Feuille TD 3 - Codes correcteurs - Construction de corps fini

1 Code de répétition pure

On considère un code de répétition sur $V = \{0, 1\}$, c'est à dire un code (mk, k) où chaque groupe de k bits est simplement répété m fois.

1 Quelle est la distance minimale entre deux mots de code ?

Correction: *deux mots distincts ont au moins un bit de source de distincts; comme ce bit est répétéé m fois,* $\delta(C) \geq m$. *De plus soient* $\omega_0 = C(O^k) = 0^{mk}$ *et* $\omega_1 = C(1O^{k-1}) = 1^m 0^{m(k-1)}$*; d'où* $d_H(\omega_0, \omega_1) = m$ *: donc* $\delta(C) \leq m$. *Finalement:* $\delta(C) = m$.
NB: autre solution: comme un code de répétition sur un corps (ici $\mathbb{F}_2$*) est un code linéaire,* $\delta(C)$ *est le minimum des poids:* $Min(w(x); x \neq [0 \dots 0]) = m$.

2 Proposer un code de répétition qui soit 2-correcteur.

Correction: *Il faut* $m = 2 * 2 + 1 = 5$. *Donc* $(5, 1)$ *convient.* $(5k, k)$ *aussi.*

3 Quel est son rendement ?

Correction: $(5k, k)$ *est de rendement* $\frac{k}{5k} = 0.2$.

Dans toute la suite on considère un code de répétition $(n = 5k, k)$ et on désire écrire une procedure de décodage de prototype :

```
procedure decoder (y :  in array[0 ..  n-1] of bit; s :  out array[0..  k-1] of bit; erreur :
out boolean) ;
```

qui retourne en sortie le mot décodé s et un booléen qui vaut vrai si s peut être considéré correct (i.e. pas d'erreurs ou des erreurs corrigées) et faux sinon (i.e. des erreurs ont été détéctées mais pas corrigées). On étudie différentes implantations de `decoder`.

4 Combien d'erreurs peuvent être détectées ? Programmer `decoder` pour détecter un nombre maximal d'erreurs mais n'en corriger aucune.

Correction: *La distance est 5; le code permet de détecter 4 erreurs au moins (voire plus si elles concernent différents chiffres de source).*

```
procedure decoder (y :  in array[0 ..  n-1] of bit;
          s :  out array[0..k-1] of bit; erreur :  out boolean) is
  begin
   erreur := false;
    for (i=0; i<k; i++) s[i] := lirecar(); // Initialisation de s
    for (i=k; i<n; i++)
      if (lirecar() != s[i mod k]) erreurs := true ;
  end decoder ;
```

5 Combien d'erreurs peuvent être corrigées ? Programmer `decoder` pour corriger un nombre maximal d'erreurs. Combien d'erreurs sont détectées mais non corrigées ?

Correction: *La distance est 5; le code permet de corriger 2 erreurs; il suffit de calculer pour chaque symbole de source le nombre de 1 associé. Si on a une majorité de 1 (i.e. plus de 3), on déduit que le symbole source était 1; sinon la source était 0. Ceci qui corrige au plus 2 erreurs pour les 5 bits émis correspondants à un même bit de source.*

```
procedure decoder (y :  in array[0 ..  n-1] of bit;
          s :  out array[0..k-1] of bit; erreur :  out boolean) is
    NbrUn :  out array[0..k-1] of integer ;
  begin
    for (i=0; i<k; i++) NbrUn[i] := 0; // Initialisation de s
    erreur := false;
    for (i=k; i<n; i++)
      if (lirecar() == '1') then NbrUn[i mod k] := NbrUn[i mod k] = 1 ; end if;
    for (i=0; i<k; i++)
      if (NbrUn[i] >=3) s[i] := '1'; else s[i] := '0'; end if ;
  end decoder ;
```

5 Programmer `decoder` pour corriger une seule erreur mais en détecter jusqu'à 3.

Correction: *Le code corrige une erreur; il peut détecter en outre 2 ou 3 erreurs sans les corriger (mais pas 4 !). La distance est 5; si on recçoit 4 symboles identiques (4 '0' ou 4 '1'), on suppose qu'il n'y a eu qu'une erreur et on fait la corretcion. Sinon on renvoie un signal d'erreur. On corrige toutes les erreurs simples et on détecte les cas de 2 et 3 erreurs correspondant à un même bit de source.*

```
procedure decoder (y :  in array[0 ..  n-1] of bit;
          s :  out array[0..k-1] of bit; erreur :  out boolean) is
    NbrUn :  out array[0..k-1] of integer ;
  begin
    for (i=0; i<k; i++) NbrUn[i] := 0; // Initialisation de s
    erreur := false;
    for (i=k; i<n; i++)
      if (lirecar() == '1') then NbrUn[i mod k] := NbrUn[i mod k] = 1 ; end if;
    for (i=0; i<k; i++)
      if (NbrUn[i] >=4) s[i] := '1';
      elsif (NbrUn[i] <=1) s[i] := '0';
      else erreur := true ;
  end decoder ;
```

7 Ce code est-il linéaire ? Si oui, expliciter une matrice génératrice et la mettre sous forme canonique pour $k = 1$ et $k = 2$.

