1 Cryptographie élémentaire symétrique.

La méthode de chiffrement par substitution monoalphabétique est un protocole de cryptographie symétrique, consistant à remplacer chaque lettre de l'aphabet par une autre lettre. Cette méthode date de l'antiquité. Jules César utilisait le chiffrement par décalage pour certaines de ses correspondances, notamment militaires. L'utilisation de ces méthodes dura plus d'un millénaire, jusqu'à son déchiffrement au IX siècle par le philosophe arabe Al-Kindi, selon un procédé d'analyse des fréquences des lettres.

Exercice 1 : Chiffre de César ou chiffrement par décalage Ce chiffrement consiste à décaler chaque lettre de alphabet en suivant une permutation circulaire des lettres.

1. Combien de clés distinctes existe-t-il pour le chiffrement de César ?
2. Déterminer les clés qui ont permis de crypter les messages suivants par chiffrement de César : `OTQQTNTWP`, `TYWUIJYED`
3. Utilisez une attaque statistique pour casser le cryptogramme suivant :

   `FILMK OYFYP CHYHN LYFYM YWLYN MILN`
4. Bonus : Écrire un programme affichant toutes les possibilités de messages. Indications Xcas/Python/C :
   (a) en Xcas, les commandes `asc(s)` et `char(l)` convertissent une chaine de caractère `s` en liste `l` de codes ASCII (entre 0 et 255) et réciproquement, `l .+ 1` permet d'additionner 1 à tous les éléments d'une liste `l`, `irem()` calcule le reste de la division par un entier
   (b) en Python on peut utiliser `def asc(s): return [ord(j) for j in s]` et `def char(l): return ''.join([chr(j) for j in l])`
   (c) En C, les chaines de caractères sont des tableaux de codes ASCII, il n'y a pas besoin de conversion

Le tableau suivant indique la fréquence en pourcentage des lettres en français ordinaire :

E	A	I	S	T	N	R	U	L	O	D	M	P
15,9	9,4	8,4	7,9	7,3	7,1	6,5	6,2	5,3	5,1	3,4	3,2	2,9

C	V	Q	G	B	F	J	H	Z	X	Y	K	W
2,6	2,1	1,1	1	1	0,9	0,9	0,8	0,3	0,3	0,2	0	0

Remarque Le principe de Kerckhoffs exprime que la sécurité d'un cryptosystème ne doit reposer que sur le secret de la clef, et pas sur le secret de l'algorithme. Ici l'algorithme n'est pas secret, donc ce principe est respecté. Par contre le bien trop petit nombre de clés le rend complètement vulnérable.

Exercice 2 : Chiffrement par substitution monoalphabétique (ou permutation)
Plutôt que de se restreindre aux décalages circulaires, on s'autorise dans ce mode de chiffrement toute permutation de l'alphabet.

1. Soit la clé

$$K = \begin{pmatrix} \texttt{A B C D E F G H I J K L M N O P Q R S T U V W X Y Z} \\ \texttt{A Z E R T Y U I O P Q S D F G H J K L M W X C V B N} \end{pmatrix}$$

   Crypter les messages clairs obtenus à l'exercice 1 question 2.
2. Combien de clés distinctes existe-t-il pour le chiffrement par substitution monoalphabétique ?
3. Casser le cryptogramme suivant à l'aide d'une analyse fréquentielle (les accents ont été supprimés).
   `SD GKOQDIPRSK RLP SDR CIFTQDR OGPRR O'SDR SDQPR OR PKIQPRCRDP BSQ HRKCRPPIDP O'RXRFSPRK ORL HKGAKICCRL RDKRAQLPKRL. F'RLP SD RDLRCNBR OR FQKFSQPL RBRFPKGDQJSRL HRKCRPPIDP OR CIDQHSBRK ORL OGDDRRL LGSL MGKCR NQDIQKR, GS NQPL. FRPPR CIFTQDR HRKCRP OR PKIQPRK ORL QDMGKCIPQGDL LRBGD ORL LRJSRDFRL O'QDLPKSFPQGDL HKRORMQDQRL, IHHRBRRL ISLLQ HKGAKICCRL.`
   On pourra utiliser que les digrammes les plus fréquents sont ES, LE, EN, DE, NT et RE et les lettres doublées les plus fréquentes LL, SS, EE, NN, MM et TT.

2 Arithmétique des entiers et polynômes.

Exercice 1 : Entiers en base 2, 16 et 256.

