Guide for Reed Solomon Codes

July 15, 2021

1 Galois Fields

A **field** is a set with two special elements (0 and 1) where we can add, subtract, multiply and divide any two elements, except for division by zero.

Example : $\mathbb{R}$ (the real numbers) and $\mathbb{C}$ (the complex numbers) are fields.

The previous examples have infinitely many elements. There are also fields with finitely many elements; they are called **Galois fields** or **finite fields**.
Given a Galois field, there exists a prime number p such that for any element a of the field holds

$$\underbrace{a + \cdots + a}_{p \text{ times}} = 0.$$

This prime p is uniquely determined and called the **characteristic** of the field. Moreover, there exists a positive integer n such that the field has p^n elements. This motivates us to denote the Galois field as $GF(p^n)$.

Example : $GF(256)$ (also written as $GF(2^8)$) is a Galois field with 256 elements.

1.1 Understanding $GF(256)$

1.1.1 Decimal, Binary and Polynomial Representation

The elements in $GF(256)$ can be expressed as numbers in **decimal representation**, so that $GF(256)$ contains the elements $0, 1, 2, \ldots, 254, 255$. However, this representation is not helpful to make operations by hand; for that it is more useful to use the binary representation.
Any number a from 0 to 255 can be uniquely written as a linear combination of $\{2^0, 2^1, 2^2, 2^3, 2^4, 2^5, 2^6, 2^7\}$ with coefficients a_i in $\{0, 1\}$. That is,

$$a = a_7 2^7 + a_6 2^6 + a_5 2^5 + a_4 2^4 + a_3 2^3 + a_2 2^2 + a_1 2^1 + a_0 2^0. \tag{1}$$

The **binary representation** of the decimal a is the vector $(a_7, a_6, a_5, a_4, a_3, a_2, a_1, a_0)$.

Note: We will frequently specify a particular coordinate as the 0th entry, 1st entry, etc. according to the following diagram.

(7th, 6th, 5th, 4th, 3rd, 2nd, 1st, 0th)

Closely related to the **binary representation** is the **polynomial representation** of a as a polynomial with binary coefficients,

$$a(x) = a_7x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x^1 + a_0.$$

Example 1: The element 47 can be written as

$$47 = 0 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0.$$

The binary representation of 47 is $(0, 0, 1, 0, 1, 1, 1, 1)$.

Example 2: The element 220 can be written as

$$220 = 1 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0.$$

The binary representation of 220 is $(1, 1, 0, 1, 1, 1, 0, 0)$.

Example 3: The element 183 can be written as

$$183 = 1 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0.$$

The binary representation of 183 is $(1, 0, 1, 1, 0, 1, 1, 1)$.

Conversely, if we have an element written in binary representation $(a_7, a_6, a_5, a_4, a_3, a_2, a_1, a_0)$, to find the decimal representation we must evaluate the sum (over the integers) on the right-hand side of equation (1).

Example 1: The binary representation $(1, 0, 0, 1, 1, 0, 1, 0)$ corresponds to the decimal 154, since:

$$1 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 128 + 16 + 8 + 2 = 154.$$

1.1.2 Addition

The addition in the field $GF(256)$ is not the same as in the natural numbers; for instance $47 + 183 \neq 230$ when seen as elements of $GF(256)$.
If we have elements a and b in $GF(256)$, to obtain $a + b$ we must first find their binary representations and then add by coordinate; this is, if the binary representations are $(a_7, a_6, a_5, a_4, a_3, a_2, a_1, a_0)$ and $(b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0)$, then $a + b$ is the element having binary representation

$$(a_7 + b_7, a_6 + b_6, a_5 + b_5, a_4 + b_4, a_3 + b_3, a_2 + b_2, a_1 + b_1, a_0 + b_0).$$

The addition by coordinate can be done in long addition format and obeys the additive XOR rules:

$$0 + 0 = 0, \quad 0 + 1 = 1, \quad 1 + 0 = 1 \quad \text{and} \quad 1 + 1 = 0.$$

Example 1: $1 + 1 = 0$ in $GF(256)$.
The binary representation of 1 is $(0, 0, 0, 0, 0, 0, 0, 1)$.
The binary representation of $1 + 1$ is:

$$(0 + 0, 0 + 0, 0 + 0, 0 + 0, 0 + 0, 0 + 0, 0 + 0, 1 + 1) = (0, 0, 0, 0, 0, 0, 0, 0)$$

This calculation can also be visualized as a long addition:

$$\begin{array}{cccccccccc} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ + & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

Caution: This is not the usual binary addition with carry over! **Remember: Addition = XOR!**

Now convert from binary back to decimal representation:

$$0 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 0.$$

Thus, $1 + 1 = 0$ in $GF(256)$.

