Faculté des Sciences & Techniques — Université de Limoges

Master 1ère année

Protocoles & Prog. Réseau

TD n°1

Maîtrise de Python

Manipulation de fichiers

1 – Écrire un programme qui compte le nombre de lignes d'un fichier sur disque.

2 – Écrire un programme qui ouvre un premier fichier et crée un nouveau fichier contenant une ligne sur deux du premier fichier.

Gestion des listes

3 – Écrire un programme prenant la liste des fichiers contenus dans un répertoire, et qui ouvre et affiche la première ligne de chacun de ces fichiers.
La liste des fichiers d'un répertoire est obtenue par l'instruction : `ls *.html`

4 – Écrire un programme qui réalise l'insertion d'une liste d'éléments dans une liste existante à un emplacement donné par son indice.

Utilisation des expressions rationnelles ou *Regular Expression* & Connexion TCP

5 – Écrire un programme réalisant le travail du serveur TCP suivant :

a. attente de la connexion d'un client ;
b. affichage du TSAP du client ;
c. réception d'un message en provenance d'un client sous la forme ACCESS : toto@unilim.fr ;
d. affichage à l'écran du texte : Acces de toto ;
e. fermeture de la connexion et retour en a.

6 – Écrire un programme réalisant du « *banner grabbing* », c-à-d de la capture de bannière d'accueil, pour les services SMTP, SSH, POP pour une machine donnée par son adresse symbolique et à l'aide d'une connexion TCP.
Exemple de « banner grabbing » avec l'outil de communication « à tout faire » `socat` *:*

```
xterm
pef@darkstar:~ socat stdio tcp:smtp.unilim.fr:25
220 smtp.unilim.fr ESMTP Exim 4.94.2 Mon, 12 Sep 2022 10:00:34 +0200
^C
```

On obtient la nature du serveur, ici « Sendmail » et son numéro de version.

7 – Écrire un programme qui récupère le titre d'une page html dans un fichier au format HTML.

8 – Écrire un motif pour la décomposition d'une URL (récupération des différents champs qui la compose) :
`service` `://` `adresse_machine` `:` `numéro_de_port` `/` `adresse_document`
Rappel : le numéro de port est optionnel, par défaut il correspond à la valeur 80 pour le service HTTP.

9 – Écrire un programme affichant le contenu d'une page HTML récupérée à l'aide d'une connexion TCP.

Exemple de récupération avec `socat` *:*

```
xterm
pef@darkstar:~ socat - tcp:www.unilim.fr:80
GET / HTTP/1.0

HTTP/1.1 301 Moved Permanently
Server: nginx
Date: Mon, 12 Sep 2016 10:55:34 GMT
Content-Type: text/html
Content-Length: 178
Connection: close
Location: https://ucloud.unilim.fr/

<html>
```

Manipulation des dictionnaires & opérations d'éclatement et de recomposition (split & join)

10 – Écrire un programme qui détermine le nombre d'occurrence de chaque mot d'un fichier texte.

Représentation hexadécimale

11 – Écrire un programme qui reproduit le traitement de la commande « `xxd` » du shell :

```
xterm
$ echo "Alice et Bob sont dans un bateau" | xxd
00000000: 416c 6963 6520 6574 2042 6f62 2073 6f6e  Alice et Bob son
00000010: 7420 6461 6e73 2075 6e20 6261 7465 6175  t dans un bateau
00000020: 0a                                       .
```

Mise en œuvre des instructions de manipulation binaire

12 – Écrire un programme qui :

a. ouvre un fichier texte en lecture ;
b. ouvre en écriture un nouveau fichier ;
c. traite chaque caractère de ce fichier de la manière suivante :
   - ◇ décomposition de ce caractère en sa représentation binaire
   - ◇ échange des bits de rang 3 & 4;
   - ◇ recomposition de cette représentation binaire en un caractère;
   - ◇ écriture de ce caractère dans le fichier.

13 – Utilisation d'un GCL, « générateur à congruence linéaire », est un générateur de nombres pseudo-aléatoires basé sur des congruences et une fonction affine :

$$X_{n+1} = (aX_n + c) \bmod m, \text{ où le terme } X_0 \text{ est appelé « } seed \text{ ».}$$

▷ *Pour chaque seed, on obtient une nouvelle suite de nombres.*
▷ *Les nombres de la suite ont l'apparence de l'aléas.*
▷ *Cette suite est plus ou moins grande : tout nouveau nombre étant basé sur le précédent, si un nombre apparait une deuxième fois dans la suite alors la suite se répète entièrement à partir de ce nombre.*
▷ *Le nombre de valeurs de la suite étant fini, il dépend de m, la suite se répètera forcément.*
▷ *En utilisant, un même seed, on obtient la même séquence de nombres (d'où le nom de « pseudo »-aléatoire).*

Certaines valeurs bien choisies pour m, a et c permettent d'obtenir des séquences assez longues.
On utilisera les valeurs trouvées par Donald Knuth :

m	a	c
2^{64}	6364136223846793005	1442695040888963407

a. Quelle est la taille maximale des valeurs données par le générateur avec les paramètres de D. Knuth ?
b. L'opérateur binaire « xor » permet de combiner une séquence binaire S_a avec une séquence binaire S_b en inversant les bits de S_a de même rang que les bits à 1 de S_b, ce qui donne la séquence S_r.
   Vérifiez que si on combine avec un xor S_r avec S_b, on obtient bien S_a.
c. Écrire un programme de « chiffrement » permettant de combiner un message M avec une séquence de valeurs obtenues à l'aide du générateur à congruence linéaire pour un seed donné.
   Vous vérifierez que l'opération de déchiffrement est possible en utilisant le même seed.

14 – Écrire un programme réalisant l'encodage base64 d'un fichier conformément à la RFC 2045.