MAPLE

MP⋆ 2011-2012

Une année de MAPLE en MP⋆

Guillaume CONNAN

TD N°

1

Un peu d'arithmétique

1 Ave Cesar

Pour les problèmes de codage-décodage (cf X 2008), on a souvent besoin de manipuler des listes de caractères.
On peut utiliser la bibliothèque `StringTools` et en particulier :
- `Ord("car")` qui renvoie le code ASCII de `car`. Par exemple `Ord("a")` renvoie `97` ;
- `Char(code)` effectue l'opération inverse. Par exemple `Char(97)` renvoie `"a"` ;
- `Explode("abc")` renvoie `["a","b","c"]` ;
- `Implode(["a","b","c"])` renvoie `"abc"` ;

Créez une fonction `cesar:=proc(message,cle)` qui code un message rentré sous forme d'une chaîne selon la méthode bien connue de César, en se donnant la possibilité de choisir sa clé. On travaillera avec un alphabet de 95 lettres :

Caractère		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/	0	1	2
Code	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
Caractère	3	4	5	6	7	8	9	:	;	<	=	>	?	@	A	B	C	D	E
Code	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69
Caractère	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Code	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88
Caractère	Y	Z	[	\	]	^	_	‘	a	b	c	d	e	f	g	h	i	j	k
Code	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107
Caractère	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~
Code	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126

TABLE 1.1 – Table des caractères ASCII affichables

2 Éléments inversibles de Z_n

Dans la suite, nous noterons $\mathbb{Z}_n$ l'ensemble $\mathbb{Z}/n\mathbb{Z}$.
Dressez la table de Pythagore de $(\mathbb{Z}_n, \times)$ pour $n \in \{9, 11, 12\}$.
Vous créerez une procédure `pyth:=proc(n)` qui dresse la table de $(\mathbb{Z}_n, \times)$ en commençant par exemple par :

```
pyth:=proc(n)
 local T,i,j;
 T:=array(1..n-1,1..n-1);
 ........................
   ......................
      T[i,j]:=...........
   ...
 ...
 print(T)
 end:
```

On observe par exemple que $4 \times 2 \equiv 4 \times 5[12]$ mais on n'a pas pour autant $2 \equiv 5[12]$.
Autre problème : on a $4 \times 3 \equiv 0[12]$, alors que ni 4 ni 3 n'est nul.
Donnez une condition nécessaire et suffisante pour qu'un élément de $(\mathbb{Z}_n, \times)$ soit inversible.
Créez une procédure `invm(a,n)` qui calcule, s'il existe, l'inverse de `a` dans $(\mathbb{Z}_n, \times)$.
On pourra utiliser la commande `igcdex(a,b,'u','v')` qui renvoie $a \wedge b$ et qui stocke des coefficients de Bézout dans les variables `u` et `v`.

```
> igcdex(7,9,'u','v');
                                  1
> u,v;
                                4, -3
```

Comparez `invm(a,n)` avec `1/a mod n`...

3 Chiffrement affine

On utilise encore notre alphabet de 95 caractères et on va coder les nombres associés par une fonction du style :

$$f : x \mapsto ax + b\ [95]$$

où a et b sont des entiers.
En utilisant certaines procédures utilisées précédemment avec le code de César, créez une procédure `code_affine:=proc(message,a,b)` qui code une message à l'aide de la fonction précédente.
Comment décoder un message ? Créez une procédure `decode_affine:=proc(crypte,a,b)` qui décode un message crypté à l'aide de la fonction $f : x \mapsto ax + b\ [95]$.

