Quelques activités arithmétiques liées aux codes correcteurs et à la cryptographie

Robert Rolland(*)

1. Avertissement

Une partie des activités proposées dans la suite de cet article sont extraites de la brochure [5]. Nous renvoyons le lecteur à cette référence pour des compléments ou des aides concernant les thèmes abordés ici, ainsi que pour un cours d'arithmétique respectant les programmes actuels des classes terminales des lycées.

2. Enseigner l'arithmétique

Durant quelques années, l'arithmétique a complètement disparu des programmes de mathématiques du secondaire. Ce choix a été regrettable. En effet l'enseignement de cette discipline possède un grand intérêt pour diverses raisons que je voudrais énumérer ; celles-ci peuvent en outre aider à préciser l'esprit dans lequel on devrait le dispenser.

- **Histoire et culture.** L'arithmétique est un des secteurs scientifiques les plus anciens. Ses problèmes internes ont motivé durant des siècles des développements fondamentaux dans diverses parties des mathématiques.
- **Raisonnement, démonstrations.** Les méthodes de raisonnement et de démonstration utilisées dans ce domaine sont d'une grande richesse et d'une grande variété. Ceci est particulièrement intéressant à l'heure actuelle, où dans l'enseignement scientifique scolaire, la part de pensée abstraite est réduite à la portion congrue au bénéfice de larges pans descriptifs.
- **Applications récentes.** Dans une période récente de nombreux problèmes concrets liés à l'informatique, l'électronique ainsi qu'à la représentation, la compression, l'intégrité et la confidentialité des données, ont été résolus dans le cadre de l'arithmétique. L'arithmétique a trouvé là un champ d'applications et de problèmes ayant une implication immédiate dans la vie de tous les jours (sécurité des communications sur les réseaux informatiques et téléphoniques, commerce électronique, sécurité des transactions bancaires, etc.).
- **Algorithmique.** De nombreux problèmes donnent lieu à la mise en place d'algorithmes tant pour démontrer l'existence de solutions que pour en faire un calcul effectif.

(*) IREM de Marseille

- **Expérimentation, test des hypothèses.** En arithmétique de nombreuses situations se prêtent particulièrement bien à une phase exploratoire consistant en une expérimentation sur ordinateur dans le but de conjecturer un résultat, de tester des hypothèses.

Je voudrais enfin insister sur l'importance des mathématiques et du raisonnement abstrait dans la démarche scientifique. L'expérimentation a bien entendu son rôle, mais elle doit forcément être sous-tendue par une réflexion théorique. De ce point de vue il ne faut pas oublier que les théories mathématiques, éventuellement très abstraites, développées aujourd'hui sont les bases d'applications de demain ou d'après-demain. Le meilleur exemple en est justement l'arithmétique dont on a pu dire à une certaine époque qu'elle était une réflexion de salon et qui se retrouve aujourd'hui au centre de réalisations concrètes très riches.

Pour toutes ces raisons il convenait que l'arithmétique élémentaire ait une place dans la formation des élèves scientifiques. C'est ce que prévoit maintenant le programme de la spécialité mathématique des classes terminales.

3. Quelques rappels

Nous rappelons ici quelques résultats importants et quelques méthodes classiques dont nous aurons besoin. Cela concerne essentiellement le théorème de Bézout ainsi que l'utilisation des congruences.

3.1. Théorème de Bézout

3.1.1. Les théorèmes

Théorème 3.1. *Si* pgcd$(a,b) = d$, *il existe deux entiers* u *et* v *tels que* $ua + vb = d$.

Dans le cas où a et b sont premiers entre eux, la réciproque est vraie. On dispose alors du théorème suivant :

Théorème 3.2. *Deux nombres entiers* a *et* b *sont premiers entre eux si et seulement s'il existe des entiers* u *et* v *tels que* $au + bv = 1$.

Ce résultat joue un rôle important et permet d'établir diverses propositions, par exemple le lemme d'Euclide-Gauss :

Théorème 3.3. *(Lemme d'Euclide-Gauss) Si* c *divise* ab *et si* c *est premier avec* b, *alors* c *divise* a.

Preuve. Si c est premier avec b, alors on peut trouver u et v tels que $uc + bv = 1$. Par suite $auc + abv = a$. Mais auc est divisible par c et abv aussi, donc a est divisible par c.

