

Mps : Police et enquête

Boris Laval

23 janvier 2013

Un document largement inspiré du travail en groupe avec Olivier Sicard et Teddy Wong (Lycée Bellepierre)

Table des matières

1	Séance I	2
1.1	Le scénario	2
1.2	Indice	2
1.3	Le code de César	2
1.4	Analyse fréquentielle	3
2	Séance II	5
2.1	Initiation à Python	5
2.1.1	Typage : des nombres et des chaînes	5
2.1.2	Boucle pour	6
2.2	Le code pin	6
2.2.1	Si alors sinon	6
2.2.2	Boucle tant que	6
2.2.3	Les différents algorithmes	6
2.2.4	Un programme final	8
3	Séance III	9
3.1	Initiation à Python	9
3.1.1	Longueur d'une chaîne	9
3.1.2	La chaîne : une suite de lettre	9
3.1.3	La chaîne : une suite de nombre	9
3.1.4	La chaîne : une suite de nombre	9
3.1.5	Les codes ascii	9
3.2	César	11
3.2.1	Algorithme	11
3.2.2	Programme	11
3.3	Analyse fréquentielle	11
3.3.1	Algorithme	11
3.3.2	Programme	12

1 Séance I

1.1 Le scénario

Nous sommes au milieu du printemps, dans un lotissement paisible. Un matin, un homme trapu se rend chez une cliente qu'il connaît bien. Cette case est habitée par Mme Untel.

Il sonne plusieurs fois. La maison semble vide. Il décide d'en faire le tour. Il découvre alors la fenêtre brisée. Mr Lapose sort alors son mètre et mesure les dimensions de la fenêtre pour faire sa commande. Il appelle une dernière fois. Personne ne répond. Curieux, il jette un oeil à l'intérieur et stupeur, il découvre le corps de Mme Untel.

En arrivant sur place, les policiers estiment la situation suffisamment grave pour avertir le commissariat qui contacte le procureur. Celui-ci commande l'intervention des techniciens de la police scientifique.

En arrivant sur place, les techniciens photographient les lieux : Une maison plutôt propre, un canapé, une télévision 3D, un sac de sport contenant une raquette de tennis, un ordinateur Apple, un téléphone portable, une baie vitrée est explosée, un verre avec une poudre blanche posé sur le bar, des traces de débris provenant de la fenêtre (portant des empreintes)?..

Puis, ils repèrent et prélèvent différents indices : Un verre avec une poudre blanche avec différentes empreintes. Des taches de sang. Des traces de terre sur le sol. Une pierre arrondie sur la moquette. Un mégot dans le cendrier.

Le téléphone, l'ordinateur et tous ces indices sont rapportés au laboratoire afin d'être étudiés.

Par ailleurs, des interrogations sont menées auprès du voisinage de la victime. Les voisins confirment que Mme Untel est une femme sans histoire. Mr Lapose confie qu'il connaissait très bien la victime.

Dans l'ordinateur du bureau, dans la corbeille du gestionnaire de messagerie et au milieu de quelques courriels d'échange de nouvelles avec la famille, est retrouvé un courriel qui semble codé. Bizarre ...!!

1.2 Indice

DE WO WKXAEOC. T'KS RKDO NO WO LUYDDSB MYXDBO DYS. DYEDOC MOC
ROEBOC VYSX NO DYS WO DEOXD. BOXNOJ-FYEC MO CYSB à V'OXNBYSR RKLS-
DEOV. X'RéCSDO ZKC à AESDDOB EX MVSOXD, CYSC à V'ROEBO, TO D'OX CEZVSO,
DE WO WKXAEOC WYX KUYEB... T'KS ZOEB NO DO ZOBNO. ZKBNYX, TO CKSC
AEO TO CESC DBYZ TKVYECO.

1.3 Le code de César

Nous sommes en 46 av JC. Vous venez de capturer un esclave Romain appartenant à l'armée de César. Sous ses cheveux, est inscrit sur son crâne un message qui semble être crypté. Après une torture en règle l'esclave avoue que ce message était destiné à Cléopâtre, reine d'Egypte et amour de César. Malheureusement aucune torture ne délia la langue du prisonnier concernant la façon de déchiffrer le message.

Voici ce qui est écrit :

ZU CU IKYI CYI Q DK
 QK REHT TU BQ WHQDTU RBUKU
 Z'QY CYI CED SEUKH Q DK
 UJ YB QLQYJ TUI RBUKI

Saurez-vous le décrypter ?