4 Chiffrement de Vigenère

Blaise de VIGENÈRE naquit en 1523 à Saint-Pourçain sur Sioule, célèbre pour ses vins d'Auvergne. Il fit paraître en 1586 le *Traicté des chiffres ou secrètes manières d'escrire*[a].
Nous allons étudier un cas particulier. Choisissons une clé qui peut être un mot, une phrase, un ensemble de phrase, un livre entier. Pour nous simplifier la vie, prenons par exemple la clé « VENTRILOQUIST »[b].
On veut chiffrer le message « LES SANGLOTS LONGS DES VIOLONS DE L'AUTOMNE »
On a besoin d'un tableau de chiffrement où on affiche les 26 lettres de notre alphabet actuel sur chaque ligne, en décalant les lettres d'un cran d'une ligne à l'autre.
On peut le faire à la main...ou demander à Maple de le faire grâce à la commande **`Char`** et en travaillant modulo 26.
Retirons les espaces de notre message et disposons les lettres ainsi :

```
LESSANGLOTSLONGSDESVIOLONSDELAUTOMNE
VENTRILOQUISTVENTRILOQUISTVENTRILOQU
```

Pour chiffrer le L, il suffit de regarder dans notre tableau la lettre se trouvant sur la ligne du L et la colonne du V : c'est G.
On fait pareil pour le E qui est chiffré en I, le S en F, le S suivant en L, etc.

a. Le manuscrit est disponible sur Gallica
b. En plus d'un mot anglais désignant une personne parlant sans ouvrir la bouche, ce terme était également le nom d'un réseau de résistance dirigé par Philippe de VOMÉCOURT en Sologne et qui sabota les voies de chemin de fer menant vers la Normandie à partir du 5 juin 44 suite à la réception d'un fameux message codé...

Comme c'est assez répétitif, on va utiliser Maple pour faire le sale boulot. Il suffit de mathématiser un peu le travail et tout sera plus simple.
Vous devez obtenir par exemple :

```
> code_vige('LESSANGLOTSLONGSDESVIOLONSDELAUTOMNE','VENTRILOQUIST')
            'GIFLRVRZENADHIKFWVAGWEFWFLYIYTLBZADY'
> decode_vige('GIFLRVRZENADHIKFWVAGWEFWFLYIYTLBZADY','VENTRILOQUIST')
              'LESSANGLOTSLONGSDESVIOLONSDELAUTOMNE'
```

Plus la clé est longue, plus grande est la sécurité. D'ailleurs, le principe de la machine ENIGMA était semblable, avec une clé de longueur 26^n où n est le nombre de rotors utilisés.
Auparavant, les messages étaient cryptés à l'aide de clés représentés par des livres entiers.
Cependant, si la clé est trop courte, une étude statistique et arithmétique permet de casser le code comme le démontra Charles BABBAGE en 1846, près de trois siècles après la parution du traité de VIGENÈRE.

5 Fonction indicatrice d'Euler

5.1 Rappel

Il existe plusieurs définitions équivalentes de la fonction indicatrice d'EULER φ. Pour être en lien avec ce qui a été vu précédemment, nous dirons que $\varphi(n)$ est l'ordre du groupe des inversibles de $\mathbb{Z}_n$.
Ainsi, on obtient le théorème d'EULER : si a et n sont premiers entre eux, $a^{\varphi(n)} \equiv 1\ [n]$.

5.2 Théorème chinois

Notons $\dot{a}[n]$ la classe d'un entier a modulo n. Soit n et k deux entiers premiers entre eux.
Voici une version simplifiée mais suffisante pour la suite de nos aventures du fameux théorème :
L'application f définie par :

$$f : \begin{array}{ccc} \mathbb{Z}_{nk} & \to & \mathbb{Z}_n \times \mathbb{Z}_k \\ \dot{x}[nk] & \mapsto & (\dot{x}[n], \dot{x}[k]) \end{array}$$

est bien définie et c'est un isomorphisme d'anneaux.

5.3 Propriétés de φ

Pouvez-vous le démonter puis en déduire que pour tout couple (n,k) d'entiers naturels premiers entre eux :

$$\varphi(n \times k) = \varphi(n) \times \varphi(k)$$

Soit p un nombre premier et n un entier naturel. Prouvez que $\varphi\left(p^n\right) = p^n - p^{n-1}$ puis que, pour tout entier naturel non nul k, $\varphi(k) = k \prod_{p|k} \left(1 - \frac{1}{p}\right)$.

