Université Bordeaux 1 **MHT839 – Master**
Mathématiques Année 2012–2013

Devoir Surveillé, 12 avril 2013
Durée 2h.

Quelques commandes sage sont rappelées en fin d'énoncé.

Soit p un nombre premier. Ce travail porte sur un algorithme permettant de calculer la partie sans facteur carré d'un polynôme de $\mathbb{F}_p[x]$, au sens rappelé ci-dessous.

Soit P un polynôme non nul de $\mathbb{F}_p[x]$. Alors il existe un entier naturel r, des polynômes unitaires irréductibles deux à deux distincts $P_1, \ldots, P_r$ de $\mathbb{F}_p[x]$, des entiers naturels non nuls $e_1, \ldots, e_r$ et un élément $\alpha \in \mathbb{F}_p^*$ tels que

$$P = \alpha \prod_{i=1}^{r} P_i^{e_i}.$$

Définition 1. La partie sans facteur carré de P est égale à

$$\prod_{i=1}^{r} P_i.$$

1) Soit $Q = \sum_{i=0}^{n} q_i x^i \in \mathbb{F}_p[x]$. Montrer que $Q^p = \sum_{i=0}^{n} q_i x^{pi}$.

2) Soit $R \in F_p[x]$. Montrer que la dérivée R' de R est nulle si et seulement s'il existe $Q \in \mathbb{F}_p[x]$ tel que $R = Q^p$.

3) Soit P un polynôme unitaire de $\mathbb{F}_p[x] \setminus \{0\}$. On l'écrit comme ci-dessus (avec $\alpha = 1$ puisque P est unitaire)

$$P = \prod_{i=1}^{r} P_i^{e_i}.$$

Soient $I = \{i \in \{1, \ldots, r\} \ : \ p \not| e_i\}$, $J = \{i \in \{1, \ldots, r\} \ : \ p | e_i\}$,

$$U = \frac{P}{\mathrm{pgcd}(P, P')} \quad \text{et} \quad V = \frac{P}{\mathrm{pgcd}(U^d, P)},$$

où $d = \deg P$. Montrer que

$$U = \prod_{i \in I} P_i \quad \text{et} \quad V = \prod_{i \in J} P_i^{e_i}.$$

4) Montrer que V s'écrit $V = W^p$, où $W \in \mathbb{F}_p[x]$.

5) Soit $P = x^{11} + x^{10} + 2x + 2 \in \mathbb{F}_5[x]$. Calculer successivement P', $\mathrm{pgcd}(P, P')$, U, $\mathrm{pgcd}(U^d, P)$, V et W. En déduire la partie sans facteur carré de P (on ne demande pas de justification, les calculs peuvent être faits sur sage et recopiés sur papier).

6) On peut calculer la partie sans facteur carré de P en utilisant la méthode suivante.

On calcule les polynômes U et V de la question 3. On cherche $W \in \mathbb{F}_p[x]$ tel que $V = W^p$, puis on recommence en appliquant le même processus à W (à la place de P), jusqu'à obtenir un polynôme V égal à 1. La partie sans facteur carré est alors le produit des polynômes U obtenus à chacune des étapes de cet algorithme.

Écrire cet algorithme pour sage. On pourra décomposer le problème : écrire d'abord une fonction `UV` qui calcule U et V, puis une fonction `Racine` qui, étant donné V, calcule W, et enfin la fonction principale `PSFC` (pour partie sans facteur carré). Pour tester cet algorithme, on pourra par exemple utiliser le polynôme de la question 5, et $P = (x+1)^4(x^2+2)^5(x+2)^{25} \in \mathbb{F}_5[x]$.

7) Soit $q = p^k$, où k désigne un entier strictement supérieur à 1. Sur $\mathbb{F}_q[x]$, l'algorithme de la question 6 s'applique de la même façon, à condition de savoir calculer un polynôme $W \in \mathbb{F}_q[x]$ tel que $W^p = V$, si $V \in \mathbb{F}_q[x]^p$.

a) Soit $\mathbb{F}_{27}$ défini comme le quotient $\mathbb{F}_3[y]/(y^3 - y + 1)$, et soit a l'image de y dans ce quotient. Sur sage, cela peut se coder de la manière suivante.

```
A.<y>=PolynomialRing(GF(3))
k.<a>=GF(27,modulus=y**3-y+1)
```

Soit $V = x^{12} + (a^2+1)x^6 + (a+2)x^3 + a \in \mathbb{F}_{27}[x]$. Calculer $W \in \mathbb{F}_{27}[x]$ tel que $W^3 = V$ (on fera les calculs sur sage, mais on indiquera sur papier la méthode employée).

b) Écrire un programme `RacineFq` qui, étant donné un polynôme $V \in \mathbb{F}_q[x]^p$, calcule $W \in \mathbb{F}_q[x]$ tel que $W^p = V$.

Quelques commandes sage.

`P.derivative()` (Pour calculer la dérivée de P).

`GF(5)` (pour définir le corps fini $\mathbb{F}_5$).

`pr.<x>=PolynomialRing(K)` (pour l'anneau $K[x]$).

`gcd(P,Q)` : $\operatorname{pgcd}(P, Q)$.

`P.degree()` : $\deg P$.

Pour calculer $\sum_{i=0}^{t} l[i]$, où l est une liste : `add(l[i] for i in range(t+1))`.

Pour le quotient d'un entier a par un entier $b \neq 0$: `a//b`.

Pour le quotient et le reste de P par $Q \neq 0$ (dans `pr`$= K[x]$) : `P.quo_rem(Q)`.

Pour la liste des coefficients d'un polynôme P : `P.coeffs()`

Pour calculer $Q^k \bmod P$ (dans `pr`$= K[x]$) :

```
quoP.<y>=pr.quotient_ring(P)
(Q(y)**k).lift()
```