Statistiques et algorithmique, 1

1 Activité proposée

1.1 Pré-requis, objectifs

L'activité proposée ici a été développée dans le contexte suivant :

- les notions de série statistiques et de fréquence ont été revues
- les élèves ont manipulés le logiciel Python, les boucles et test conditionnel sont connus

Les objectifs sont de découvrir les chaînes de caractères, leurs manipulation avec le logiciel et de les utiliser dans un contexte cryptographique pour faire des calculs de fréquence d'apparition de lettres.

1.2 Modalités

Cette activité peut être développée sur une séance d'une heure, en demi-groupe en salle informatique ou avec une classe mobile. Elle peut aussi être développée en classe entière et l'usage d'une classe mobile par équipe de deux élèves.
Le texte à décrypter sera rendu disponible aux élèves par voie numérique (réseau établissement, ENT) via un fichier texte par exemple. Un copier-coller permettra d'éviter la tâche fastidieuse et périlleuse de recopie.

Dans la question 5 du II.2, nous n'attendons pas un décryptage exhaustif en testant les 26 lettres. Le professeur doit être vigilant à bien cadrer ce moment pour que les élèves utilisent la fonction fréquence.

2 Commentaires sur la partie Aller plus loin

Cette partie est développée pour les élèves les plus rapides ou pour donner matière à des devoirs maisons. Elle permet de faire découvrir aux élèves la table ASCII et d'aborder la représentation de l'information.
Elle permet de réinvestir la notion de fonction algorithmique.

Cryptographie - "Chiffrement de Cesar"

Capacités attendues	**Mathématiques :** Manipuler les fréquences **Algorithmiques :** Utiliser chaîne de caractères Utiliser une fonction algorithmique

1 Manipuler les "chaînes de caractères" avec le langage Python

Dans le module Python, on définit deux variables de type str (string / chaîne de caractères) par :
texte1="**bonjour**" et **texte2**="**aujourd'hui il fait froid.**".
Utiliser la console pour tester et indiquer ce que fait chacune de ces commandes suivantes :

Commande	Description	Objectif
len(texte1) len(texte2)		
texte1[3] texte2[10] texte1[0] texte2[7] texte1[20] texte2[27]		
texte1[2 :4] texte2[0 :11]		
texte1+texte2		

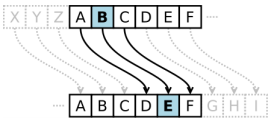
2 En cryptographie

2.1 Vocabulaire et principe

Chiffrement : procédé suivant lequel un document compréhensible de tous est transformé en un autre, incompréhensible.
Déchiffrement : procédé inverse permettant de revenir au document initial.

Un algorithme cryptographique : procédé mathématique utilisé pour le chiffrement et le déchiffrement. Il décrit pas à pas toutes les étapes nécessaires à la transformation du document. Il existe plein de méthodes de chiffrement.
Le chiffrement par décalage consiste à associer une lettre en effectuant un décalage d'un nombre de rangs choisi (appelé clé) :

- Décalage clavier français avec une clé de 1 : on décale les lettres de 1 rang vers la droite en regardant le clavier de l'ordinateur : A devient Z, Z devient E, E devient R, etc...
- Décalage alphabétique avec une clé de 1 : on décale les lettres de 1 rang vers la droite dans l'alphabet : A devient B, B devient C, C devient D, etc...
  Le chiffrement de César est un chiffrement par décalage alphabétique avec une clé de 3.



2.2 Au travail, espion !

Pour faciliter la mise en oeuvre nous allons utiliser l'alphabet réduit utilisé par César lui-même, c'est-à-dire les 26 lettres majuscules (les phrases utilisées ne devront donc contenir ni espace, ni accent, ni ponctuation).

Ouvrir le fichier **« Cesar.py »** que vous trouverez sur le réseau.

1. Combien y a-t-il de fonctions programmées ? Quels sont leurs arguments ? A votre avis, quel est le rôle de chacune de ces fonctions ?
2. En utilisant Python, après avoir compilé le script « Cesar.py » avec la flèche verte :
3. Donner le cryptogramme correspondant à votre prénom avec une clé de 10 :
4. Trouver la clé qui permet de décrypter le cryptogramme « GWFAT » :
5. On cherche maintenant à déchiffrer le message codé suivant (disponible sur le réseau) :
   MZYUZFCLFUZFCOSFTYZFDLGZYDOPNZFGPCEFYPXPESZOPAZFCOPNCJAEPCFYXPDDLRPNZOPLGPN
   WLXPESZOPOPNPDLCNPEEPXPESZOPMLDPDFCWPEFOPOPDQCPBFPYNPDPDEPQQTNLNPDAZFCNLDPC

EZFDWPDNZOPDXZYZDJWWLMTBFPDNPBFTCPYONPDNZOPDAPFQTLMWPDXLWRCPEZFETWDAPFGP
YEPECPFETWTDPDPQQTNLNPXPYEPYACPYLYEBFPWBFPDACPNLFETZYDPWPXPYELTCPD

On pourrait tester les 26 clés possibles mais nous allons chercher une méthode permettant de prévoir la bonne clé facilement en utilisant le document suivant :

Voici un tableau donnant les fréquences moyennes (en %) des lettres utilisées dans les textes écrits en français (dans les conditions définies plus haut) :