Correction: *Le code est évidemment linéaire et une matrice génératrice est :* $G = [I_k I_k \dots I_k)$ *où* I_k *est la matrice identité* $k \times k$ *et est répétée* m *fois;* G *est bien sous forme canonique.*

- *pour un code (5,1,5) (i.e.* $k = 1$*) :* $G = [11111]$
- *pour un code (10,2,5) (i.e.* $k = 2$*) :* $G = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$

8 On veut corriger des paquets d'erreurs aléatoires, très peu probables, mais pouvant affecter jusqu'à e bits consécutifs (on parle de *paquet d'erreurs*).
Quelle longueur e de paquet d'erreurs le code de répétition $(5k, k)$ permet-il de corriger ? Proposer un code permettant de corriger des paquets d'erreurs de longueur inférieure à 100 (on explicitera k).

Correction: *L'entrelacement des bits de source fait que l'on peut corriger tout paquet d'erreurs de longueur inférieure à* $2k$ *avec le code de répétition* $(5k, k)$*. Pour corriger un paquet de longueur 100, il suffit de prendre* $k = e/2 = 50$.

2 Construction de corps finis. L'exemple de $\mathbb{F}_4$ et $\mathbb{F}_8$

On admet le théorème suivant (cf annexes et polycopié pour compléments).

- *Un ensemble V de cardinal fini q peut être muni d'une structure de corps ssi q est de la forme $q = p^k$ où p est un nombre premier et k un entier non nul.*
- *Soit $q = p^k$ où p est un nombre premier et k un entier non nul; il existe un unique corps fini à q éléments, noté $\mathbb{F}_q$.*
  Ce corps est de caractéristique p (i.e. $p \times x = 0 \forall x \in \mathbb{F}_{p^k}$

  NB: Comme $\mathbb{Z}/p\mathbb{Z}$ est un corps, on en déduit $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z}$.

Une caractérisation possible de $\mathbb{F}_q$ avec $q = p^k$e t p premier. On note $\mathbb{F}_p[X]$ l'anneau des polynômes à coefficients dans $\mathbb{F}_p$.
Un polynôme $Q \in \mathbb{F}_p[X]$ de degré k est dit *irréductible* ssi il est premier avec tous les polynômes de $\mathbb{F}_p[X]$ de degré inférieur à k; i.e.

$$\forall R \in \mathbb{F}_p[X] \quad \text{pgcd}(Q, R) = 1.X^0$$

On admet que pout tout p premier et k entier, il existe des polynômes irréductibles de degré k dans $\mathbb{F}_p[X]$.
Comme dans le cas des entiers, l'algorithme d'Euclide appliqué aux deux polynômes Q et R, premiers entre eux, permet de calculer les coefficients de Bezout A et B : tels que

$$A(X) \times R(X) + B(X) \times Q(X) = \text{pgcd}(Q, R) = 1.X^0.$$

On en déduit que R est inversible modulo Q, d'inverse $R^{-1} = A \bmod Q$. Par suite, l'anneau des restes $(\mathbb{F}_p[X]/Q, +_{\text{mod } Q}, \times_{\text{mod } Q}, 0.X^0, 1.X^0)$ est un corps. Comme $\mathbb{F}_p[X]/Q$ est de cardinal p^k, on en déduit:

$$\mathbb{F}_{p^k}[X] \equiv \mathbb{F}_p[X]/Q.$$

1. Exemple: le corps $\mathbb{F}_4$.

a Donner une condition nécessaire et suffisante pour qu'un polynôme de dans $\mathbb{F}_2[X]$ de degré $2 \leq n \leq 3$ soit irréductible. En déduire tous les polynômes irréductibles de degré 2 et 3.

Correction: *Condition nécessaire pour que P soit irréductible de degré $n \leq 2$: doit être de la forme $X^n + X^0 + \sum_{i=1}^{n-1} a_i X^i$ avec $\sum_{i=1}^{n-1} a_i = 1 \mod 2$.*
Cette condition est suffisante pour $2 \leq n \leq 3$. D'où
De degré 2 : $1 + X + X^2$ car n'admet ni 0 ni 1 comme racine.
De degré 3 : $X^3 + x + 1$ et $X^3 + X^2 + 1$.

b Soit $\mathbb{F}_4 = \{e_0, e_1, e_2, e_3\}$, avec la convention e_0 élément neutre pour l'addition et e_1 élément neutre pour la multiplication.
En utilisant la question a, expliciter comment effectuer les opérations $(+, \times,$ inverse) dans F_4.

Correction: *On choisit $Q = X^2 + X + 1$ polynôme irréductible de degré 2 de $\mathbb{F}_2[X]$. On a : $\mathbb{F}_4 \equiv \mathbb{F}_2[X]/X^2 + X + 1$.*

D'où $\mathbb{F}_4 = \{0, 1, X, X+1\}$. *On effectue les opérations d'addition et de multiplication modulo* $(X^2 + X + 1)$ *d'où les tables.*

$+$	e_0	e_1	e_2	e_3
e_0	e_0	e_1	e_2	e_3
e_1		e_0	e_3	e_2
e_2			e_0	e_1
e_3				e_0

$\times$	e_0	e_1	e_2	e_3
e_0	e_0	e_0	e_0	e_0
e_1		e_1	e_2	e_3
e_2			e_3	e_1
e_3				e_2