Example 2: $2 + 1 = 3$ in $GF(256)$.
The binary representations of 2 and 1 are $(0, 0, 0, 0, 0, 0, 1, 0)$ and $(0, 0, 0, 0, 0, 0, 0, 1)$, respectively.
The long addition of $2 + 1$ is:

$$\begin{array}{cccccccccc} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ + & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{array}$$

Now convert from binary back to decimal representation:

$$0 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 2 + 1 = 3.$$

Thus, $2 + 1 = 3$ in $GF(256)$.

Example 3: $4 + 1 = 5$ in $GF(256)$.
The binary representations of 4 and 1 are $(0, 0, 0, 0, 0, 1, 0, 0)$ and $(0, 0, 0, 0, 0, 0, 0, 1)$, respectively.
The long addition of $4 + 1$ is:

$$\begin{array}{cccccccccc} & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ + & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{array}$$

Now convert from binary back to decimal representation:

$$0 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 4 + 1 = 5.$$

Thus, $4 + 1 = 5$ in $GF(256)$.

Example 4: $8 + 1 = 9$ in $GF(256)$.
The binary representations of 8 and 1 are $(0, 0, 0, 0, 1, 0, 0, 0)$ and $(0, 0, 0, 0, 0, 0, 0, 1)$, respectively.
The long addition of $8 + 1$ is:

$$\begin{array}{cccccccccc} & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ + & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{array}$$

Now convert from binary back to decimal representation:

$$0 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 8 + 1 = 9.$$

Thus, $8 + 1 = 9$ in $GF(256)$.

Example 5: $47 + 183 = 152$ in $GF(256)$.
The binary representations of 47 and 183 are $(0, 0, 1, 0, 1, 1, 1, 1)$ and $(1, 0, 1, 1, 0, 1, 1, 1)$, respectively.
The long addition of $47 + 183$ is:

$$\begin{array}{cccccccccc} & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ + & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ \hline & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \end{array}$$

Now convert from binary back to decimal representation:

$$1 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 128 + 16 + 8 = 152.$$

Thus, $47 + 183 = 152$ in $GF(256)$.

Example 6: $231 + 183 = 80$ in $GF(256)$.
The binary representations of 231 and 183 are $(1,1,1,0,0,1,1,1)$ and $(1,0,1,1,0,1,1,1)$, respectively.
The long addition of $231 + 183$ is:

$$\begin{array}{cccccccccc} & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ + & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ \hline & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \end{array}$$

Now convert from binary back to decimal representation:

$$0 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 64 + 16 = 80.$$

Thus, $231 + 183 = 80$ in $GF(256)$.

1.1.3 Subtraction

Subtraction in $GF(256)$ is the same as addition, i.e., for any a and b in $GF(256)$ holds $a - b = a + b$.

1.1.4 Multiplication

The multiplication in $GF(256)$ has primitive polynomial 0x11D; this means, it can be seen as polynomial multiplication in $\mathbb{Z}_2[x]$ modulo $p(x) = x^8 + x^4 + x^3 + x^2 + 1$.

The multiplication being modulo $p(x)$ indicates how to multiply an element a in $GF(256)$ by 2. If a has binary representation $(a_7, a_6, a_5, a_4, a_3, a_2, a_1, a_0)$, then the coordinates of the binary representation of the product $a \cdot 2$ are:

$$\begin{cases} (a \cdot 2)_7 &= a_6 \\ (a \cdot 2)_6 &= a_5 \\ (a \cdot 2)_5 &= a_4 \\ (a \cdot 2)_4 &= a_3 + a_7 \\ (a \cdot 2)_3 &= a_2 + a_7 \\ (a \cdot 2)_2 &= a_1 + a_7 \\ (a \cdot 2)_1 &= a_0 \\ (a \cdot 2)_0 &= a_7 \end{cases} \tag{2}$$

Example 1: $171 \cdot 2 = 75$ in $GF(256)$.
The binary representation of 171 is $(1,0,1,0,1,0,1,1)$. By rule (2), the binary representation of $171 \cdot 2$ is:

$$(0, 1, 0, 1+1, 0+1, 1+1, 1, 1) = (0,1,0,0,1,0,1,1)$$

The decimal representation of the product is $2^6 + 2^3 + 2 + 1 = 64 + 8 + 2 + 1 = 75$.

Now, to multiply any two elements of $GF(256)$, decompose one of the elements as a sum of powers of two, then distribute this sum with respect to the product, and use (2) to multiply by 2 as many times as needed.

Example 2: $3 \cdot 3 = 5$ in $GF(256)$.
The binary representation of 3 is $(0,0,0,0,0,0,1,1)$ and $3 = 2 + 1$. Then,

$$\begin{aligned} 3 \cdot 3 &= 3 \cdot (2+1) \\ &= 3 \cdot 2 + 3 \cdot 1 && \text{by distributivity} \\ &= (0,0,0,0,0,0,1,1) \cdot 2 && 3 \cdot 2 \end{aligned}$$

$+ (0,0,0,0,0,0,1,1) \cdot 1$ $3 \cdot 1$

$= (0,0,0,0,0,1,1,0)$ the 7th entry of $(0,0,0,0,0,0,1,1)$ is zero, so $\cdot 2$ just shifts every entry to the left

$+ (0,0,0,0,0,0,1,1)$

$= (0,0,0,0,0,1,0,1)$ which in decimal form equals $2^2 + 1 = 5$.

