UNIVERSITE CHEIKH ANTA DIOP DE DAKAR

ECOLE SUPERIEURE POLYTECHNIQUE

DEPARTEMENT INFORMATIQUE

DIC 1 && Licence GL Langage C

TD & TP N° 2 [1] [2]

Préambule

La cryptologie est une science très ancienne : les hommes ont toujours eu besoin de dissimuler des informations et de transmettre des messages en toute confidentialité.
Elle regroupe la cryptographie et la cryptanalyse : la première a pour but de concevoir des systèmes visant à assurer la sécurité des communications sur un canal public et la seconde vise à trouver des failles dans ces systèmes.
La cryptographie est traditionnellement utilisée pour dissimuler des messages aux yeux de certains utilisateurs et le chiffrement des communications militaires a depuis l'Antiquité été une préoccupation majeure des diverses forces armées. Le *chiffrement* regroupe les techniques mises en oeuvre pour brouiller la signification d'un message qui est matériellement visible. Le contenu du message ne doit alors être retrouvé que par les personnes auxquelles le message est adressé. Le chiffrement fait appel à deux processus élémentaires impliquant la transformation des lettres d'un message pour satisfaire ces propriétés : la *substitution* qui consiste à remplacer, sans en bouleverser l'ordre, les symboles d'un texte clair par d'autres symboles et la *transposition* qui repose sur le bouleversement de l'ordre des symboles (mais pas leur identité).

CHIFFREMENT PAR SUBSTITUTION MONO-ALPHABÉTIQUE

Le *chiffrement par substitution* consiste à remplacer dans un message un ou plusieurs symboles par un ou plusieurs symboles (généralement du même alphabet) tout en conservant l'ordre de succession des symboles du message. Le chiffrement par substitution mono-alphabétique le plus simple est le *chiffrement par décalage*, aussi connu sous le nom de *chiffrement de César*. Il consiste simplement à décaler les lettres de l'alphabet d'un nombre de positions constant vers la droite ou la gauche.

[1] Si vous vous attendez à une correction de notre part vous risquez d'être déçus. ☹

[2] Bon courage ☺

Par exemple, en décalant les lettres de trois rangs vers la gauche (comme le faisait J. César),
Le texte clair *veni vidi vici* devient *yhql ylgl ylfl*.

Exercice 1

Ecrire un programme permettant de saisir un texte terminé en tapant le mot « FIN ». Ensuite de parcourir le texte et de recenser le nombre d'espaces, de caractères alphabétiques, le nombre de chiffres, le nombre de ponctuations, le nombre de retour à la ligne, etc.
Ecrire un programme qui prend en entrée un texte puis efface tous les espaces superflus. Exemple :
' Ceci est une banane . ' Sera remplacé par :
'Ceci est une banane.'
Ecrire un programme qui prend en entrée un texte puis détermine si le texte est une tautologie ou pas.
NB : Une tautologie est un texte dans lequel tous les mots commencent
par une même lettre. Exemple de tautologie : ' Le lion lape le lait.'

Exercice 2 Cryptographie 1 : Chiffre de César I

Dans cet exercice, le but est de mettre en place un algorithme de codage simple par décalage des lettres de l'alphabet. Le principe est que si la clé est 3 par exemple alors on va faire un décalage de trois caractères, ainsi le A devient le D, le B devient le E, ..., le Z devient le ? Etc. C'est circulaire.
D'abord on considérera que la clé est égale à 3.
1 Déclarez et initialisez un tableau texteClair avec le texte à crypter 2 Ecrire une fonction crypter_caractere qui permet de crypter un caractère et renvoie le code correspondant
3 Tester dans une fonction main, la fonction précédente qui sera appliquée sur le texteClair en entier et le crypte.
4 Ecrire une fonction decrypter_code qui permet de décrypter le code correspondant à un caractère puis renvoie le caractère
5 Tester dans une fonction main, la fonction précédente qui sera appliquée sur le texte crypté en entier et affiche le message en clair.