Rem : Ce n'est pas le texte de César original mais sa traduction française.

Exercice 1

Décrypter le texte suivant sachant que la clé est 10 : ebqoxd ovswsxob vk mslvo.

Exercice 2

Sachant que le texte en clair suivant : rendez vous rue de la paix donne :
 mziyzu qjpn mpz yz gv kvds.
 Déterminer la clé de cryptage.

Exercice 3

Décrypter le texte suivant en supposant que le mot "ennemi" y figure :
 stywj jssjrn ij ytzotzwx jxy ij wjytzw

Exercice 4

Décrypter le texte suivant :
 tw j l fyp elfap olyd yzecz dpcgtnp tw qlfe wl opxldbfc

Exercice 5

Décrypter le texte suivant dans lequel les espaces entre les mots ont été supprimés :
 rggfikhvqcrwfidlevjvtivkvtfddvgivmlvkevgivmvevqgrjergfcztv

1.4 Analyse fréquentielle

Exercice 1

Choisir 5 lignes de texte dans votre livre de Français et en comptant le nombre d'occurrences de chacune des lettres, remplir le tableau suivant :

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Cette répartition des fréquences des lettres n'est qu'approximative, cela dépend de nombreux paramètres tels que le niveau de langue du texte, ainsi que du style d'écriture (Par exemple un message militaire utilisera souvent de nombreuses abréviations).

Exercice 2

Utilisez l'analyse fréquentielle pour décoder le message de César.

Exercice 3

Un poème à décoder grâce à l'analyse fréquentielle

JUAR LSMX J'MWU EUMKSSU XUTVU

LRHXUDU SUX PRQYWX ARMWKX

XYKH UZ DRZKW, SR IHKXU UXZ EHRKTVU, VUSRX!

SUX IURMN AYMHX XYWZ EKWKX!

ZVUYLVKSU PRMZKUH

2 Séance II

On commence par une initiation à Python que l'on prolonge par l'écriture d'un algorithme connu de tous, la saisie d'un code pin à l'ouverture d'un portable.

L'idée est de tester les algorithmes proposés par la classe. On les traduit en python. L'exécution permet d'observer les défauts, ce qui justifie la réécriture de l'algorithme.

2.1 Initiation à Python

2.1.1 Typage : des nombres et des chaînes

```
#initialisation
a=0
b=0
#saisie des 2 nombres
a=input('entre le premier nombre? ')
b=input('entre le deuxième nombre? ')
#affiche la somme
print a+b
```

Prolongement : Améliorer l'affichage afin d'afficher le résultat sous la forme $3+5=8$

```
#initialisation
a=0
b=0
#saisie des 2 nombres
a=input('entre le premier nombre? ')
b=input('entre le deuxième nombre? ')
#affiche la somme
print a,'+',b,'=',a+b
```

On refait la même addition avec des chaînes de caractères

```
#saisie de 2 textes
c=raw_input('entre le premier texte? ')
d=raw_input('entre le deuxième texte? ')
#affiche la somme
print c+d
```

On remarque alors que l'addition $54+34=5434$, ce qui peut poser problème.

2.1.2 Boucle pour

```
#affiche les nombres de 0 à 19
for i in range(20):
    print i
```

```
#affiche les nombres à partir de 5, plus petit que 20, de 3 en 3
for i in range(5,20,3):
    print i
```

2.2 Le code pin

2.2.1 Si alors sinon

```
code=input('entrer votre code?')
if code==1234:
    print 'code bon'
else:
    print 'code mauvais'
```

2.2.2 Boucle tant que

```
#On demande le code
code=input('entrer votre code?')

#tant que le code n'est pas valable, on affiche code mauvais et on redemande un code
while code!=1234:
    print 'code mauvais'
    code=input('entrer votre code?')

#on affiche le code est bon
print 'code bon'
```