5.4 Calcul de φ(n)

Déterminez une procédure `eul:=proc(n)` qui calcule $\varphi(n)$.
Vous pourrez utiliser la fonction Maple `ifactors(n)` qui renvoie, pour des entiers positifs non nuls, une liste contenant 1 et la liste des diviseurs premiers de n et de leur valuation.
Par exemple :

```
> ifactors(60);
                    [1, [[2, 2], [3, 1], [5, 1]]]
```

5.5 Une première application

Calculez $\varphi(100)$.
Trouvez, DE TÊTE, les trois derniers chiffres de :

$$789587546378637837837834545364653897897^{2085827145914585514551351351351474540982582580001}$$

Vérifiez directement avec Maple...
Qu'en concluez-vous sur le calcul de $x^m[n]$ lorsque x et n sont premiers entre eux ?

5.6 Résolution de $x^m \equiv a$ [n]

Par exemple, on cherche un entier x tel que $x^{130355971} \equiv 3208718$ [9108163].

6 Oraux de Centrale

6.1 Exercice

Soit p un nombre premier. Pour tout entier $k \in \{1,...,p-1\}$, on pose :

$$p_k = \frac{(p-1)!}{k(p-k)}$$

1. Cette question est à traiter à l'aide d'un logiciel de calcul formel. Les instructions `ithprime` et `numer` de Maple pourront être utiles.
   (a) Étudier la divisibilité de l'entier $\sum_{k=1}^{p-1} p_k$ par p pour les 20 nombres premiers p.
   (b) On note $\frac{a_p}{b_p}$ la fraction irréductible représentant le rationnel $\sum_{k=1}^{p-1} \frac{1}{k}$. Étudiez la divisibilité de a_p par p^2 pour les 20 premiers nombres premiers.
2. On suppose que le nombre premier p est supérieur ou égal à 5 et on se place dans le corps $\mathbb{Z}_p$. On note $\overline{k}$ la classe d'un entier $k \in \mathbb{Z}$ dans $\mathbb{Z}_p$.
   (a) Montrer que $\overline{(p-1)!} = -1$.
   (b) Montrer que pour tout $k \in \{1,...,p-1\}$, on a : $\overline{p_k} = \left(\overline{k^2}\right)^{-1}$.
   (c) Montrer que $\sum_{k=1}^{p-1} \left(\overline{k^{-1}}\right)^2 = \sum_{k=1}^{p-1} \overline{k}^{-2}$. Conclure quant à la divisibilité de $\sum_{k=1}^{p-1} p_k$ par p.
3. Exprimer $pb_p \sum_{k=1}^{p-1} p_k$ en fonction de a_p. Qu'en conclut-on ?

6.2 Exercice sur la fonction de Möbius

La fonction $\mu : \mathbb{N}^\star \to \{-1,0,1\}$ de MÖBIUS est définie suivant $\mu(1) = 1$ et, pour tout $n \geqslant 2$, $\mu(n) = (-1)^r$ si n est un produit de r nombres premiers deux à deux distincts, $\mu(n) = 0$ si n est divisible par le carré d'un nombre premier.

1. Montrer que μ est une fonction *multiplicative*, c'est-à-dire : $\mu(1) = 1$ et, si a et b sont premiers entre eux, $\mu(ab) = \mu(a)\mu(b)$.
2. On note $d \mid n$ pour signifier que l'entier naturel $d \in \mathbb{N}^\star$ divise $n \in \mathbb{N}^\star$. Montrer que :
$$\forall n \geqslant 2, \quad \sum_{d|n} \mu(d) = 0$$
3. Calculer les $\mu(n)$ pour $n \in \{1,2,...,20\}$ à l'aide de Maple *sans utiliser* de commande Maple détectant les nombres premiers (de type `isprime` par exemple)

ÉCOLE POLYTECHNIQUE

ÉCOLE SUPÉRIEURE DE PHYSIQUE ET CHIMIE INDUSTRIELLES

CONCOURS D'ADMISSION 2008

FILIÈRE MP - OPTION PHYSIQUE ET SCIENCES DE L'INGÉNIEUR **FILIÈRE PC**

COMPOSITION D'INFORMATIQUE

(Durée : 2 heures)

L'utilisation des calculatrices **n'est pas autorisée** pour cette épreuve.