2. Construction de $\mathbb{F}_{p^d}$ à partir d'un polynôme primitif. Un polynôme irréductible $P = \alpha^d + ...$ de degré d dans $\mathbb{F}_p[\alpha]$ est dit **primitif** si et seulement si α est générateur de $\mathbb{F}^\star_{2^d}$, i.e.

$$\mathbb{F}^\star_{p^d} \equiv \{\alpha^i \bmod P;\ 1 \leq i < q\}.$$

Pour $d = 3, .., 10$, les polynômes suivants à coefficients dans $\mathbb{F}_2$ sont primitifs :

degré d	Polynôme primitif
3	$1 + \alpha + \alpha^3$
4	$1 + \alpha + \alpha^4$
5	$1 + \alpha^2 + \alpha^5$
6	$1 + \alpha + \alpha^6$

degré d	Polynôme primitif
7	$1 + \alpha^3 + \alpha^7$
8	$1 + \alpha + \alpha^2 + \alpha^7 + \alpha^8$
9	$1 + \alpha^4 + \alpha^9$
10	$1 + \alpha^3 + \alpha^{10}$

c Proposer une construction de $\mathbb{F}_8$ utilisant deux tables auxiliaires (on s'inspirera de l'annexe qui donne la construction de $\mathbb{F}_{256}$).

Correction: *D'après la table,* $1 + \alpha + \alpha^3$ *est un polynôme primitif de* $\mathbb{F}_2$. *On a :* $\mathbb{F}_8 \equiv \mathbb{F}_2[X]/X^3 + X + 1$ *et on a :*

$$\mathbb{F}_8 = \{e_0 = 0; e_1 = \alpha^0; e_2 = \alpha^1; e_3 = \alpha^2; e_4 = \alpha^3; e_5 = \alpha^4; e_6 = \alpha^5; e_7 = \alpha^6;$$

toutes les multiplications par α *étant faites modulo* $1 + \alpha + \alpha^3$.

On code chaque élément par un indice: $e_0 = 0$ *par l'indice [-1] et, pour* $i = 0 \ldots 6$, *on code* $e_{i+1} = \alpha^i$ *par l'indice* $[i]$.

Pour les opérations, on traite à part le cas où un opérande est nul (i.e. le cas $e_0 = 0 = [-1]$*); sinon, on a :*

- *si* $i, j \neq -1$ *:* $[i] \times_{\mathbb{F}_8} [j] = [(i + j) \mod 7]$
- *si* $i \neq -1$*:* $[i]^{-1} = [(7 - i) \mod 7]$
- *si* $i \neq -1$*:* $-[i] = [i]$
- *pour* $j > i \geq 0$*:* $[i] +_{\mathbb{F}_8} [j] = [i] \times_{\mathbb{F}_8} (1 + [j]/[i]) = [i] \times_{\mathbb{F}_8} [[0] + [j - i]]$.
  On stocke l'indice $[k]$ *de* $[0] + [j - i]$ *dans une table* $T[j - i] = k$*; on a alors:* $[i] +_{\mathbb{F}_8} [j] = [i + T[j - i]]$

Le tableau ci-dessous résume les correspondances pour l'indice et la table T *:*

	Polynôme associé	$[i]$	$T(i) = [1 + [i]]$
e_0	0	$[-1]$	$[0]$
e_1	1	$[0]$	$[-1]$
e_2	α	$[1]$	$[3]$
e_3	α^2	$[2]$	$[6]$
e_4	$\alpha^3 = \alpha + 1$	$[3]$	$[1]$
e_5	$\alpha^4 = \alpha^2 + \alpha$	$[4]$	$[5]$
e_6	$\alpha^5 = \alpha^2 + \alpha + 1$	$[5]$	$[4]$
e_7	$\alpha^6 = \alpha^2 + 1$	$[6]$	$[2]$

Annexe 1. Structure des corps finis. *Cet exercice en annexe démontre la plupart des propriétés utilisées pour le calcul dans les corps finis.*
Soit K un corps fini quelconque de cardinal $q > 0$.

1 A l'aide de l'application $\psi : \mathbb{Z} \to K$, définie par :

$$\forall n \in \mathbb{Z}\ \psi(n) = \underbrace{1 + 1 + \ldots + 1}_{n \text{ fois}} = n.1,$$

montrer qu'il existe un unique nombre premier p, dit *caractéristique* de K, tel que : $\forall x \in K\ px = 0$.

Correction: $\Psi(0) = 0, \Psi(1) = 1, \Psi(n_1 + n_2) = \Psi(n_1) + \Psi(n_2), \Psi(n_1 n_2) = \Psi(n_1) \times \Psi(n_2)$ $\Longrightarrow$ *Ψ homomorhisme d'anneau. Comme K est fini et Z infini, Ψ est non-injectif* $\Longrightarrow \exists n \neq 0 : \Psi(n) = 0$.
Si n non premier; soit $n = n_1 n_2$: on a $\Psi(n_1) \times \Psi(n_2) = 0$ donc $\Psi(n_1) = 0$ ou $\Psi(n_2) = 0$ (K est un corps donc anneau intègre). Donc il existe p premier tel que $\Psi(p) = 0$.
Unicité de p : si p_1 et p_2 premiers et $\Psi(p_1) = \Psi(p_2) = 0$. Bezout $\Longrightarrow \exists a, b$: $ap_1 + bp_2 = 1$ d'ou $\Psi(1) = 0$: absurde.