A	B	C	D	E	F	G	H	I	J	K	L	M
9.42	1.02	2.64	3.39	15.87	0.95	1.04	0.77	8.41	0.89	0.00	5.34	3.24
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
7.15	5.14	2.86	1.06	6.46	7.90	6.26	6.24	2.15	0.00	0.30	0.24	0.32

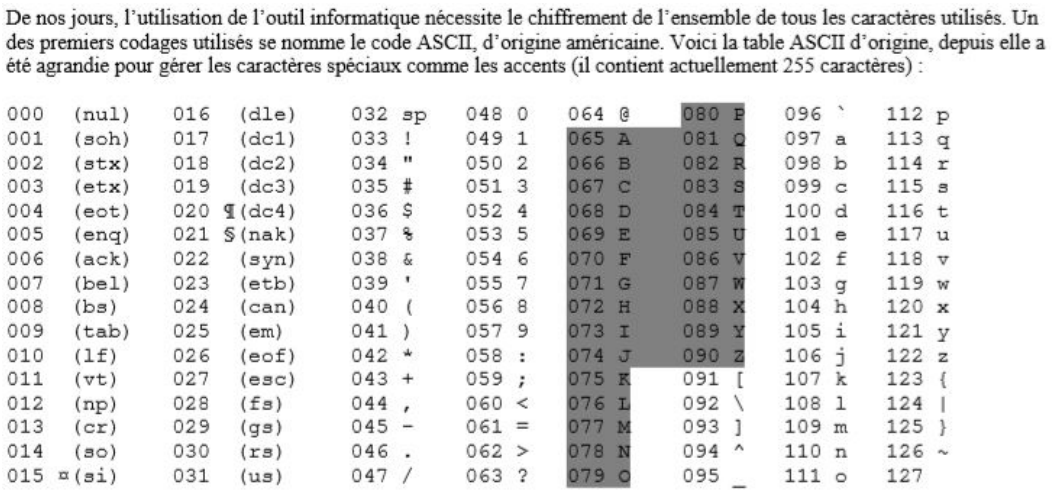
(a) Compléter l'algorithme suivant permettant de calculer la fréquence d'apparition (exprimée en %) d'une lettre dans un texte :

```
fonction frequence(texte,lettre)
    C ← 0
    Pour k de 1 à longueur(texte)
    Si texte[k]=lettre
        C ← C + 1
    FinSi
    FinPour
    Renvoyer .....
```

(b) Programmer cet algorithme et l'utiliser pour trouver la clé de déchiffrage du message codé ci-dessus.

2.3 Pour aller plus loin

1. De nos jours, l'outil informatique permet le chiffrement de l'ensemble de tous les caractères utilisés. La table ASCII utilisée pour cela contient actuellement 255 caractères.



De nos jours, l'utilisation de l'outil informatique nécessite le chiffrement de l'ensemble de tous les caractères utilisés. Un des premiers codages utilisés se nomme le code ASCII, d'origine américaine. Voici la table ASCII d'origine, depuis elle a été agrandie pour gérer les caractères spéciaux comme les accents (il contient actuellement 255 caractères) :

000	(nul)	016	(dle)	032	sp	048	0	064	@	080	P	096	`	112	p
001	(soh)	017	(dc1)	033	!	049	1	065	A	081	Q	097	a	113	q
002	(stx)	018	(dc2)	034	"	050	2	066	B	082	R	098	b	114	r
003	(etx)	019	(dc3)	035	#	051	3	067	C	083	S	099	c	115	s
004	(eot)	020	¶(dc4)	036	$	052	4	068	D	084	T	100	d	116	t
005	(enq)	021	§(nak)	037	%	053	5	069	E	085	U	101	e	117	u
006	(ack)	022	(syn)	038	&	054	6	070	F	086	V	102	f	118	v
007	(bel)	023	(etb)	039	'	055	7	071	G	087	W	103	g	119	w
008	(bs)	024	(can)	040	(	056	8	072	H	088	X	104	h	120	x
009	(tab)	025	(em)	041	)	057	9	073	I	089	Y	105	i	121	y
010	(lf)	026	(eof)	042	*	058	:	074	J	090	Z	106	j	122	z
011	(vt)	027	(esc)	043	+	059	;	075	K	091	[	107	k	123	{
012	(np)	028	(fs)	044	,	060	<	076	L	092	\	108	l	124	|
013	(cr)	029	(gs)	045	-	061	=	077	M	093	]	109	m	125	}
014	(so)	030	(rs)	046	.	062	>	078	N	094	^	110	n	126	~
015	¤(si)	031	(us)	047	/	063	?	079	O	095	_	111	o	127	

- La fonction *chr* renvoie le caractère associée à l'entier dans la table ASCII.
- La fonction *ord* renvoie l'entier dans la table ASCII associé au caractère.

Explications *chr* et *ord*.

(a) Que retourne $chr(68)$? $chr('A')$? $ord(68)$? $ord('A')$?

(b) Dans les fonctions codage et décodage, justifier les nombres 91, 26 et 64.

2. Pour un usage régulier, il serait pratique d'avoir une fonction algorithmique qui à partir d'un cryptogramme renvoie le tableau des fréquences. Programmer cette fonction.