3.2. Résolution complète de $ua + vb = d$ (où $d = \text{pgcd}(a,b)$)

Supposons tout d'abord a et b premiers entre eux. On sait qu'il existe un couple (u_0,v_0) tel que $u_0a + v_0b = 1$. En existe-t-il d'autres ? Si oui, trouver tous les couples (u,v) tels que $ua + vb = 1$.

Soit (u,v) un couple quelconque répondant à la question. Alors

$$a(u-u_0)+b(v-v_0)=0.$$

Donc b divise $a(u-u_0)$, et puisque b est premier avec a, b divise $u-u_0$. Par suite u est nécessairement de la forme $u=u_0+kb$. Si bien que

$$akb+b(v-v_0)=0$$

ou encore

$$v=v_0-ka.$$

Ainsi tout couple (u,v) répondant à la question vérifie

$$u=u_0+kb,$$

$$v=v_0-ka$$

pour un certain $k\in\mathbf{Z}$. D'autre part on vérifie immédiatement, que pour tout $k\in\mathbf{Z}$, le couple (u,v) défini précédemment convient.

Remarque. Existe-t-il une solution (u,v), où u et v sont petits en valeur absolue ? Supposons $a>0$, $b=1$. Alors $u_0=0$, $v_0=1$ convient. On a bien entendu un résultat analogue pour $a=1$, $b>0$.
Supposons maintenant $a>1$, $b>1$. Puisque pour tout couple (u,v) qui convient on a

$$u=u_0+kb$$

où (u_0,v_0) est une solution particulière et que réciproquement si u est de cette forme, il existe v tel que le couple (u,v) convienne, on peut supposer, quitte à faire la division euclidienne par b que cette solution particulière u_0 vérifie $0\leq u_0<b$. De plus il n'y a qu'une solution telle que u_0 soit dans cet intervalle. On ne peut pas avoir $u_0=0$ car alors v_0b vaudrait 1 ce qui est impossible puisque $b>1$. Donc

$$0<u_0<b.$$

Pour v_0 nous avons alors

$$v_0=\frac{1-au_0}{b},$$

d'où

$$\frac{1}{b}-a<v_0<0,$$

ce qui donne

$$-a<v_0<0.$$

En particulier si b (ou a) est petit, on peut trouver rapidement une solution par recherche exhaustive. Par exemple si $b=3$ alors $u_0=1$ ou 2 et on cherche lequel des deux nombres $1-a$, $1-2a$ est divisible par 3.

Supposons qu'on veuille maintenant résoudre

$$ua+vb=d$$

où $d=\text{pgcd}(a,b)$. Dans ces conditions on a $ua'+bv'=1$ avec $a'=\dfrac{a}{\text{pgcd}(a,b)}$ et

$b' = \dfrac{b}{\text{pgcd}(a,b)}$. Mais on sait qu'alors a' et b' sont premiers entre eux. On est donc ramené au problème précédent.

3.2. Congruences

La notion de congruence est une notion naturelle très courante. Par exemple on identifie usuellement 14h avec 2h de l'après midi, ce qui constitue un calcul modulo 12.

Définition 3.1. *Soit n un entier strictement positif. Deux entiers x et y étant donnés nous dirons que x est congru à y modulo n et nous noterons*

$$x \equiv y \quad (n),$$

si $x - y$ est un multiple de n, c'est-à-dire si on peut écrire

$$x = y + kn.$$

Remarquons que la définition nous permet de dire que :

- x est congru à lui-même modulo n,
- si x est congru à y modulo n, alors y est congru à x modulo n,
- si x est congru à y modulo n et y est congru à z modulo n, alors x est congru à z modulo n.

La congruence s'interprète aussi très bien avec la division euclidienne :

Théorème 3.4. *L'entier x est congru à l'entier y modulo n si et seulement si les restes des divisions euclidiennes de x par n et de y par n sont égaux.*

Ceci donne un intérêt particulier à ce reste commun. On notera

$$x \bmod n$$

le reste de la division de x par n. C'est l'unique nombre r tel que $0 \leq r < n$ qui soit congru à x modulo n.

Attention, certains langages de programmation ne donnent pas exactement ce résultat (en particulier si x est négatif). Ils donnent aussi un résultat pour n négatif (cas qu'on a exclu ici).

La congruence se comporte bien vis à vis des opérations habituelles.