2 En déduire que le cardinal de K est une puissance de p, en se servant du fait que K est un espace vectoriel sur ses sous-corps. Indication : exhiber un sous-corps de K isomorphe à $\mathbb{F}_p$.

Correction: *Soit Ψ_p la restriction de Ψ à F_p. Soit $k = \{\Psi_p(0), \Psi_p(1), \Psi_p(p-1)\}$; k est isomorphe à F_p (Ψ_p est injectif et k et F_p ont même cardinal); donc k est un sous-corps de K.*
K est un espace vectoriel sur k; soit m sa dimension. Alors card(K)=card(k)^m.

3 On admet que deux corps de même cardinal q sont isomorphes à un même corps, que l'on note $\mathbb{F}_q$. On s'intéresse maintenant à la représentation de cet objet, afin de pouvoir faire des calculs.

- Soit $P \in \mathbb{F}_p[X]$, irréductible de degré m. Quelle est la structure de $\mathbb{F}_p[X]/(P(X))$? NB : il est possible de construire cet objet, par exemple à l'aide de polynômes cyclotomiques.

  Correction: *C'est un anneau (cf exo1); Comme tout élément $\neq 0$ est inversible, c'est un corps. Sa caractéristique est p; son cardinal est p^m.*
  Polynôme cyclotomique: facteur irréductible de $X^n - 1$. Soit d un entier facteur de n; on a: $x^n - 1 = \prod_{d\ n} \Phi_d(x)$ et $\Phi_d(x) = \prod_{1 \leq k \leq d, \gcd(k,d)=1} (x - \omega^k)$ (ω est une racine d-ieme de l'unité), soit encore $\Phi_d(x) = \prod_{d\ n} (x^d - 1)^{\mu(n/d)}$ [Knuth ex32 page 440].

- Montrer que $\mathbb{F}_q^*$ est un groupe (multiplicatif) cyclique et que $\forall x \in \mathbb{F}_q : x^q = x$.

  Correction: *$F_p = Z/pZ$ est un corps, donc $F_p[x]$ est principal. Comme P est irreductible, $F_p[X]/PF_p[X]$ (isomorphe à F_q) est aussi principal. Dans un anneau principal, tout idéal est monogène; Donc F_q est monogène. Donc $\exists a \in \mathbb{F}_q^*$: $F_q = \{a^i, i \in \mathbb{Z}\}$. Par suite F_q^* est cyclique.*
  Comme F_q^* est de cardinal $(q-1)$, $a^{q-1} = 1$. On a aussi, $\forall x \in \mathbb{F}_q^* : x^q = a^{iq} = a^i a^{q-1} i = a^i = x$, qui est aussi vrai pour $x = 0$.

Déduire de ces deux points deux représentations différentes de $\mathbb{F}_q$, et discuter de la réalisabilité des opérations arithmétiques selon la représentation choisie.

Correction: – *sol 1: On calcule modulo un polynôme irréductible de degré m à coefficients dans F_p.*
Brutalement le coût est $m \log m \log p \log\log p \log\log\log p$. En regroupant les multiplications de coef et comme $p = O(q)$ et $m = O(\log q)$, on obtient $\log q \log\log q \log\log\log q$

– *sol 2 : Soit a un élément quelconque non nul et non unitaire de F_p et e_i le vecteur de F_p^m dont toutes les composantes sont nulles sauf la ieme qui vaut a. En utilisant la question 2, on obtient que $B = (e_i)_{i=1..m}$ est une base de F_p^m qui est isomorphe à F_q. On représente donc F_q par un vecetur de m entiers dans Z/pZ. L'addition est l'addition de vecteur. La multiplication est plus compliquee (équivalente a un produit de polynôme); on peut la tabuler. Le coût des opérations est : $\log^2 q \log\log q \log\log\log q$.*

– *sol 3 : On prend un symbole a qui représente un élément générateur de F_q; tout elément x de $F_q^\star$ est représenté par l'entier i tel que $x = a^i$. 0 est représenté par un symbole N. La multiplication correspond alors à l'addition d'entiers de $\log p$ bits, avec la convention $N.x = N$, donc de coût $\log q$.*
Pour l'addition, c'est alors plus complique : on peut aussi la tabuler. Le coût des opérations est en $\log q \log\log q$... si on tabule pas et $\log q$ (addition d'entiers plus petit que q ou calcul d'adresse pour accès à la table qui contient q^2 entrées) en $\log q$ théorique et $O(1)$ pratique.
Évidemment ce n'est pas un coût uniforme (précalcul) mais c'est le plus efficace en pratique pour q petit.

Annexe 2: Construction de $\mathbb{F}_{256}$.
Pour construire le corps à 255 éléments $\mathbb{F}_{256}$, on considère $P(\alpha) = 1 + \alpha + \alpha^2 + \alpha^7 + \alpha^8$ primitif sur $\mathbb{F}_2[X]$ de degré 8. $\mathbb{F}_{256}$ est donc isomorphe à $\mathbb{Z}/2\mathbb{Z}[\alpha]/P(\alpha)$.

Comme P est primitif, il y a –au moins– deux représentations possibles des éléments du corps en machine, qui peuvent être utilisées de manière complémentaire pour réaliser l'addition et la multiplication.

