Université de Bordeaux — **Algèbre et calcul formel – Master**
Mathématiques — Année 2016–2017

FEUILLE D'EXERCICES n° 7

Travail sur machine
PGCD, théorème de Bézout, théorème des restes chinois

Exercice 1 – [Calcul modulaire]
Il y a plusieurs méthodes pour faire des calculs modulo un entier n.

On peut utiliser `mod` : essayer `mod(5,7)**(-1)`. Remarquer (par exemple en utilisant la commande `parent`) que le résultat obtenu reste un entier modulo 7. Si l'on veut ajouter un entier, ça va marcher quand même (essayer par exemple `mod(5,7)**(-1)+2`). Par contre, une commande du style

```
mod(5,7)**(-1)+mod(3,5)**(-1)
```

n'est pas acceptée, et c'est bien normal. Pour revenir sur $\mathbb{Z}$, on peut utiliser la commande `lift`. Par exemple, la commande précédente, qui n'a aucun sens telle qu'elle est écrite ci-dessus, prend un sens dans $\mathbb{Z}$.

```
lift(mod(5,7)**(-1))+lift(mod(3,5)**(-1))
```

Essayer

```
((25*16)*lift((-mod(25,21)*mod(16,21))^(-1))
+21*16*lift((11*(mod(21,25)*mod(16,25))^(-1)))
+21*25*lift((mod(21,16)*mod(25,16))^(-1)))%(21*25*16)
```

À quoi correspond un tel calcul ?

Exercice 2 – [Restes Chinois]

1) En utilisant la commande `crt`, résoudre les systèmes

$$\begin{cases} x \equiv -1 \mod 21 \\ x \equiv 11 \mod 25 \\ x \equiv 1 \mod 16 \end{cases}, \quad \begin{cases} x \equiv -1 \mod 21 \\ x \equiv 11 \mod 15 \\ x \equiv 1 \mod 10 \end{cases} \text{ et } \begin{cases} x \equiv -1 \mod 21 \\ x \equiv 10 \mod 15 \\ x \equiv 1 \mod 10 \end{cases}$$

Note. Cette commande `crt` s'applique à tout anneau euclidien par exemple à $k[x]$, où k est un corps.

Note. On connaît bien l'algorithme correspondant dans le cas où les moduli sont deux à deux premiers entre eux. Il ne s'applique tel qu'à l'un de ces trois exemples. L'exercice 4 de la feuille 6 montre comment on peut adapter l'algorithme au cas de moduli non deux à deux premiers entre eux.

2) Après une série de rapines, une troupe de 14 pirates partage (équitablement) le butin et laisse le reliquat, 3 écus, au cuisinier chinois, le $15^{\text{ème}}$ homme d'équipage. Le lendemain, un flibustier tombe à la mer et n'est pas repéché à temps ; après avoir envisagé le versement de sa part à des œuvres, les pirates refont le partage en incluant sa part ; le cuistot reçoit 2 écus. La semaine se passe sans encombres, mais trois pirates ivres se disputent sur leurs parts respectives et deux d'entre eux sont tués. Notre cuisinier récupère 5 écus. La fin du mois est mauvaise et 3 pirates périssent dans une embuscade ; mais le cuistot est content : il garde ses 5 écus.

Quel magot peut-il espérer empocher quand il décide d'empoisonner le reste de la bande ?

3) En utilisant crt, déterminer l'unique polynôme de $\mathbb{Q}[x]_3$ tel que $P(0) = 3$, $P(1) = -1$, $P(2) = 1$, $P(3) = 2$.

4) Même question dans $\mathbb{Q}[x]_6$ avec les contraintes $P(0) = 2$, $P(1) = 1$, $P'(1) = -1$, $P''(1) = 3$, $P'''(1) = 6$, $P(2) = 1$, $P'(2) = -1$.

Exercice 3 – [Un peu de calcul matriciel]

Dans l'exercice 3 de ce travail, on étudie la résolution dans $\mathbb{Z}^n$ d'une équation $a_1x_1 + \cdots + a_nx_n = b$. Pour cela, nous utiliserons un peu de calcul matriciel. C'est pourquoi cet exercice donne quelques commandes sage pour de tels calculs.