Le langage de programmation choisi par le candidat doit être spécifié en tête de la copie.

⋆ ⋆ ⋆

Ave Cesar (zud bdrzq)

On cherche à crypter un texte t de longueur n composé de caractères en minuscules (soit 26 lettres différentes) représentés par des entiers compris entre 0 et 25 ($0 \leftrightarrow$ `a`, $1 \leftrightarrow$ `b`, $\ldots 25 \leftrightarrow$ `z`). Nous ne tenons pas compte des éventuels espaces.

Ainsi, le texte `ecolepolytechnique` est représenté par le tableau suivant où la première ligne représente le texte, la seconde les entiers correspondants, et la troisième les indices dans le tableau t.

`e`	`c`	`o`	`l`	`e`	`p`	`o`	`l`	`y`	`t`	`e`	`c`	`h`	`n`	`i`	`q`	`u`	`e`
4	2	14	11	4	15	14	11	24	19	4	2	7	13	8	16	20	4
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

Codage de César

Ce codage est le plus rudimentaire que l'on puisse imaginer. Il a été utilisé par Jules César (et même auparavant) pour certaines de ses correspondances. Le principe est de décaler les lettres de l'alphabet vers la gauche de 1 ou plusieurs positions. Par exemple, en décalant les lettres de 1 position, le caractère `a` se transforme en `z`, le `b` en `a`, ... le `z` en `y`. Le texte `avecesar` devient donc `zudbdrzq`.

Question 1 Que donne le codage du texte `maitrecorbeau` en utilisant un décalage de 5 ?

Question 2 Écrire la fonction codageCesar(t, n, d) qui prend en arguments le tableau t, sa longueur n et un entier d ; et qui retourne un tableau de même taille que t contenant le texte t décalé de d positions.

Question 3 Écrire de même la fonction decodageCesar(t, n, d) prenant les mêmes arguments mais qui réalise le décalage dans l'autre sens.

Pour réaliser ce décodage, il faut connaître la valeur du décalage. Une manière de la déterminer automatiquement est d'essayer de deviner cette valeur. L'approche la plus couramment employée est de regarder la fréquence d'apparition de chaque lettre dans le texte crypté. En effet, la lettre la plus fréquente dans un texte suffisamment long en français est la lettre `e`.

Question 4 Écrire la fonction frequences(t', n) qui prend en argument un tableau t' de taille n représentant le texte crypté ; et qui retourne un tableau de taille 26 dont la case d'indice i contient le nombre d'apparitions du nombre i dans t ($0 \leq i < 26$).

Question 5 Écrire la fonction decodageAuto(t', n) qui prend en argument le tableau t' de taille n représentant le texte crypté ; et qui retourne le texte t d'origine (en calculant la clé pour que la lettre `e` soit la plus fréquente dans le texte décrypté).

Codage de Vigenère

Au XVIème siècle, Blaise de Vigenère a modernisé le codage de César très peu résistant de la manière suivante. Au lieu de décaler toutes les lettres du texte de la même manière, on utilise un texte clé qui donne une suite de décalages.

Prenons par exemple la clé `concours`. Pour crypter un texte, on code la première lettre en utilisant le décalage qui envoie le `a` sur le `c` (la première lettre de la clé). Pour la deuxième lettre, on prend le décalage qui envoie le `a` sur le `o` (la seconde lettre de la clé) et ainsi de suite. Pour la huitième lettre, on utilise le décalage `a` vers `s`, puis, pour la neuvième, on reprend la clé à partir de sa première lettre. Sur l'exemple `ecolepolytechnique` avec la clé `concours`, on obtient : (la première ligne donne le texte, la seconde le texte crypté et la troisième la lettre de la clé utilisée pour le décalage)

e	c	o	l	e	p	o	l	y	t	e	c	h	n	i	q	u	e
g	q	b	n	s	j	f	d	a	h	r	e	v	h	z	i	w	s
c	o	n	c	o	u	r	s	c	o	n	c	o	u	r	s	c	o

Question 6 Donner le codage du texte `becunfromage` en utilisant la clé de codage `jean`.