Proposition 3.1. *Si*

$$x_1 \equiv y_1 \quad (n)$$

et

$$x_2 \equiv y_2 \quad (n),$$

alors

$$x_1 + x_2 \equiv y_1 + y_2 \quad (n)$$

et

$$x_1x_2 \equiv y_1y_2 \quad (n).$$

En particulier si

$$x \equiv y \quad (n),$$

alors

$$x^k \equiv y^k \quad (n).$$

3.3. Algorithmes

Les applications de l'arithmétique et le développement des moyens de calcul ont rendu cruciales les questions sur la possibilité effective de réaliser certaines opérations. Prenons par exemple un nombre n produit de deux grands nombres premiers p et q de 150 chiffres chacun. Peut-on calculer p et q connaissant n ? Bien entendu il existe un algorithme naïf : essayer de diviser n successivement par tous les nombres qui lui sont inférieurs. Mais compte tenu de la taille des nombres utilisés, cet algorithme ne peut se réaliser en un temps raisonnable. Jusqu'à présent, bien que des progrès soient faits sur les algorithmes utilisés et sur la puissance de calcul des machines, on ne sait pas résoudre pratiquement un tel problème. En revanche si a et b sont deux nombres très grands, on sait trouver en temps raisonnable $d = \text{pgcd}(a,b)$ ainsi que u et v tels que $au + bv = d$.

Ainsi, il y a des opérations réalisables pratiquement et d'autres qui dans l'état actuel de nos connaissances et de nos moyens techniques ne le sont pas.

Citons quelques exemples de calculs réalisables avec des nombres donnés très grands :

- Trouver $d = \text{pgcd}(a,b)$ ainsi que u et v tels que $au + bv = d$ (algorithme d'Euclide étendu par exemple).
- Calculer $a^k \bmod n$.
- Trouver de grands nombres premiers.

Voici maintenant quelques problèmes actuellement hors d'atteinte pour des nombres très grands :

- Factoriser un nombre n produit de deux grands nombres premiers.
- Étant donnés des nombres a, b, n tels qu'il existe k vérifiant $a^k \equiv b \ (n)$, trouver k (problème du logarithme discret).
- Étant donnés un nombre n, produit de deux grands nombres premiers inconnus, et b tel qu'il existe x vérifiant $x^2 \equiv b$ (n), trouver x (problème de la racine carrée).

3.3.1. Algorithme d'Euclide étendu

On sait que si $\text{pgcd}(a,b) = d$, il existe deux entiers u et v tels que $ua + vb = d$. Nous supposerons que $a > 0$ et $b \geq 0$. Le cas général s'en déduit.

Voici un algorithme (**algorithme d'Euclide étendu**, adaptation de l'algorithme d'Euclide) qui permet de trouver explicitement d, ainsi qu'un couple (u,v) qui convient.

Rappelons que l'algorithme d'Euclide calcule le reste r_2 de la division euclidienne de $r_0 = a$ par $r_1 = b$:

$$r_0 = q_1 r_1 + r_2,$$

puis de proche en proche r_k reste de la division euclidienne de r_{k-2} par r_{k-1} :

$$r_{k-2} = q_{k-1} r_{k-1} + r_k,$$

jusqu'à obtenir un reste r_{n+1} nul. Le dernier reste r_n non nul est pgcd(a,b). Dans l'algorithme d'Euclide étendu on ajoute à chaque pas du calcul la phase suivante :
en supposant que pour tout $j \leq k-1$ on ait pu écrire

$$r_j = u_j a + v_j b$$

(on constate que ceci est vrai pour r_0 avec $u_0 = 1$ et $v_0 = 0$ et pour r_1 avec $u_1 = 0$ et $v_1 = 1$), alors

$$r_{k-2} = q_{k-1} r_{k-1} + r_k,$$

ce qui nous donne

$$r_k = (u_{k-2} - q_{k-1} u_{k-1})a + (v_{k-2} - q_{k-1} v_{k-1})b.$$

On obtient donc

$$r_k = u_k a + v_k b,$$

avec

$$u_k = u_{k-2} - q_{k-1} u_{k-1}$$

et

$$v_k = v_{k-2} - q_{k-1} v_{k-1}.$$

Ainsi le couple (u_n, v_n) dont on a expliqué le calcul convient.

