] :  
            fdr = open(nom_fichier, "w")  
            fdr.write(suffixes[resultat.group(1)])  
            fdr.writelines(contenu_fichier)  
            fdr.close()  
        else :  
            print nom_fichier, "    le fichier a déjà l'entête nécessaire"  
    else :  
        print nom_fichier, " -> fichier n'ayant pas de suffixe ou possédant un suffixe inconnu"
```