Question 7 Écrire la fonction codageVigenere(t, n, c, k) qui prend comme arguments un tableau t de taille n représentant le texte à crypter, et un tableau d'entiers c de longueur k donnant la clé servant au codage ; et qui retourne un tableau de taille n contenant le texte crypté t'.

Maintenant, on suppose disposer d'un texte t' assez long crypté par la méthode de Vigenère, et on veut retrouver le texte t d'origine. Pour cela, on doit trouver la clé c ayant servi au codage. On procède en deux temps : 1) détermination de la longueur k de la clé c, 2) détermination des lettres composant c.

La première étape est la plus difficile. On remarque que deux lettres identiques dans t espacées de $\ell \times k$ caractères (où ℓ est un entier et k la taille de la clé) sont codées par la même lettre dans t'. Mais cette condition n'est pas suffisante pour déterminer la longueur k de la clé c puisque des répétitions peuvent apparaître dans t' sans qu'elles existent dans t. Par exemple, les lettres `t` et `n` sont toutes deux codées par la lettre `h` dans le texte crypté à partir de `ecolepolytechnique` avec `concours` comme clé. Pour éviter ce problème, on recherche les répétitions non pas d'une

lettre mais de séquences de lettres dans t' puisque deux séquences de lettres répétées dans t, dont les premières lettres sont espacées par $\ell \times k$ caractères, sont aussi cryptées par deux mêmes séquences dans t'.

Dans la suite de l'énoncé, on ne considère que des séquences de taille 3 en supposant que toute répétition d'une séquence de 3 lettres dans t' provient *exclusivement* d'une séquence de 3 lettres répétée dans t. Ainsi, la distance séparant ces répétitions donne des multiples de k.

La valeur de k est obtenue en prenant le PGCD de tous ces multiples. Si le nombre de répétitions est suffisant, on a de bonnes chances d'obtenir la valeur de k. On suppose donc que cette assertion est vraie.

Question 8 Écrire la fonction pgcd(a, b) qui calcule le PGCD des deux entiers strictement positifs a et b par soustractions successives de ses arguments.

Question 9 Écrire la fonction pgcdDesDistancesEntreRepetitions(t', n, i) qui prend en argument le texte crypté t' de longueur n et un entier i $(0 \leq i < n - 2)$ qui est l'indice d'une lettre dans t' ; et qui retourne le pgcd de toutes les distances entre les répétitions de la séquence de 3 lettres $\langle t[i], t[i+1], t[i+2] \rangle$ dans la suite du texte $\langle t[i+3], t[i+4], \ldots t[n-1] \rangle$. Cette fonction retourne 0 s'il n'y a pas de répétition.

Question 10 Écrire la fonction longueurDeLaCle(t', n) qui prend en argument le texte crypté t' de longueur n ; et qui retourne la longueur k de la clé de codage.

Question 11 Donner le nombre d'opérations réalisées par la fonction `longueurDeLaCle` en fonction de la longueur n ? (On ne comptera que le nombre d'appels à la fonction PGCD).

Question 12 Une fois la longueur de la clé connue, donner une idée d'algorithme permettant de retrouver chacune des lettres de la clé. (Il s'agit de décrire assez précisément l'algorithme plutôt que d'écrire le programme).

Question 13 Écrire la fonction decodageVigenereAuto(t', n) qui prend en argument le tableau t' de taille n représentant le texte crypté ; et qui retourne le texte t d'origine. (On n'hésitera pas à recopier des parties de texte dans des tableaux intermédiaires).

* *
*