1. Soit n_1 l'entier s'écrivant `0x1234` en base 16. Donner n_1 en base 2 et en base 10.
2. Soit n_2 l'entier s'écrivant 6000 en base 10. Écrire n_2 en base 16 puis en base 2.
3. Montrer qu'on obtient l'écriture en base 256 d'un entier en regroupant par paire son écriture en base 16.
4. Soit n un entier dont l'écriture en base 2 ne comporte que des 1. Montrer qu'il existe un entier $m > 0$ tel que $n = 2^m - 1$. Quel est l'analogue en base 16 ?

Exercice 2 : Opérations en base 2.

1. Rappeler les tables d'addition et de multiplication en base 2. Comparer avec le et logique et le ou exclusif.
2. Calculer la somme, la différence et le produit des 2 entiers s'écrivant `0b1101` et `0b1011` en base 2 en utilisant l'algorithme "école primaire" en base 2. Vérifiez vos résultats en les comparant à ceux obtenus en base 10.
3. Effectuer la division euclidienne de `0b1101000` par `0b1011` en utilisant l'algorithme de la potence (comme à l'école primaire, mais en base 2)

Exercice 3 :. base 10 et 16 L'application `openssl` permet d'afficher des entiers utilisés pour le cryptage RSA, en base 16 par groupe de 2 chiffres séparés par le signe :. Ainsi l'entier $n = 42348$ en base 10 sécrit `a5:6c`.

1. Déterminer en base 10 l'entier n représenté par `04:23:48`
2. Déterminer la représentation de l'entier n dont l'écriture en base 10 est 123456789.
3. Pour afficher un résultat facile à lire, la représentation des grands entiers se fait sur plusieurs lignes, chaque ligne contient au plus 15 groupes de 2 chiffres. Le début de l'affichage d'un entier n de 2048 bits $2^{2047} \leq n < 2^{2048}$ est le suivant :

   ```
   f2:0e:d4:9d:44:04:c4:c8:6a:5b:c6:9a:d6:df:9c:
   f5:56:f2:0d:ad:6c:34:b4:48:f7:a7:a8:27:a0:c8:
   ...
   ```

   Combien de lignes faut-il pour représenter n ? La dernière ligne est-elle complète ? Sinon, combien de groupes de deux chiffres contient la dernière ligne ?

Exercice 4 : Multiplication de polynôme et d'entiers en base 10 (exemple)
Calculer le produit des deux polynômes $3x^2 + 9x + 8$ et $4x^2 + 7x + 6$. Même question pour 398 et 476 en posant le produit. Observez les similitudes et les différences.

Exercice 5 : Multiplication de polynômes et d'entiers en base 256 (cas général)
On représente un polynôme m par la liste $[m_0, ..., m_d]$ de ses coefficients ($m_d \neq 0$) de même pour n :

$$m = \sum_{j=0}^{d} m_j X^j, \quad n = \sum_{k=0}^{\delta} n_k X^k \quad \Rightarrow \quad mn = \sum_{j=0}^{d} \sum_{k=0}^{\delta} m_j n_k X^{j+k}$$

L'algorithme de multiplication va initialiser le résultat sous forme d'une liste de 0 de taille suffisante, puis faire une double boucle, à chaque itération de la boucle intérieure, on calcule $m_j n_k$ on additionne au coefficient d'indice $j + k$ du résultat,

1. Déterminer la taille de la liste nécessaire pour représenter le produit
2. Écrire un algorithme en langage naturel effectuant le produit (indices commençant à 0)
3. Montrer que le nombre d'opérations élémentaires est proportionnel à $d\delta$
4. Bonus : implémenter et tester que le temps de calcul est proportionnel à $d\delta$.
5. On représente un entier naturel m par son écriture en base 256 (i.e. la liste $[m_0, ..., m_d]$ de ses coefficients en base 256), avec $m_j \in [0, 255]$ et $m_d \neq 0$, de même pour n :

$$m = \sum_{j=0}^{d} m_j 256^j, \quad n = \sum_{k=0}^{\delta} n_k 256^k \quad \Rightarrow \quad mn = \sum_{j=0}^{d} \sum_{k=0}^{\delta} m_j n_k 256^{j+k}$$

Comment peut-on utiliser l'algorithme précédent pour calculer le produit mn ? Attention aux retenues !

Exercice 6 : multiplication rapide (expérience)
Déterminer avec l'instruction `time()` de Xcas le temps nécessaire pour faire le produit de 2 entiers de un million de chiffres en base 10 (par exemple `a:=10^(10^6); time(a*a);`, puis de 2 entiers de 10 millions de chiffres. Est-ce compatible avec l'algorithme de multiplication de l'exercice 3/4 ?
Vous pouvez faire la même expérience en Python avec `import timeit` et par exemple
`timeit.timeit("a*a",setup="a=10**(10**6)",number=1)`

Exercice 7 : calcul de l'inverse
Soit a un nombre réel ou complexe non-nul.