- $\mathbb{F}_{256} = \{0\} \cup \{\alpha^i : 0 \le i \le 254\}$, c'est à dire que le monôme α^i $(0 \le i \le 254)$ est représenté par l'entier i.
Par convention, on représente l'élément 0 par -1. La multiplication est facile à implémenter dans ce cas. Si on note $[i]$ l'élément du corps fini représenté par l'entier i, on a : $[i] \times [j] = \begin{cases} [i+j \mod 255] & \text{si } i, j \neq -1 \\ [-1] & \text{sinon} \end{cases}$
Pour la division : si $i \neq -1$, $[i]^{-1} = [255 - i]$.

- $\mathbb{F}_{256} = \{q \in \mathbb{Z}/2\mathbb{Z}[x], \deg(q) \le 7\}$, c'est à dire l'anneau des restes modulo le polynôme P. Comme le corps de base est $\mathbb{Z}/2\mathbb{Z}$, chaque polynôme $q(x)$ peut alors représenté par un entier: sa valeur $q(2)$ en $x = 2$.
L'addition est alors facile à implémenter. Si on note (i) l'élément du corps fini représenté par l'entier i, on a $(i) + (j) = (i \text{ xor } j)$ où *xor* désigne le ou-exclusif bit à bit.

- Une alternative pour l'addition est de tabuler les valeurs de $(1+[k])$ pour tout $[k]$ non nul (ie $k \neq -1$). En effet, soit $\text{TabXplusUn}(k) = j$ avec $[j] = 1 + [k]$. On a alors, pour $j \ge i$ et $i \neq -1$: $[i] + [j] = [i].(1 + [j]/[i]) = [i].(1 + [j - i mod 255]) = [i].[\text{TabXplusUn}(j - i)] = [i + \text{TabXplusUn}(j - i) mod 255]$.
En pratique, pour le cas d'un petit corps (comme ici avec 256 éléments), cette tabulation du corps est préférable [cf `www.linalg.org/field.html` et en particulier `Givaro`].

Les deux dernières colonnes du tableau ci-dessous donnent le codage correspondant à $\mathbb{F}_{256}$ pour ces deux représentations par entier.

α^i	$Q_i = \alpha^i \bmod P$	$[i]$	$(Q_i(2))$
0	0	-1	0
1	1	0	1
a^1	a^1	1	2
a^2	a^2	2	4
a^3	a^3	3	8
a^4	a^4	4	16
a^5	a^5	5	32
a^6	a^6	6	64
a^7	a^7	7	128
a^8	$a^7 + a^2 + a + 1$	8	135
a^9	$a^7 + a^3 + 1$	9	137
a^{10}	$a^7 + a^4 + a^2 + 1$	10	149
a^{11}	$a^7 + a^5 + a^3 + a^2 + 1$	11	173
a^{12}	$a^7 + a^6 + a^4 + a^3 + a^2 + 1$	12	221
a^{13}	$a^5 + a^4 + a^3 + a^2 + 1$	13	61
a^{14}	$a^6 + a^5 + a^4 + a^3 + a$	14	122
a^{15}	$a^7 + a^6 + a^5 + a^4 + a^2$	15	244
a^{16}	$a^6 + a^5 + a^3 + a^2 + a + 1$	16	111
a^{17}	$a^7 + a^6 + a^4 + a^3 + a^2 + $ a	17	222
a^{18}	$a^5 + a^4 + a^3 + a + 1$	18	59
a^{19}	$a^6 + a^5 + a^4 + a^2 + a$	19	118
a^{20}	$a^7 + a^6 + a^5 + a^3 + a^2$	20	236
a^{21}	$a^6 + a^4 + a^3 + a^2 + a + 1$	21	95
a^{22}	$a^7 + a^5 + a^4 + a^3 + a^2 + a$	22	190
a^{23}	$a^7 + a^6 + a^5 + a^4 + a^3 + a + 1$	23	251
a^{24}	$a^6 + a^5 + a^4 + 1$	24	113
a^{25}	$a^7 + a^6 + a^5 + a$	25	226
a^{26}	$a^6 + a + 1$	26	67
a^{27}	$a^7 + a^2 + a$	27	134
a^{28}	$a^7 + a^3 + a + 1$	28	139
a^{29}	$a^7 + a^4 + 1$	29	145
a^{30}	$a^7 + a^5 + a^2 + 1$	30	165
a^{31}	$a^7 + a^6 + a^3 + a^2 + 1$	31	205
a^{32}	$a^4 + a^3 + a^2 + 1$	32	29
a^{33}	$a^5 + a^4 + a^3 + a$	33	58
a^{34}	$a^6 + a^5 + a^4 + a^2$	34	116
a^{35}	$a^7 + a^6 + a^5 + a^3$	35	232
a^{36}	$a^6 + a^4 + a^2 + a + 1$	36	87
a^{37}	$a^7 + a^5 + a^3 + a^2 + a$	37	174
a^{38}	$a^7 + a^6 + a^4 + a^3 + a + 1$	38	219
a^{39}	$a^5 + a^4 + 1$	39	49
a^{40}	$a^6 + a^5 + a$	40	98
a^{41}	$a^7 + a^6 + a^2$	41	196
a^{42}	$a^3 + a^2 + a + 1$	42	15
a^{43}	$a^4 + a^3 + a^2 + a$	43	30
a^{44}	$a^5 + a^4 + a^3 + a^2$	44	60
a^{45}	$a^6 + a^5 + a^4 + a^3$	45	120
a^{46}	$a^7 + a^6 + a^5 + a^4$	46	240
a^{47}	$a^6 + a^5 + a^2 + a + 1$	47	103
a^{48}	$a^7 + a^6 + a^3 + a^2 + a$	48	206
a^{49}	$a^4 + a^3 + a + 1$	49	27
a^{50}	$a^5 + a^4 + a^2 + a$	50	54
a^{51}	$a^6 + a^5 + a^3 + a^2$	51	108