Exercice 3 Cryptographie 2 : Chiffre de César II

Cet exercice est une généralisation du précédent. En effet, ici on ne connaît pas par avance la clé et ce sera à l'utilisateur de donner la clé. Adaptez l'exercice précédent à la situation.
NB : Votre sens proverbial de l'élégance vous interdira bien sûr une série de vingt-six.

Exercice 4 Cryptographie 3 : Alphabet Aléatoire

Une technique ultérieure de cryptographie consista à opérer non avec un décalage systématique, mais par une substitution aléatoire. Pour cela, on utilise un alphabet-clé, dans lequel les lettres se succèdent de manière désordonnée, par exemple :
HYLUJPVREAKBNDOFSQZCWMGITX. C'est cette clé qui va servir ensuite à coder le message. Selon notre exemple, les A deviendront des H, les B des Y, les C des L, etc. Ecrire un algorithme qui effectue ce cryptage (l'alphabet-clé sera saisi par l'utilisateur, et on suppose qu'il effectue une saisie correcte).

Exercice 5 Cryptographie 4 : Le chiffre de Vigenère

Un système de cryptographie beaucoup plus difficile à briser que les précédents fut inventé au XVIe siècle par le français Vigenère. Il consistait en une combinaison de différents chiffres de César. On peut en effet écrire 25 alphabets décalés par rapport à l'alphabet normal :

- l'alphabet qui commence par C et finit par ...ZAB
- etc.

Le codage va s'effectuer sur le principe du chiffre de César : on remplace la lettre d'origine par la lettre occupant la même place dans l'alphabet décalé. Mais à la différence du chiffre de César, un même message va utiliser non un, mais plusieurs alphabets décalés. Pour savoir quels alphabets doivent être utilisés, et dans quel ordre, on utilise une clé. Si cette clé est "VIGENERE" et le message "Il faut coder cette phrase", on procèdera comme suit : La première lettre du message, I, est la 9e lettre de l'alphabet normal. Elle doit être codée en utilisant l'alphabet commençant par la première lettre de la clé, V. Dans cet alphabet, la 9e lettre est le D. I devient donc D. La deuxième lettre du message, L, est la 11e lettre de l'alphabet normal. Elle doit être codée en utilisant l'alphabet commençant par la deuxième lettre de la clé, I. Dans cet alphabet, la 11e lettre est le S. L devient donc S, etc. Quand on arrive à la dernière lettre de la clé, on recommence à la première. Ecrire l'algorithme qui effectue un cryptage de Vigenère, en demandant bien sûr au départ la clé à l'utilisateur.

Exercice 6 Cryptanalyse du chiffre de Vigenère

Fonction 1 Lecture du message secret à décoder :

Secret :=
KQOWEFVJPUJUUNUKGLMEKJINMWUXFQMKJBGWRLF
NFGHUDWUUMBSVLPSN\
CMUEKQCTESWREEKOYSSIWCTUAXYOTAPXPLWPNTC
GOJBGFQHTDWXIZAYGFFNSXC\
SEYNCTSSPNTUJNYTGGWZGRWUUNEJUUQEAPYMEKQ
HUIDUXFPGUYTSMTFFSHNUOC\
ZGMRUWEYTRGKMEEDCTVRECFBDJQCUSWVBPNLGO
YLSKMTEFVJJTWWMFMWPNMEMT\

MHRSPXFSSKFFSTNUOCZGMDOEOYEEKCPJRGPMURSK
HFRSEIUEVGOYCWXIZAYGOS\
AANYDOEOYJLWUNHAMEBFELXYVLWNOJNSIOFRWU
CCESWKVIDGMUCGOCRUWGNMAA\
FFVNSIUDEKQHCEUCPFCMPVSUDGAVEMNYMAMVLF
MAOYFNTQCUAFVFJNXKLNEIWC\
WODCCULWRIFTWGMUSWOVMATNYBUHTCOCWFYT
NMGYTQMKBBNLGFBTWOJFTWGNTE\
JKNEEDCLDHWTVBUVGFBIJGYYIDGMVRDGMPLSWGJL
AGOEEKJOFEKNYNOLRIVRWV\
UHEIWUURWGMUTJCDBNKGMBIDGMEEYGUOTDGGQE
UJYOTVGGBRUJYS

