Eléments à retenir sur l'arithmétique :

Système de numération :

- Numération en base 10, 8, 2 et 16
- Conversion entre les bases
- Représentation des nombres à virgule
- Opérations de base sur les différentes bases : +, -, *, /
- Notions d'arrondi

Divisibilité :

- Division Euclidienne
- Nombres premiers
- Décomposition en nombre premier
- PGCD : Plus grand commun diviseur
    - Par décomposition
    - Algorithme d'Euclide

Congruence :

- Qu'est-ce que la congruence ?
- Propriétés sur la congruence
    - Si $a \equiv b[n]$ alors $a-b = k.n$ avec k entier
    - Si $a \equiv b[n]$, $c \equiv d[n]$ et n entier naturel alors
        - $a + c \equiv b + d\ [n]$
        - $a - c \equiv b - d\ [n]$
        - $pa \equiv pb[n]$ pour tout entier naturel p
        - $ac \equiv bd[n]$
        - $a^p \equiv b^p\ [n]$ pour tout entier naturel p

- Congruence et cryptographie.
    - Revenir dans l'intervalle défini.

Exercice type :

Kim et John, cette année en BTS SIO, aimeraient s'envoyer des messages sans que quelqu'un ne puisse les lire. Ils décident de mettre au point des codes. Ils vont essayer de voir comment ils fonctionnent.

Kim se souvient que le cours d'arithmétique parlait de la conversion entre nombre décimaux et binaires et aimerait bien utiliser ce procédé dans son codage. Sur le principe elle aimerait :

- Convertir chaque lettre en nombre décimal :

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

- Convertir ce nombre décimal en binaire
- Appliquer à ce nombre binaire une opération XOR (ou exclusif) avec une clé :

a	b	a XOR b
0	0	0
0	1	1
1	0	1
1	1	0

- Reconvertir la séquence binaire en décimal puis en caractère.
- Envoyé son message tout en donnant à John la clé pour décoder.

Passer d'une base à l'autre :

1. Les lettres de 'A' à 'Z' auront une représentation décimale entre 0 et 25. De combien de chiffres ai-je besoin au minimum dans mon nombre binaire si je veux convertir une lettre ?
2. Appelons k, le nombre que je viens de trouver à la question a. Kim ne veut pas se limiter au minimum car elle sait que plus la clé est petite, plus il est facile de décoder le message. Elle voudrait donc que 5 lettres consécutives soient codées de façon différente. Autrement dit, elle voudrait une clé de longueur 5.k. Quelle est la taille de la clé nécessaire à Kim ?
3. Nous allons voir comment convertir notre premier message 'YO' :
    a. Retrouver les représentations décimales des lettres 'Y' et 'O'
    b. Convertir ces nombres décimaux en binaire
    c. Coder les lettres en utilisant l'opérateur XOR avec la clé : 11011 00110
    d. Convertir les codes binaires obtenus en décimal
    e. Donner le mot codé
4. Que se passe t'il si nous refaisons le même traitement sur la chaîne que nous venons d'obtenir ?

Utiliser les notions des divisibilités

John se souvient d'avoir entendu parler de codage affine mais ne sait plus trop comment ça marche. Il suffit selon lui de multiplier la valeur décimale d'une lettre par un nombre entier quelconque et si le résultat dépasse 25, on prend le reste de la division entière par 26. Sur le principe il aimerait :

- Convertir chaque lettre en nombre décimal :

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

- Multiplier ce nombre décimal par k entier.
- Calculer le reste de la division entière du produit par 26
- Reconvertir la séquence décimale en caractère.
- Envoyé son message à Kim en lui indiquant la valeur de k.

Pour le décodage, John a momentanément oublié comment cela fonctionne, mais nous allons déjà travailler sur la partie encodage.

1. John a-t-il raison de penser que si je prends le reste de la division entière par 26 je vais forcément obtenir un nombre compris entre 0 et 25 ?
2. John utilise donc un multiplicateur k = 4 pour son premier codage. Il veut coder le message 'RE' :
    a. Convertir les lettres en nombres décimaux
    b. Multiplier ces nombres par 4
    c. Prendre le reste de la division entière par 26
    d. Retrouver les lettres correspondantes à la valeur décimale
    e. Donner le message codé
3. Selon vous, est-ce que ce code est utilisable ?
4. John se souvient tout d'un coup que le nombre k doit respecter une propriété. Le PGCD de k et 26 doit être 1 pour que le codage fonctionne.
    a. Décomposer 26 en produit de facteurs premiers
    b. Donner la liste des valeurs de k possibles pour un codage affine avec k compris entre 0 et 20.

Utiliser les notions de congruences :

John décide finalement d'utiliser la valeur k = 9 pour coder son message. Il reste à John à retrouver le mécanisme de décodage. Il se souvient que l'on peut le retrouver avec les notions de congruence. Pour décoder il faut trouver un nombre k1 tel que l'on peut décoder en réalisant les opérations suivantes :

- Convertir chaque lettre en nombre décimal
- Multiplier ce nombre décimal par k1
- Calculer le reste de la division entière du produit par 26
- Reconvertir la séquence décimale en caractère.
- Lire le message décodé.

Bref, on réalise la même opération que pour coder en changeant juste k par k1. Mais comment déterminer ce k1 ? Ah si, John se souvient, k1 est déterminé à partir de k : $k \cdot k1 \equiv 1 \ [26]$. Dans notre exemple, k = 9.

1. Montrer que $729 \equiv 1 \ [26]$
2. Montrer que 729 peut s'écrire sous la forme : $729 = s^2$
3. Montrer que 1 peut s'écrire : $1 = t^2$
4. Exprimer l'expression vue en 1. En fonction de s et t.
5. Estimer la valeur de k1 et vérifier votre résultat.
6. Montrer à l'aide d'un tableur que le décodage fonctionne pour les valeurs de k et k1 données.