Donnons l'algorithme sous forme de programme :

```
R0 := a ;  (a ≥ 0)
R1 := b ;  (b ≥ 0)
U0 := 1 ;
U1 := 0 ;
V0 := 0 ;
V1 := 1 ;
tant que R1 > 0 faire
    début
        Q := Quotient_Division(R0,R1) ;
        R := Reste_Division(R0,R1) ;
        U := U0 – Q * U1 ;
        V := V0 – Q * V1 ;
        R0 := R1 ;
        R1 := R ;
        U0 := U1 ;
        U1 := U ;
        V0 := V1 ;
        V1 := V ;
    fin.
```

En sortie, R0 = pgcd(a,b), U0 = u et V0 = v.

Afin de préciser le coût de l'algorithme d'Euclide (cf. [4], page 25) introduisons la suite de Fibonacci qui est définie par $F_0 = 0$, $F_1 = 1$, $F_{n+1} = F_n + F_{n-1}$. En posant $\phi = \frac{1+\sqrt{5}}{2}$ (nombre d'or), on peut calculer

$$F_n = \frac{1}{\sqrt{5}}(\phi^n - (-\phi)^{-n}),$$

ce qui permet de dire que

$$F_n = \text{round}\left(\frac{1}{\sqrt{5}}\phi^n\right),$$

où round(x) est l'entier le plus proche de x. Comme $F_{n-1} < F_n$, la formule

$$F_{n+1} = F_n + F_{n-1}$$

exprime la division euclidienne de F_{n+1} par F_n. L'algorithme d'Euclide appliqué à ces deux nombres donne pour résultat $F_1 = 1$ et possède n pas. C'est le cas le plus défavorable, c'est-à-dire le cas où l'algorithme est le plus long relativement à la taille des données. Plus précisément :

Théorème 3.1. *Soit* N *un entier > 1 et n le plus grand entier tel que* $F_{n+1} < N$. *Si* $0 \leq a \leq b < N$, *alors le nombre de pas de l'algorithme d'Euclide appliqué au couple* (a,b) *est majoré par n. Ce nombre de pas est exactement n si et seulement si* $a = F_n$ *et* $b = F_{n+1}$.

On pourra trouver une preuve de ce résultat dans [4] (p. 25) ou [7] (p. 47).

3.3.2. Calcul d'une puissance. Utilisation de la base 2 pour accélérer le calcul de x^n

En calculant directement $x \cdot x \ldots \cdot x$, on fait $n - 1$ multiplications. Si on décompose n en base 2 on obtient

$$n = a_k 2^k + \ldots + a_1 2 + a_0.$$

Alors

$$x^n = x^{a_k 2^k + \ldots + a_1 2 + a_0} = x^{a_0} \times (x^2)^{a_1} \times ((x^2)^2)^{a_2} \times \ldots \times ((x^{2^{k-1}})^2)^{a_k}.$$

Il y a donc au maximum k carrés à effectuer de proche en proche et au maximum k autres produits, donc au plus $2k$ multiplications.

Par exemple si $n = 15$ alors on calcule x^2 puis $x^4 = (x^2)^2$ puis $x^8 = (x^4)^2$ et enfin $x^{15} = x \times x^2 \times x^4 \times x^8$, soit 6 multiplications. Remarquons qu'une décomposition astucieuse de 15 en 6 + 9 permet de ne faire que 5 multiplications.

Cette méthode s'applique bien entendu au calcul de x^k mod n, en prenant bien soin, pour ne pas introduire de nombres gigantesques, de prendre le modulo à chaque étape et non pas à la fin du calcul.

Détaillons sous forme de programme un calcul de a^k mod n, inspiré de ce qui vient d'être dit :

```
A := a ;
K := k ;
N := n ;
R := 1;
tant que K > 0 faire
    si K pair
        alors  début
                   A := A * A mod N ;
                   K := K/2 ;
               fin
        sinon  début
                   R := R * A mod N ;
                   K := K – 1 ;
               fin.
```

En sortie R contient a^k mod n.

4. Détection et correction des erreurs

On pourra trouver des solutions ou des indications pour les exercices de ce paragraphe dans [5].

4.1. Clés de contrôle

Dans la vie courante, on est amené à manipuler des numéros d'identification, par exemple les numéros I.N.S.E.E. (Institut National de la Statistique et d'Études Économiques), les numéros de comptes bancaires, les numéros I.S.B.N. (International Standard Book Number), etc. Ces numéros pouvant être assez longs, on les munit d'une clé qui permet de détecter (pas toujours) des erreurs de saisie éventuelles.

