

Maîtrise de Python

■ ■ ■ Manipulation de fichiers

- 1 – Écrire un programme qui compte le nombre de lignes d'un fichier sur disque.
- 2 – Écrire deux programmes :
  - a. le premier qui effectue l'écriture de données dans un fichier texte (organisées sous forme de lignes de textes, où chaque ligne est composée de caractères et terminée par un `\n`).  
Ces informations sont le *nom*, *prénoms* et *adresse* d'un individu.  
Elles sont saisies au clavier et enregistrées dans le fichier « `carnet_adresse.txt` » :
    - une ligne par individu,
    - chaque ligne composée de la manière suivante : `<nom> : <prénoms> : <adresse>`
 Exemple : `Toto : Titi : 123 av Albert Thomas`
  - b. le second qui lit le fichier créé par le premier et affiche les informations *nom*, *prénom* et *adresse*, chacune sur une ligne, précédée de l'intitulé du champs (vous séparerez les infos de chaque individu par une ligne contenant des #).

■ ■ ■ 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 *`

- 4 – Écrire un programme permettant de vérifier que le numéro d'une carte de crédit est correct :

4	1	3	7	8	9	4	7	1	1	7	5	5	9	0	4
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

On double chaque valeur de rang impair :

8	1	6	7	16	9	8	7	2	1	14	5	10	9	0	4
1		3		5		7		9		11		13		15	
				1+6=7						1+4=5		1+0=1			

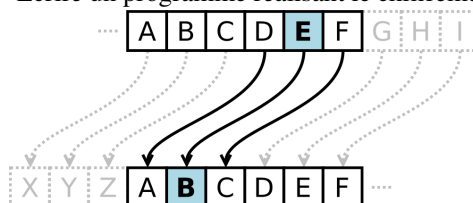
Si la valeur est sur 2 chiffres on fait la somme de ces deux chiffres.

D'où le résultat final :

8	1	6	7	7	9	8	7	2	1	5	5	1	9	0	4
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

On fait la **somme** de tous les chiffres obtenus :  $8 + 1 + 6 + 7 + 7 + 9 + 8 + 7 + 2 + 1 + 5 + 5 + 1 + 9 + 0 + 4 = 80$ , si le **dernier chiffre** est 0  $\Rightarrow$  le numéro de la carte est **valide**, sinon le numéro est invalide.

- 5 – Écrire un programme qui compte le nombre d'occurrences de chaque lettre de l'alphabet dans un texte et tri les caractères suivant l'ordre décroissant de ce nombre d'occurrences.
- 6 – Écrire un programme réalisant le chiffrement de César après saisie du décalage :

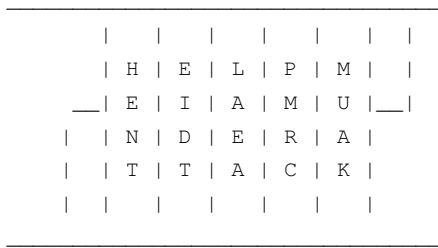
<table border="1"> <tr><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td><td>G</td><td>H</td><td>I</td></tr> </table>	A	B	C	D	E	F	G	H	I	Plain:    ABCDEFGHIJKLMNOPQRSTUVWXYZ Cipher:   XYZABCDEFGHIJKLMNOPQRSTUVWXYZ
A	B	C	D	E	F	G	H	I		
	Plaintext:  the quick brown fox jumps over the lazy dog Ciphertext: QEB NRFZH YOLTK CLU GRJMP LSBO QEB IXWV ALD									

- 7 – Écrire un programme d'aide à la cryptanalyse du chiffrement César à partir de l'analyse fréquentielle des caractères présents dans le chiffré et par comparaison aux fréquences les plus élevées des caractères de la langue française.

Vous déchiffrez le message suivant :

`BULML TTLWH ZZHKB ULTHP UMHA BLBZL ZVBSL CHUAI HSHUJ HUASL MLZAV ULASV BYSLA`

- 8 – Écrire un programme de déchiffrement du code César par la méthode « brute-force ».
- 9 – Écrire un programme réalisant le chiffrement par la méthode la « Scytale » grecque :



Le texte :

Help me I am under attack

devient :

HENTEIDTLAEAPMRCMUAK

- 10 – Écrire un programme permettant de déterminer si un message est chiffré suivant la méthode de la Scytale ou par le chiffrement de César.