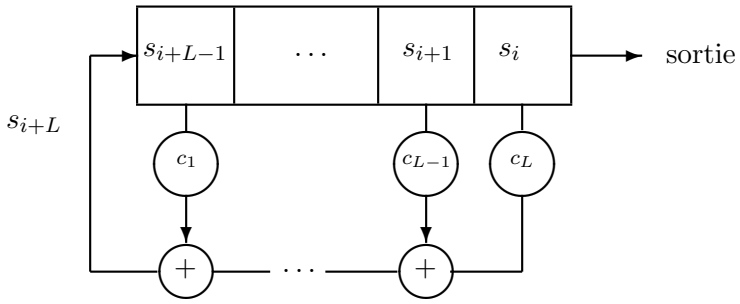
Registres à décalage à rétroaction linéaire et algorithme de Berlekamp-Massey

Une méthode classique pour générer une suite binaire pseudo-aléatoire est d'utiliser un registre à décalage à rétroaction linéaire (LFSR pour Linear Feedback Shift Register). Un LFSR de longueur L est composé d'un registre à décalage contenant une suite de L bits $(s_i, \ldots, s_{i+L-1})$, et d'une fonction de rétroaction linéaire.



A chaque top d'horloge, le bit de poids faible s_i constitue la sortie du registre, et les autres bits sont décalés vers la droite. Le nouveau bit s_{i+L} placé dans la cellule de poids fort du registre est donné par une fonction linéaire des bits $(s_i, \ldots, s_{i+L-1})$

$$s_{i+L} = c_1 s_{i+L-1} + c_2 s_{i+L-2} + \ldots + c_L s_i$$

où les coefficients de rétroaction $(c_i)_{1 \leq i \leq L}$ sont des éléments de $\mathbf{F}_2$.

Les bits $(s_0, \ldots, s_{L-1})$, qui déterminent entièrement la suite, constituent *l'état initial du registre.*

Les coefficients de rétroaction sont usuellement représentés par un polynôme de $\mathbf{F}_2[X]$ de degré L, appelé *polynôme de rétroaction du registre*:

$$P(X) = 1 + c_1 X + c_2 X^2 + \ldots + c_L X^L \ .$$

Un résultat classique est que la période de la suite binaire engendrée par un LFSR est maximale (et vaut $2^L - 1$) quand le polynôme de rétroaction est primitif. On se placera donc dans ce cas dans toute la suite.

On peut donc utiliser un LFSR comme générateur pseudo-aléatoire dans les applications cryptographiques. La clef secrète du système peut alors correspondre à l'initialisation et aux coefficients de rétroaction du registre.

Un tel générateur pseudo-aléatoire s'avère toutefois d'une grande faiblesse cryptographique. En 1969, J. Massey a montré que l'algorithme proposé par Berlekamp pour le décodage des codes BCH pouvait être adapté pour retrouver le polynôme de rétroaction d'un LFSR à partir uniquement des $2L$ premiers bits de la suite produite $\mathbf{s}$.

Cet algorithme consiste à construire pour les valeurs successives de N un LFSR de longueur L_N et de polynôme de rétroaction f_N qui génère les N premiers bits de la suite $\mathbf{s}$. On peut montrer que pour $N = 2L$, l'algorithme retourne le polynôme de rétroaction du LFSR de départ.

ENTRÉE : s_0,$s_1, \ldots, s_{n-1}$ une suite de longueur n.

INITIALISATION

$$f(X) = 1, \quad L = 0, \quad m = -1, \quad g(X) = 1$$

POUR N VARIANT DE 0 À $n-1$

1. Calculer $d = s_N + \sum_{i=1}^{L} c_i s_{N-i} \bmod 2$.
2. Si $d = 1$ alors

   – $t(X) = f(X)$ et $f(X) = f(X) + g(X)X^{N-m}$.
   – Si $2L \leq N$ alors $L = N + 1 - L$, $m = N$, $g(X) = t(X)$.

Exemple : Application de l'algorithme de Berlekamp-Massey à la suite binaire de longueur 9, $\mathbf{s} = 001101110$.