4.1.1. Numéro I.N.S.E.E

Le numéro I.N.S.E.E. d'un individu est constitué de 15 chiffres. En lisant de gauche à droite, le premier est 1 ou 2 suivant qu'il s'agit d'un homme ou d'une femme. Les deux chiffres suivants désignent les deux derniers chiffres de l'année de naissance, les deux suivants le mois de naissance, les deux suivants le département, les trois suivants la commune de naissance, les trois suivants le numéro d'inscription sur le registre d'état civil, les deux derniers forment une clé K calculée de la manière suivante : désignons par A le nombre entier constitué par les 13 chiffres de gauche ; soit r le reste de la division euclidienne de A par 97 ; on prend $K = 97 - r$.

a) Vérifiez pour votre numéro I.N.S.E.E.

b) Écrivons A sous la forme

$$A = H \times 10^6 + L$$

avec $0 \leq L < 10^6$.

Montrer que r est aussi le reste de la division euclidienne de $27 \times H + L$ par 97.

c) Soit A_1 le nombre constitué par un numéro I.N.S.E.E. (y compris la clé). Montrer que si un des chiffres de A_1 et un seul est erroné, l'erreur est détectée. Montrer que si deux chiffres consécutifs distincts sont permutés, l'erreur est détectée.

d) Donner un exemple d'erreur non détectée.

4.1.2. Clé de relevé d'identité bancaire (RIB)

Le relevé d'identité bancaire comporte de gauche à droite 5 chiffres pour le code de la banque, 5 chiffres pour le code du guichet, 11 chiffres pour le numéro de compte, 2 chiffres pour la clé. La clé K est calculée de la manière suivante : soit A le nombre constitué par les 21 chiffres de gauche ; on calcule le reste r de la division euclidienne de $100 \times A$ par 97. On prend $K = 97 - r$.

a) Calculer la clé pour le relevé 14607 00052 05215075057 xx.

b) Comment mener le calcul avec une calculette ?
(indication : écrire $100 \times A = H \times 10^{12} + M \times 10^6 + L$).

c) Soit A_1 le nombre constitué par un RIB (y compris la clé). Montrer que si un des chiffres de A_1 et un seul est erroné, l'erreur est détectée. Montrer que si deux chiffres consécutifs distincts sont permutés, l'erreur est détectée.

4.1.3. Numéro I.S.B.N.

L'*International Standard Book Number* utilise des mots de longueur 10 constitués avec les chiffres 0, 1, ..., 9 et le symbole X (qui représente le nombre 10) ; le symbole X ne sera utilisé, si nécessaire, que pour la clé.

Exemples : 2 84180 013 X, 2 84225 000 1, 0 471 62187 0, 0 12 163251 2.

Le premier chiffre représente le pays, un bloc de chiffres est attribué à un éditeur, un autre bloc est le numéro donné par l'éditeur, le dernier symbole est la clé, calculée de telle sorte que si $a_1a_2\ldots a_{10}$ désigne un numéro I.S.B.N.,

$$\sum_{i=1}^{10} i a_{11-i}$$

soit divisible par 11.

a) Vérifier les exemples donnés.

b) Montrer que si un chiffre (et un seul) est erroné, l'erreur est détectée.

c) Montrer que si deux chiffres distincts sont permutés, l'erreur est détectée.

d) Trouver toutes les valeurs de a et de b telles que 2 84225 0ab 1 soit un code I.S.B.N. valide.

e) Pourquoi prendre la somme des ia_{11-i} et pas seulement la somme des a_i ?

4.1.4. Le code UPC (universal product code)

Le code I.S.B.N. a le désavantage d'utiliser pour la clé un symbole « parasite » (le X). Ceci provient du fait que l'on travaille modulo 11. Peut-on faire une étude analogue en travaillant modulo 10 ? Voici le code UPC, utilisé avec les codes barres, qui est basé sur ce principe. Le code UPC utilise des nombres de 12 chiffres $a_1 \ldots a_{12}$ (11 chiffres pour désigner un produit, et une clé), de telle sorte que

$$\sum_{i=0}^{5} 3a_{2i+1} + \sum_{i=1}^{6} a_{2i}$$

soit divisible par 10.

a) Calculer la clé si le nombre formé par les 11 chiffres de gauche est 35602387190.

b) Montrer que si un chiffre (et un seul) est erroné, l'erreur est détectée.

c) Montrer que, sauf cas particulier à déterminer, la permutation de deux chiffres successifs distincts est détectée (hélas il y a des cas particuliers ; personne n'est parfait !).