Fonction 2 : kasiski(texte,l) Recherche dans le texte la distance entre deux occurences de la même chaine de longueur *l*. Il faut chercher le pgcd de toutes ces distances, mais attention, il peut y avoir de fausses valeurs de distances (l'hypothèse étant que deux occurences de la même chaîne dans le texte secret correspondent à deux occurences identiques dans le texte en clair, ce qui n'est pas toujours vrai, mais très fréquent). Une méthode qui n'est pas la plus efficace, mais qui est la plus simple, consiste à extraire toutes les sous-chaînes de longueur *l* du texte et pour chacune de chercher si elle apparait ailleurs dans le texte. On mesure alors la distance entre les chaines. La longueur de la clé est un diviseur de la majorité des distances. Pour la longueur de texte que j'ai utilisé, je vous conseille *l* = 4. **Pour le TD, la longueur des clés est comprise entre 5 et 9**. Un moyen simple consiste à faire une boucle sur les longueurs possibles de la clé, en comptant chaque fois combien il y a de distances qui sont multiples de cette longueur. En prenant le maximum de ces nombres, on en déduit la longueur de la clé.

```
> kasiski(secret,4;
5
```

Fonction 3 : rechercheCle(texte,tailleCle) Recherche la clé de codage connaissant sa longueur. Attention, c'est la procédure la plus difficile à écrire du devoir.

- Comme on connait la longueur de la clé *l*, on va réécrire la chaîne secrète sous la forme de *l* chaînes (la première chaîne correspond aux caractères 1,*l*+1,2*l*+1,... du message, la deuxième chaîne correspond aux caractères 2,*l*+ 2,2*l*+ 2,... du message, et ainsi de suite) : $s_1,\ldots,s_l$. Chacune de ces chaînes correspondra à un alphabet qui est obtenu avec à un décalage $d_1,\ldots,d_l$ par rapport à l'alphabet normal.
- On va calculer les décalages entre les alphabets : d_2-d_1, d_3-d_1, ... d_l-d_1. pour calculer le décalage entre 2 alphabets 1 et *k* on procède de la manière suivante : On calcule $x_i = Ic(s_1.cesar(s_k,i)), i =$

```
1
```

1..26. Le décalage sera l'indice correspondant au maximum des x_i (cela veut dire que les deux chaînes sont codées avec le même alphabet). Je vous rappelle que *Ic* correspond à l'indice de coincidence et *cesar*(*s*,*i*) correspond à un codage avec un décalage de *i*. (*Attention dans cette phase on met*

bout à bout les deux chaines de caractères, même si cela n'a aucun sens!! Je vous rappelle que l'on ne s'intéresse qu'a la fréquence des lettres. Dans la formule précédente le "." correspond à la concaténation des chaînes)

- Maintenant il faut fixer le décalage du premier alphabet. On met alors toutes les chaînes ensemble et on cherche la lettre la plus fréquente : cela va correspondre au E. Cela nous permet de trouver le premier alphabet et d'en déduire enfin la clé de codage

> rechercheCle(secret,5);

Fonction 4 : encodeVigenere(texte,cle) Encode un texte avec la clé selon le système de Vigenère. Une procédure facile, pour vous reposer de la précédente.

Fonction 5 : decodeVigenere(texte,cle) Decode un texte codé avec le système de Vigenère en utilisant la clé.

Fonction 6 : decrypteVigenere(texte) decrypte un texte codé avec le système de Vigenère, sans connaitre la cle. Il suffit de mettre tout ensemble. La procédure finale!!

```
decrypteVigenere := proc(texte)
local n, cle; # On devine la longueur de la cle n := kasiski(texte,4); print('La longueur de la cle' = n); # On devine la cle cle := rechercheCle(texte,n); print('La cle est'= cle); # On decode le texte
decodeVigenere(texte,cle);
end:
```

Un bon moyen de tester votre programme consiste à écrire un texte en clair et décryper ce qui est codé :
on doit retrouver le même texte.

> decrypteVigenere(encodeVigenere(clair,'MICHEL')); La longueur de la cle = 6
La cle est = MICHEL
..... le texte en clair
Bon courage.