N	s_N	d	L	$f(X)$	m	$g(X)$
			0	1	-1	1
0	0	0	0	1	-1	1
1	0	0	0	1	-1	1
2	1	1	3	$X^3 + 1$	2	1
3	1	1	3	$X^3 + X + 1$	2	1
4	0	1	3	$X^3 + X^2 + X + 1$	2	1
5	1	1	3	$X^2 + X + 1$	2	1
6	1	0	3	$X^2 + X + 1$	2	1
7	1	1	5	$X^5 + X^2 + X + 1$	7	$X^2 + X + 1$
8	0	1	5	$X^5 + X^3 + 1$	7	$X^2 + X + 1$

Le LFSR de longueur 5 ayant pour polynôme de rétroaction $X^5 + X^3 + 1$ génère donc la suite donnée.

Pour plus d'information sur cet algorithme, on pourra se référer à :
MASSEY (J.L.). – Shift-register synthesis and BCH decoding. *IEEE Transactions on Information Theory*, vol. 15, janvier 1969, pp. 122–127.

1 Implémentation d'un LFSR

Le but de cette première partie est d'écrire un programme qui prend en argument un entier N et le nom de fichier dans lequel est décrit le polynôme de rétroaction du LFSR, et qui écrit dans un fichier les N premiers bits de la suite produite par ce LFSR avec une initialisation aléatoire.

Les spécifications du registre seront données dans un fichier contenant la liste des degrés des monômes intervenant dans le polynôme de rétroaction P, *classés par ordre décroissant*. Ainsi, le polynôme $1 + X^3 + X^{40}$ sera représenté par la ligne

```
40 3 0
```

1.1 Lecture du polynôme de rétroaction

Dans tout le programme, on représentera les polynômes à coefficients binaires et l'état des registres par des tableaux d'entiers de type `unsigned long`. Par exemple, le polynôme $1 + X^2 + X^3 + X^5 + X^6$ sera représenté par un tableau dont l'unique élément est l'entier $1 + 2^2 + 2^3 + 2^5 + 2^6 = 109$.

Pour faciliter le calcul du bit de rétroaction, on fera en sorte que les tableaux d'entiers représentant respectivement l'état du registre et les coefficients de rétroaction correspondent aux bits ordonnés de la façon suivante :

- $(s_{i+L-1}, \ldots, s_{i+1}, s_i)$ pour le tableau donnant l'état ;
- $(c_1, \ldots, c_{L-1}, c_L)$ pour le tableau représentant les coefficients de rétroaction.

De cette façon, le calcul du bit de rétroaction va pouvoir se faire à l'aide d'un ET entre les 2 tableaux.

Écrire une fonction qui, pour une variable `L` de type `unsigned int` et un flot (type `FILE*`) donnés, affecte à `L` le degré du polynôme lu dans le flot et retourne un tableau d'`unsigned long` correspondant aux coefficients $(c_1, \ldots, c_{L-1}, c_L)$.

1.2 Initialisation d'un LFSR par une valeur aléatoire

Écrire une fonction qui génère aléatoirement l'initialisation d'un LFSR de longueur L sous forme d'un tableau d'`unsigned long`. Son en-tête sera

```
unsigned long *lfsr_random_init(unsigned int L)
```

1.3 Implémentation d'un LFSR

Écrire une fonction `lfsr_clock` qui correspond à un top d'horloge du LFSR. Cette fonction a pour en-tête

```
unsigned int lfsr_clock(unsigned long *lfsr, unsigned long *Q,
unsigned int L)
```

Le paramètre `lfsr` donne le contenu du registre à un instant donné, `Q` est le polynôme représentant les coefficients de rétroaction et `L` la longueur du registre. La fonction `lfsr_clock` retournera le bit produit en sortie du LFSR et aura pour effet de mettre à jour le contenu du registre.

1.4 Programme principal

Écrire un programme qui, à partir de la description du système donné dans un fichier, initialise le LFSR par une valeur aléatoire et affiche les N premiers bits de sa sortie. Le nom du fichier contenant la description du système et la valeur de N seront passés en arguments du programme. On entrera par exemple la ligne de commande

```
generateur description.txt 10000
```

2 Algorithme de Berlekamp-Massey

L'objectif de cette deuxième partie est d'écrire un programme qui prend comme argument un fichier contenant une suite binaire et un entier L correspondant à la longueur maximale

attendue pour le LFSR, et qui applique l'algorithme de Berlekamp-Massey aux $2L$ premiers bits de la suite.

On pourra écrire une fonction qui prend comme paramètre un tableau d'`unsigned long` représentant un polynôme et qui affiche à l'écran ce polynôme sous forme habituelle. Le prototype de cette fonction sera

```
void affiche_polynome(unsigned long *);
```