2.2.3 Les différents algorithmes

Les algorithmes précédents sont incomplets, car le smartphone n'autorise que 3 essais. Les élèves vont élaborer leurs propres algorithmes, on va alors les traduire fidèlement en python et les exécuter. Le va-et-vient entre les algorithmes et leurs exécutions favorise la compréhension.

```
demande du code
si le code est mauvais
alors on redemande le code
sinon on affiche bienvenue
```

```
demande du code
on initialise le nombre d'essais à 1

Tant que le code n'est pas bon
  On affiche code mauvais
  On affiche le nombre d'essais restant
  On demande un autre code

Si le code est bon on affiche Bienvenue
```

```
demande du code
on initialise le nombre d'essais à 1

Tant que le code n'est pas bon
  On affiche code mauvais
  On affiche le nombre d'essais restant
  On demande un autre code
  On actualise le nombre d'essais

Si le code est bon on affiche Bienvenue
```

```
demande du code
on initialise le nombre d'essais à 1

Tant que le code n'est pas bon et qu'il reste des essais
  On affiche code mauvais
  On affiche le nombre d'essais restant
  On demande un autre code
  On actualise le nombre d'essais

Si le code est bon on affiche Bienvenue sinon on bloque
```

2.2.4 Un programme final

```
#demande du code
code=input('entrer votre code?')
#on initialise le nombre d'essais à 1
essai=1

#tant que le code n'est pas valable et qu'il reste encore des essais
while code!=1234 and essai<3:
    # On affiche code mauvais
    print 'code mauvais'
    # On affiche le nombre d'essais restant
    print 'il reste ',3-essai
    # On redemande un code
    code=input('entrer votre code?')
    # On actualise le nombre d'essais
    essai=essai+1

#Si le code est bon on affiche Bienvenue sinon on bloque
if code==1234:
    print 'code bon'
else:
    print 'bloquage'
```

3 Séance III

La séance commence par une initiation à python pour donner des outils aux élèves pour traiter l'énigme. L'objectif est de produire un programme permettant de décoder. Les élèves vont expliciter la méthode des décalages exécutée à la main lors de la première séance en algorithmes. Comme prolongement, on peut traiter l'analyse fréquentielle.

3.1 Initiation à Python

3.1.1 Longueur d'une chaîne

```
mot=raw_input('Entre le mot?')
print len(mot)
```

3.1.2 La chaîne : une suite de lettre

```
mot=raw_input('Entre le mot?')
d=input('Entre le rang?')
print mot[d]
```

3.1.3 La chaîne : une suite de nombre

```
mot=raw_input('Entre le mot?')
d=input('Entre le rang?')
print mot[d],ord(mot[d])
```

3.1.4 La chaîne : une suite de nombre

```
mot=raw_input('Entre le mot?')
for l in mot:
    print l
    #print ord(l)
```

3.1.5 Les codes ascii

```
for i in range(128):
    print i,chr(i)
```

```
code=input('code ascii ?')  
print chr(code)
```

3.2 César

3.2.1 Algorithme

```
On récupère le texte à décoder
On demande le décalage
On parcourt le texte lettre par lettre et pour chaque lettre:
    On décale
    On ajoute la lettre obtenu au texte en clair
On affiche le texte en clair
```

3.2.2 Programme

```
#On récupère le texte à décoder
tmp=open('adecoder.txt','r')
texte=tmp.read()
#transforme en majuscule
texte=texte.upper()

#On demande le décalage#
d=input ('decalage')

#On parcourt le texte lettre par lettre et pour chaque lettre:
rendu=''
for l in texte:
#On décale
    if 64<ord(l)<91:#est une majuscule? 65=A et 90=Z
        if ord(l)+d>90:c=chr(ord(l)-26+d)#si le décallage dépasse Z, on reprend à partir de A
        else: c=chr(ord(l)+d) #sinon on decalle normalement
    else:c=l#les espaces ou caractères spéciaux ne sont pas transformés
    #On ajoute la lettre obtenu au texte en clair
    rendu=rendu+c #le texte transformé recoit le caractère transformé.
print texte # on affiche la source
print rendu #on affiche le rendu.
tmp.close()
```

3.3 Analyse fréquentielle

3.3.1 Algorithme

```
On récupère le texte à analyser
On initialise les compteurs par lettre à 0
On parcourt le texte lettre par lettre et pour chaque lettre, on actualise son compteur
On affiche les décomptes
```

3.3.2 Programme

```
#On récupère le texte à analyser
tmp=open('adecoder.txt','r')
texte=tmp.read()
tmp.close()
texte=texte.upper()

#On initialise les compteurs par lettre à 0
n=range(26)
for i in range(26):
    n[i]=0

#On parcourt le texte lettre par lettre et pour chaque lettre, on actualise son compteur
for l in texte:
    if 64<ord(l)<91:
        tmp=ord(l)-65
        n[tmp]=n[tmp]+1

#on affiche les décomptes
for i in range(26):
    print chr(65+i),n[i]
```