α^i	$Q_i = \alpha^i \bmod P$	$[i]$	$(Q_i(2))$
a^{52}	$a^7 + a^6 + a^4 + a^3$	52	216
a^{53}	$a^5 + a^4 + a^2 + a + 1$	53	55
a^{54}	$a^6 + a^5 + a^3 + a^2 + a$	54	110
a^{55}	$a^7 + a^6 + a^4 + a^3 + a^2$	55	220
a^{56}	$a^5 + a^4 + a^3 + a^2 + a + 1$	56	63
a^{57}	$a^6 + a^5 + a^4 + a^3 + a^2 + a$	57	126
a^{58}	$a^7 + a^6 + a^5 + a^4 + a^3 + a^2$	58	252
a^{59}	$a^6 + a^5 + a^4 + a^3 + a^2 + a + 1$	59	127
a^{60}	$a^7 + a^6 + a^5 + a^4 + a^3 + a^2 + a$	60	254
a^{61}	$a^6 + a^5 + a^4 + a^3 + a + 1$	61	123
a^{62}	$a^7 + a^6 + a^5 + a^4 + a^2 + a$	62	246
a^{63}	$a^6 + a^5 + a^3 + a + 1$	63	107
a^{64}	$a^7 + a^6 + a^4 + a^2 + a$	64	214
a^{65}	$a^5 + a^3 + a + 1$	65	43
a^{66}	$a^6 + a^4 + a^2 + a$	66	86
a^{67}	$a^7 + a^5 + a^3 + a^2$	67	172
a^{68}	$a^7 + a^6 + a^4 + a^3 + a^2 + a + 1$	68	223
a^{69}	$a^5 + a^4 + a^3 + 1$	69	57
a^{70}	$a^6 + a^5 + a^4 + a$	70	114
a^{71}	$a^7 + a^6 + a^5 + a^2$	71	228
a^{72}	$a^6 + a^3 + a^2 + a + 1$	72	79
a^{73}	$a^7 + a^4 + a^3 + a^2 + a$	73	158
a^{74}	$a^7 + a^5 + a^4 + a^3 + a + 1$	74	187
a^{75}	$a^7 + a^6 + a^5 + a^4 + 1$	75	241
a^{76}	$a^6 + a^5 + a^2 + 1$	76	101
a^{77}	$a^7 + a^6 + a^3 + a$	77	202
a^{78}	$a^4 + a + 1$	78	19
a^{79}	$a^5 + a^2 + a$	79	38
a^{80}	$a^6 + a^3 + a^2$	80	76
a^{81}	$a^7 + a^4 + a^3$	81	152
a^{82}	$a^7 + a^5 + a^4 + a^2 + a + 1$	82	183
a^{83}	$a^7 + a^6 + a^5 + a^3 + 1$	83	233
a^{84}	$a^6 + a^4 + a^2 + 1$	84	85
a^{85}	$a^7 + a^5 + a^3 + a$	85	170
a^{86}	$a^7 + a^6 + a^4 + a + 1$	86	211
a^{87}	$a^5 + 1$	87	33
a^{88}	$a^6 + a$	88	66
a^{89}	$a^7 + a^2$	89	132
a^{90}	$a^7 + a^3 + a^2 + a + 1$	90	143
a^{91}	$a^7 + a^4 + a^3 + 1$	91	153
a^{92}	$a^7 + a^5 + a^4 + a^2 + 1$	92	181
a^{93}	$a^7 + a^6 + a^5 + a^3 + a^2 + 1$	93	237
a^{94}	$a^6 + a^4 + a^3 + a^2 + 1$	94	93
a^{95}	$a^7 + a^5 + a^4 + a^3 + a$	95	186
a^{96}	$a^7 + a^6 + a^5 + a^4 + a + 1$	96	243
a^{97}	$a^6 + a^5 + 1$	97	97
a^{98}	$a^7 + a^6 + a$	98	194
a^{99}	$a + 1$	99	3
a^{100}	$a^2 + a$	100	6
a^{101}	$a^3 + a^2$	101	12
a^{102}	$a^4 + a^3$	102	24
a^{103}	$a^5 + a^4$	103	48
a^{104}	$a^6 + a^5$	104	96