1. Calculer les racines de $x = x(2 - ax)$.
2. A partir d'une approximation x_0 de $1/a$ on définit récursivement une suite en posant $x_n = x_{n-1}(2 - ax_{n-1})$.
   Montrer que $1-ax(2-ax) = (1-ax)^2$ et en déduire que la suite $x_0, x_1, \ldots$ converge quadratiquement (l'erreur après n itérations est à peu près le carré de l'erreur précédente) vers $1/a$ pour $|1-ax_0| < 1$.
3. Donner un nombre d'itérations suffisant pour calculer l'inverse d'un réel appartenant à l'interval fermé $[2/3, 4/3]$ avec une précision d'au moins 1000 chiffres à partir de $x_0 = 1$.
4. Vous voulez créer une bibliothèque permettant de travailler avec des réels arbitrairement précis. Comment pourrait-on implémenter la division ?

Exercice 8 : produit vs factorisation
Expérience : tester le temps nécessaire pour effectuer le produit n de 2 nombres premiers p et q ayant environ 25 chiffres chacun puis le temps nécessaire pour factoriser n avec Xcas ou avec un autre logiciel qui factorise des entiers (PARI-GP, msieve...).

Si on fait le produit $n = pq$ de deux nombres premiers p et q distincts ayant k chiffres, on a vu que le temps de calcul est majoré par Ck^2 (et même moins si on utilise un algorithme rapide pour k grand). Réciproquement, on souhaite factoriser n en testant tous les entiers possibles sachant que n est le produit de deux premiers. Écrire un algorithme en langage naturel effectuant cette factorisation. Montrer que le nombre de divisions à effectuer est au pire de l'ordre de 10^k et le temps d'exécution de l'algorithme de l'ordre de $k^2 10^k$. Pour quelles valeurs de k peut-on factoriser n par cette méthode en un temps raisonnable ? Comparer avec les valeurs de k pour lesquelles on peut faire le produit pq en un temps raisonnable.

Exercice 9 : PGCD

1. Calculer le p.g.c.d. de 126 et 230, puis de 427 et 715.
2. Montrer que le plus grand diviseur commun à a, b et c est $\mathrm{pgcd}(\mathrm{pgcd}(a, b), c)$.
3. Calculer le PGCD de 180, 606 et 751.
4. Même question pour 342, 405 et 720. Que peut-on dire des entiers de la forme $342k + 405l + 720m$ avec $k, l, m \in \mathbb{Z}$?

Exercice 10
Pour quelles valeurs de $n \in \mathbb{Z}$ les entiers n et $n + 1$ sont-ils premiers entre eux ?
Pour quelles valeurs de $n \in \mathbb{Z}$ les entiers $n - 1$ et $n + 1$ sont-ils premiers entre eux ?

Exercice 11 : Fibonacci
On définit la suite de Fibonacci par $F_{n+2} = F_n + F_{n+1}, \quad F_0 = 1, F_1 = 1$

1. Calculer les 10 premiers termes de la suite F_n.
2. Montrer que F_n est strictement croissante à partir du rang 1. En déduire le quotient et le reste de la division de F_{n+2} par F_{n+1}
3. Déterminer le PGCD de F_n et F_{n+1}.
4. Déterminer N le nombre d'étapes dans l'algorithme d'Euclide (avec reste positif) pour calculer le PGCD de F_{n+1} et F_n.
5. (Bonus) Montrer que si $a \leq F_{n+1}$ et $b \leq F_n$ sont premiers entre eux, alors le nombre d'étapes pour calculer leur PGCD est au plus N. Indication : S'il y a $N + 1$ étapes, montrer par récurrence sur k que le reste à l'étape $N + 1 - k$ est plus grand ou égal à F_{k+1}.

Exercice 12 PGCD binaire
Implémenter l'algorithme du PGCD binaire de deux entiers.

Exercice 13 PGCD de polynômes
Effectuer la division euclidienne de x^3-1 par x^2-1, en déduire un PGCD de ces 2 polynômes. Même question pour x^3-1 et x^2+1.

Exercice 14 : Bézout
Déterminer u, v et d =pgcd(a,b) tels que $au+bv=d$ pour $a,b=45,76$ puis pour $a,b=126,230$.