1) On peut définir un vecteur et une matrice de la manière suivante.

```
w = vector([1,1,-4])
A = matrix([[1,2,3],[3,2,1],[1,1,1]]); A
```

Remarquons d'abord que les indices commencent à 0. Si l'on tape `A[0,0]`, on obtient `1`.

2) Exécuter les commandes

```
A.det()
A*w
w*A
parent(A)
parent(w)
```

3) On peut aussi commencer par définir l'espace matriciel où se trouveront les matrices. Exécuter les commandes

```
EM=MatrixSpace(ZZ,3)
EV=VectorSpace(ZZ,3)
```

Erreur ! C'est que $\mathbb{Z}$ n'est pas un corps, on ne peut donc pas définir d'espace vectoriel sur $\mathbb{Z}$. On peut écrire à la place

```
EV=VectorSpace(QQ,3)
```

ou bien définir le module $\mathbb{Z}^3$ sur $\mathbb{Z}$.

```
EV=FreeModule(ZZ,3)
w=EV([1,1,-4])
A=EM([1,2,3,3,2,1,1,1,1])
A,w
V=EM(1)
V
V[0,1]=2
V
```

Exercice 4 – [Autour de Bézout]

Note. Les commandes de l'exercice précédent ne sont utiles qu'à partir de la question **??**.

1) Le logiciel sage permet de calculer le pgcd de deux entiers à l'aide de la commande `gcd`. La commande `xgcd` donne en plus les coefficients de Bézout. En utilisant cette commande, écrire un algorithme qui prend en entrée une famille d'entiers $(a_1, \dots, a_n)$ et donne en sortie $(u_1, \dots, u_n) \in \mathbb{Z}^n$ tel que

$$\sum_{i=1}^{n} a_i u_i = \mathrm{pgcd}(a_1, \dots, a_n).$$

Remarque. La commande `gcd([a1,...,an])` donne $\mathrm{pgcd}(a_1, \dots, a_n)$. Par contre, `xgcd([a1,...,an])` ne donne rien.

2) Soient deux entiers a et b tels que $(a, b) \neq (0, 0)$. Soient u et v deux entiers tels que $au + bv = \mathrm{pgcd}(a, b)$. Déterminer en fonction de a, b, u, v et $d = \mathrm{pgcd}(a, b)$ une matrice $U \in \mathcal{M}_2(\mathbb{Z})$ de déterminant 1 telle que $(a, b)U = (\mathrm{pgcd}(a, b), 0)$.

3) En s'inspirant de la question précédente, montrer qu'il existe une matrice U dans $\mathcal{M}_n(\mathbb{Z})$ de déterminant 1) telle que

$$(a_1, \dots, a_n)U = (\mathrm{pgcd}(a_1, \dots, a_n), 0, \dots, 0)$$

et programmer le calcul de cette matrice.

Pour cela, on peut d'abord calculer une matrice V_1 telle que

$$(a_1, \dots, a_n)V_1 = (\mathrm{pgcd}(a_1, a_2), 0, a_3, \dots, a_n),$$

puis une matrice V_2 telle que

$$(\mathrm{pgcd}(a_1, a_2), 0, a_3, \dots, a_n)V_2 = (\mathrm{pgcd}(a_1, a_2, a_3), 0, 0, a_4, \dots, a_n)$$

et itérer le processus. Alors la matrice U cherchée est égale au produit des V_i.

4) Si $b \in \mathbb{Z}$ et $(a_1, \dots, a_n) \in \mathbb{Z}^n$ sont fixés, expliquer (sans le programmer) comment résoudre l'équation $\sum a_i x_i = b$ en nombres entiers (x_i) [*intercaler* $UU^{-1} =$ Id]. Résoudre les équations $1009x + 345y + 56z = 1$ et $143x + 195y + 165z = 3$.

5) Comment résoudre un système de plusieurs équations sur $\mathbb{Z}^n$? Il n'est pas demandé de programmer ce calcul.