α^i	$Q_i = \alpha^i \bmod P$	$[i]$	$(Q_i(2))$	α^i	$Q_i = \alpha^i \bmod P$	$[i]$	$(Q_i(2))$
a^{105}	$a^7 + a^6$	105	192	a^{157}	$a^7 + a + 1$	157	131
a^{106}	$a^2 + a + 1$	106	7	a^{158}	$a^7 + 1$	158	129
a^{107}	$a^3 + a^2 + a$	107	14	a^{159}	$a^7 + a^2 + 1$	159	133
a^{108}	$a^4 + a^3 + a^2$	108	28	a^{160}	$a^7 + a^3 + a^2 + 1$	160	141
a^{109}	$a^5 + a^4 + a^3$	109	56	a^{161}	$a^7 + a^4 + a^3 + a^2 + 1$	161	157
a^{110}	$a^6 + a^5 + a^4$	110	112	a^{162}	$a^7 + a^5 + a^4 + a^3 + a^2 + 1$	162	189
a^{111}	$a^7 + a^6 + a^5$	111	224	a^{163}	$a^7 + a^6 + a^5 + a^4 + a^3 + a^2 + 1$	163	253
a^{112}	$a^6 + a^2 + a + 1$	112	71	a^{164}	$a^6 + a^5 + a^4 + a^3 + a^2 + 1$	164	125
a^{113}	$a^7 + a^3 + a^2 + a$	113	142	a^{165}	$a^7 + a^6 + a^5 + a^4 + a^3 + a$	165	250
a^{114}	$a^7 + a^4 + a^3 + a + 1$	114	155	a^{166}	$a^6 + a^5 + a^4 + a + 1$	166	115
a^{115}	$a^7 + a^5 + a^4 + 1$	115	177	a^{167}	$a^7 + a^6 + a^5 + a^2 + a$	167	230
a^{116}	$a^7 + a^6 + a^5 + a^2 + 1$	116	229	a^{168}	$a^6 + a^3 + a + 1$	168	75
a^{117}	$a^6 + a^3 + a^2 + 1$	117	77	a^{169}	$a^7 + a^4 + a^2 + a$	169	150
a^{118}	$a^7 + a^4 + a^3 + a$	118	154	a^{170}	$a^7 + a^5 + a^3 + a + 1$	170	171
a^{119}	$a^7 + a^5 + a^4 + a + 1$	119	179	a^{171}	$a^7 + a^6 + a^4 + 1$	171	209
a^{120}	$a^7 + a^6 + a^5 + 1$	120	225	a^{172}	$a^5 + a^2 + 1$	172	37
a^{121}	$a^6 + a^2 + 1$	121	69	a^{173}	$a^6 + a^3 + a$	173	74
a^{122}	$a^7 + a^3 + a$	122	138	a^{174}	$a^7 + a^4 + a^2$	174	148
a^{123}	$a^7 + a^4 + a + 1$	123	147	a^{175}	$a^7 + a^5 + a^3 + a^2 + a + 1$	175	175
a^{124}	$a^7 + a^5 + 1$	124	161	a^{176}	$a^7 + a^6 + a^4 + a^3 + 1$	176	217
a^{125}	$a^7 + a^6 + a^2 + 1$	125	197	a^{177}	$a^5 + a^4 + a^2 + 1$	177	53
a^{126}	$a^3 + a^2 + 1$	126	13	a^{178}	$a^6 + a^5 + a^3 + a$	178	106
a^{127}	$a^4 + a^3 + a$	127	26	a^{179}	$a^7 + a^6 + a^4 + a^2$	179	212
a^{128}	$a^5 + a^4 + a^2$	128	52	a^{180}	$a^5 + a^3 + a^2 + a + 1$	180	47
a^{129}	$a^6 + a^5 + a^3$	129	104	a^{181}	$a^6 + a^4 + a^3 + a^2 + a$	181	94
a^{130}	$a^7 + a^6 + a^4$	130	208	a^{182}	$a^7 + a^5 + a^4 + a^3 + a^2$	182	188
a^{131}	$a^5 + a^2 + a + 1$	131	39	a^{183}	$a^7 + a^6 + a^5 + a^4 + a^3 + a^2 + a + 1$	183	255
a^{132}	$a^6 + a^3 + a^2 + a$	132	78	a^{184}	$a^6 + a^5 + a^4 + a^3 + 1$	184	121
a^{133}	$a^7 + a^4 + a^3 + a^2$	133	156	a^{185}	$a^7 + a^6 + a^5 + a^4 + a$	185	242
a^{134}	$a^7 + a^5 + a^4 + a^3 + a^2 + a + 1$	134	191	a^{186}	$a^6 + a^5 + a + 1$	186	99
a^{135}	$a^7 + a^6 + a^5 + a^4 + a^3 + 1$	135	249	a^{187}	$a^7 + a^6 + a^2 + a$	187	198
a^{136}	$a^6 + a^5 + a^4 + a^2 + 1$	136	117	a^{188}	$a^3 + a + 1$	188	11
a^{137}	$a^7 + a^6 + a^5 + a^3 + a$	137	234	a^{189}	$a^4 + a^2 + a$	189	22
a^{138}	$a^6 + a^4 + a + 1$	138	83	a^{190}	$a^5 + a^3 + a^2$	190	44
a^{139}	$a^7 + a^5 + a^2 + a$	139	166	a^{191}	$a^6 + a^4 + a^3$	191	88
a^{140}	$a^7 + a^6 + a^3 + a + 1$	140	203	a^{192}	$a^7 + a^5 + a^4$	192	176
a^{141}	$a^4 + 1$	141	17	a^{193}	$a^7 + a^6 + a^5 + a^2 + a + 1$	193	231
a^{142}	$a^5 + a$	142	34	a^{194}	$a^6 + a^3 + 1$	194	73
a^{143}	$a^6 + a^2$	143	68	a^{195}	$a^7 + a^4 + a$	195	146
a^{144}	$a^7 + a^3$	144	136	a^{196}	$a^7 + a^5 + a + 1$	196	163
a^{145}	$a^7 + a^4 + a^2 + a + 1$	145	151	a^{197}	$a^7 + a^6 + 1$	197	193
a^{146}	$a^7 + a^5 + a^3 + 1$	146	169	a^{198}	$a^2 + 1$	198	5
a^{147}	$a^7 + a^6 + a^4 + a^2 + 1$	147	213	a^{199}	$a^3 + a$	199	10
a^{148}	$a^5 + a^3 + a^2 + 1$	148	45	a^{200}	$a^4 + a^2$	200	20
a^{149}	$a^6 + a^4 + a^3 + a$	149	90	a^{201}	$a^5 + a^3$	201	40
a^{150}	$a^7 + a^5 + a^4 + a^2$	150	180	a^{202}	$a^6 + a^4$	202	80
a^{151}	$a^7 + a^6 + a^5 + a^3 + a^2 + a + 1$	151	239	a^{203}	$a^7 + a^5$	203	160
a^{152}	$a^6 + a^4 + a^3 + 1$	152	89	a^{204}	$a^7 + a^6 + a^2 + a + 1$	204	199
a^{153}	$a^7 + a^5 + a^4 + a$	153	178	a^{205}	$a^3 + 1$	205	9
a^{154}	$a^7 + a^6 + a^5 + a + 1$	154	227	a^{206}	$a^4 + a$	206	18
a^{155}	$a^6 + 1$	155	65	a^{207}	$a^5 + a^2$	207	36
a^{156}	$a^7 + a$	156	130	a^{208}	$a^6 + a^3$	208	72