4.1.5. Numéro de carte bancaire (règle de Luhn)

Un numéro de carte bancaire est de la forme

$$a_n a_{n-1} \ldots a_2 a_1 a_0,$$

où les a_i sont des chiffres décimaux qu'on identifiera aux nombres 0, 1, …, 9. Sur ces nombres on définit l'application

$$m(x) = \begin{cases} 2x & \text{si } 0 \leq 2x \leq 9, \\ x_1 + x_2 & \text{si } 2x = 10x_1 + x_2, \end{cases}$$

avec $0 \leq x_i \leq 9$.

a) Montrer que $m(x) = 2x \bmod 9$. En conclure que $0 \leq m(x) \leq 9$.

On impose à un numéro de carte bancaire de vérifier (règle de Luhn)

$$a_0 + m(a_1) + a_2 + m(a_3) + \ldots \equiv 0 \ \ (10).$$

Montrer que là aussi, si un chiffre (et un seul) est erroné, l'erreur est détectée et que, sauf cas particulier à déterminer, la permutation de deux chiffres successifs distincts est détectée.

4.1.6. Peut-on faire mieux ?

Ainsi le code UPC et la règle de Luhn ont l'avantage d'avoir une clé comprise entre 0 et 9, ils détectent une erreur, mais hélas ne détectent pas toujours une permutation de deux chiffres contigüs distincts. Peut-on construire un code détecteur, travaillant sur une suite de chiffres décimaux et ayant comme les deux codes précédents pour clé un chiffre décimal, qui détecte une erreur et qui détecte toujours une permutation de deux chiffres contigüs distincts ? La réponse est oui. On peut trouver un tel

exemple dans [3] (p. 590).

Montrer qu'avec un tel exemple on peut construire un tableau carré T de 10 cases sur 10 cases pour lequel chaque ligne et chaque colonne est une permutation de {0,…,9} (donc un carré latin) et pour lequel pour tout $i, j \in \{0,1\}$, on ait : si $i \neq j$ alors $T(i,j) \neq T(j,i)$.

4.2. Un code correcteur de Hamming

Ici on se propose non plus seulement de détecter, mais de corriger une erreur éventuelle. Considérons les nombres de 10 chiffres (numéros de téléphone par exemple) $a_1a_2\ldots a_{10}$ où les a_i peuvent prendre les valeurs 0, 1, …, 9. On rajoute une clé constituée de deux chiffres $a_{11}a_{12}$ où a_{11} et a_{12} peuvent prendre les valeurs 0, 1, …, 9 et aussi la valeur X, représentant le nombre 10. La clé est calculée de telle sorte que

1) a_{11} soit le reste de la division de $\sum_{i=1}^{10} a_i$ par 11,

2) a_{12} soit le reste de la division de $\sum_{i=1}^{10} i\,a_i$ par 11.

a) Calculer la clé pour le numéro 0491413940.

b) On part d'un numéro muni de sa clé $a_1a_2\ldots a_{12}$. On se propose de montrer que si en communiquant ce numéro on fait **une erreur sur un chiffre** (et pas plus), on peut reconstituer le bon numéro.

Montrer que si l'erreur est faite sur un a_i avec $1 \leq i \leq 10$, alors aucune des relations 1) et 2) n'est vérifiée.

Montrer que si l'erreur est faite sur a_{11}, la relation 1) n'est pas vérifiée, mais la relation 2) l'est.

Montrer que si l'erreur est faite sur a_{12}, la relation 1) est vérifiée mais pas la relation 2).

Montrer qu'on peut corriger l'erreur. Indiquer comment.

Exemple : Soit le numéro 049132900000. Vérifier que ce numéro n'est pas correct. En supposant qu'un seul chiffre soit faux, retrouver le bon numéro.

Le seul ennui du codage précédent est la présence éventuelle dans la clé du symbole « parasite » X. Mais on va voir qu'avec une clé de deux chiffres on ne peut pas se contenter des chiffres décimaux.

Impossibilité de réaliser une correction du type précédent uniquement avec les chiffres décimaux

Soit E l'ensemble des nombres de dix chiffres $a_1a_2\ldots a_{10}$ où les a_i sont des chiffres décimaux habituels. Soit F l'ensemble des nombres de 12 chiffres décimaux.

a) Quels sont les nombres d'éléments de E et de F ?