Exercice 15 Équations diophantiennes linéaires

1. On considère l'équation d'inconnues $x, y \in \mathbb{Z}$:

$$7x-9y=1.$$

   Montrer que que cette équation possède au moins une solution. En déduire qu'elle possède une infinité de solutions. Trouver toutes les solutions.

2. On considère l'équation d'inconnues $x, y \in \mathbb{Z}$:

$$12x+21y=5.$$

   Montrer que que cette équation ne possède pas de solution.

3. Déterminer tous les entiers x, y tels que : $11x+17y=5$.
4. Même question pour $21x-49y=14$.

Exercice 16 Un restaurant propose des menus à 13 et 19 euros. Si la recette de la journée est de 1000 euros, peut-on connaître le nombre de clients ayant choisi le menu à 13 euros et le nombre de clients ayant pris celui à 19 euros ?

Exercice 17 : programmation du calcul des coefficients de Bézout
Implémenter l'algorithme donnant l'identité de Bézout, vérifier avec les valeurs calculées à l'exercice 12. (Bonus) Même question avec des polynômes (commandes Xcas pour le quotient et le reste de la division euclidienne `quo` et `rem`)

Exercice 18 On considère l'étape de l'algorithme de Bézout qui suit celle donnant l'identité de Bézout, elle correspond à un reste nul et s'écrit donc $au_{N+1}+bv_{N+1}=0$. Montrer que $|au_{N+1}|=|bv_{N+1}|$=ppcm(a,b).

Exercice 19 Montrer que les polynômes $A=x-1$ et $B=x+1$ sont premiers entre eux et donner une identité de Bézout polynomiale $AU+BV=2$. En déduire si n est impair l'identité de Bézout pour $n+1$ et $n-1$. Même question pour n pair (indication, partir de la relation ci-dessus et d'une relation de Bézout pour $n-1$ et 2).

Exercice 20 Résoudre les systèmes suivants en $x \in \mathbb{Z}$:

1. $\left\{ \begin{array}{rcll} x & \equiv & 3 & [8] \\ x & \equiv & 6 & [27] \end{array} \right.$
2. $\left\{ \begin{array}{rcll} x & \equiv & 1 & [2] \\ x & \equiv & 2 & [3] \\ x & \equiv & 3 & [5] \\ x & \equiv & 4 & [7] \end{array} \right.$

Exercice 21 Une bande de 17 pirates dispose d'un butin composé de pièces d'or d'égale valeur. Ils décident de se le partager également et de donner le reste au cuisinier (non pirate). Celui ci reçoit 3 pièces. Mais une rixe éclate et 6 pirates sont tués. Le butin est reconstitué et partagé entre les survivants comme précédemment ; le cuisinier reçoit alors 4 pièces. Lorsqu'une épidémie survient, seul le bateau, le butin, 6 pirates et le cuisinier sont sauvés. Le butin est à nouveau partagé de la même manière et le cuisinier reçoit 5 pièces. Quelle est la fortune minimale que peut espérer le cuisinier lorsqu'il décide d'empoisonner le reste de l'équipage ?

Exercice 22 Éléments simples
Décomposer en éléments simples $\frac{7}{2\times3\times5}$. Même question pour $\frac{1}{(x-1)(x+1)(x^2+1)}$. En déduire $\int \frac{1}{x^4-1}$

Exercice 23 Écrire un algorithme de factorisation d'un entier naturel par divisions. L'implémenter et tester avec 1729, 1961, 2021, 2129.

Exercice 24 Crible d'Ératosthène
Écrire les entiers de 1 à 100, barrer 1 puis les multiples de 2 sauf 2, puis les multiples du premier entier non barré (qui est premier, ici c'est 3), continuer jusqu'à atteindre la racine carrée. Montrer que les entiers non barrés sont des nombres premiers. Bonus : implémenter cet algorithme pour obtenir la liste des entiers premiers inférieurs à un entier n fixé (indication : utiliser un tableau de booléens initialisés à vrai, l'action de barrer un nombre revient à donner à la case du tableau la valeur faux).

Exercice 25 Trouver les factorisations en nombres premiers de 1961 et 2027. Donner le nombre total de divisions euclidiennes effectuées.

1. En déduire pgcd(1961,2027).
2. Comparer le nombre de divisions effectuées par cette méthode et par l'algorithme d'Euclide pour calculer le pgcd de ces deux nombres.

Exercice 26 Soient $a, b \in \mathbb{Z}$. Montrer que si $a \wedge b = 1$, alors $a^2 \wedge b^2 = 1$.