α^i	$Q_i = \alpha^i \bmod P$	$[i]$	$(Q_i(2))$
a^{209}	$a^7 + a^4$	209	144
a^{210}	$a^7 + a^5 + a^2 + a + 1$	210	167
a^{211}	$a^7 + a^6 + a^3 + 1$	211	201
a^{212}	$a^4 + a^2 + 1$	212	21
a^{213}	$a^5 + a^3 + a$	213	42
a^{214}	$a^6 + a^4 + a^2$	214	84
a^{215}	$a^7 + a^5 + a^3$	215	168
a^{216}	$a^7 + a^6 + a^4 + a^2 + a + 1$	216	215
a^{217}	$a^5 + a^3 + 1$	217	41
a^{218}	$a^6 + a^4 + a$	218	82
a^{219}	$a^7 + a^5 + a^2$	219	164
a^{220}	$a^7 + a^6 + a^3 + a^2 + a + 1$	220	207
a^{221}	$a^4 + a^3 + 1$	221	25
a^{222}	$a^5 + a^4 + a$	222	50
a^{223}	$a^6 + a^5 + a^2$	223	100
a^{224}	$a^7 + a^6 + a^3$	224	200
a^{225}	$a^4 + a^2 + a + 1$	225	23
a^{226}	$a^5 + a^3 + a^2 + a$	226	46
a^{227}	$a^6 + a^4 + a^3 + a^2$	227	92
a^{228}	$a^7 + a^5 + a^4 + a^3$	228	184
a^{229}	$a^7 + a^6 + a^5 + a^4 + a^2 + a + 1$	229	247
a^{230}	$a^6 + a^5 + a^3 + 1$	230	105
a^{231}	$a^7 + a^6 + a^4 + a$	231	210
a^{232}	$a^5 + a + 1$	232	35
a^{233}	$a^6 + a^2 + a$	233	70
a^{234}	$a^7 + a^3 + a^2$	234	140
a^{235}	$a^7 + a^4 + a^3 + a^2 + a + 1$	235	159
a^{236}	$a^7 + a^5 + a^4 + a^3 + 1$	236	185
a^{237}	$a^7 + a^6 + a^5 + a^4 + a^2 + 1$	237	245
a^{238}	$a^6 + a^5 + a^3 + a^2 + 1$	238	109
a^{239}	$a^7 + a^6 + a^4 + a^3 + a$	239	218
a^{240}	$a^5 + a^4 + a + 1$	240	51
a^{241}	$a^6 + a^5 + a^2 + a$	241	102
a^{242}	$a^7 + a^6 + a^3 + a^2$	242	204
a^{243}	$a^4 + a^3 + a^2 + a + 1$	243	31
a^{244}	$a^5 + a^4 + a^3 + a^2 + a$	244	62
a^{245}	$a^6 + a^5 + a^4 + a^3 + a^2$	245	124
a^{246}	$a^7 + a^6 + a^5 + a^4 + a^3$	246	248
a^{247}	$a^6 + a^5 + a^4 + a^2 + a + 1$	247	119
a^{248}	$a^7 + a^6 + a^5 + a^3 + a^2 + a$	248	238
a^{249}	$a^6 + a^4 + a^3 + a + 1$	249	91
a^{250}	$a^7 + a^5 + a^4 + a^2 + a$	250	182
a^{251}	$a^7 + a^6 + a^5 + a^3 + a + 1$	251	235
a^{252}	$a^6 + a^4 + 1$	252	81
a^{253}	$a^7 + a^5 + a$	253	162
a^{254}	$a^7 + a^6 + a + 1$	254	195