Note: The decomposition into powers of 2 in $GF(256)$ coincides with the decomposition into powers of 2 in the integers. For instance, $171 = 2^7 + 2^5 + 2^3 + 2 + 1$ holds in $GF(256)$ as well as over the natural numbers.

Example 3: $171 \cdot 7 = 118$ in $GF(256)$.
The binary representation of 171 is $(1,0,1,0,1,0,1,1)$ and $7 = 4 + 2 + 1 = 2^2 + 2 + 1$. Then,

$171 \cdot 7 = 171 \cdot (2^2 + 2 + 1)$

$= 171 \cdot 2 \cdot 2 + 171 \cdot 2 + 171 \cdot 1$ by distributivity

$= (1,0,1,0,1,0,1,1) \cdot 2 \cdot 2$ $171 \cdot 2 \cdot 2$

$+ (1,0,1,0,1,0,1,1) \cdot 2$ $171 \cdot 2$

$+ (1,0,1,0,1,0,1,1) \cdot 1$ $171 \cdot 1$

$= (0,1,0,0,1,0,1,1) \cdot 2$ shift to the left and add 1 to the 2nd, 3rd and 4th entry

$+ (0,1,0,0,1,0,1,1)$ shift to the left and add 1 to the 2nd, 3rd and 4th entry

$+ (1,0,1,0,1,0,1,1)$

$= (1,0,0,1,0,1,1,0)$ the 7th entry of $(0,1,0,0,1,0,1,1)$ is zero, so $\cdot 2$ just shifts every entry to the left

$+ (0,1,0,0,1,0,1,1)$

$+ (1,0,1,0,1,0,1,1)$

$= (0,1,1,1,0,1,1,0)$ which in decimal form equals $2^6 + 2^5 + 2^4 + 2^2 + 2 = 118$.

Example 4: $108 \cdot 32 = 1$ in $GF(256)$.
The binary representation of 108 is $(0,1,1,0,1,1,0,0)$ and $32 = 2^5$. Then,

$108 \cdot 32 = 108 \cdot 2^5$

$= (0,1,1,0,1,1,0,0) \cdot 2^5$

$= (1,1,0,1,1,0,0,0) \cdot 2^4$ the 7th entry of $(0,1,1,0,1,1,0,0)$ is zero, so $\cdot 2$ just shifts every entry to the left

$= (1,0,1,0,1,1,0,1) \cdot 2^3$ shift to the left and add 1 to the 2nd, 3rd and 4th entry

$= (0,1,0,0,0,1,1,1) \cdot 2^2$ shift to the left and add 1 to the 2nd, 3rd and 4th entry

$= (1,0,0,0,1,1,1,0) \cdot 2$ shift every entry to the left

$= (0,0,0,0,0,0,0,1),$ shift to the left and add 1 to the 2nd, 3rd and 4th entry

which in decimal form equals 1.

1.1.5 Division

We can reduce the problem of division in $GF(256)$ to the problem of multiplication, since $a/b = a \cdot b^{-1}$.
For b^{-1} we may simply use a table of multiplicative inverses for $GF(256)$.

b	b^{-1}
1	1
2	142
3	244
4	71
5	167
6	122
7	186
8	173
9	157
10	221
11	152
12	61
13	170
14	93
15	150
16	216
17	114
18	192
19	88
20	224
21	62
22	76
23	102
24	144
25	222
26	85
27	128
28	160
29	131
30	75
31	42
32	108
33	237
34	57
35	81
36	96
37	86
38	44
39	138
40	112
41	208
42	31
43	74
44	38
45	139
46	51
47	110
48	72
49	137
50	111
51	46

b	b^{-1}
52	164
53	195
54	64
55	94
56	80
57	34
58	207
59	169
60	171
61	12
62	21
63	225
64	54
65	95
66	248
67	213
68	146
69	78
70	166
71	4
72	48
73	136
74	43
75	30
76	22
77	103
78	69
79	147
80	56
81	35
82	104
83	140
84	129
85	26
86	37
87	97
88	19
89	193
90	203
91	99
92	151
93	14
94	55
95	65
96	36
97	87
98	202
99	91
100	185
101	196
102	23

b	b^{-1}
103	77
104	82
105	141
106	239
107	179
108	32
109	236
110	47
111	50
112	40
113	209
114	17
115	217
116	233
117	251
118	218
119	121
120	219
121	119
122	6
123	187
124	132
125	205
126	254
127	252
128	27
129	84
130	161
131	29
132	124
133	204
134	228
135	176
136	73
137	49
138	39
139	45
140	83
141	105
142	2
143	245
144	24
145	223
146	68
147	79
148	155
149	188
150	15
151	92
152	11
153	220

b	b^{-1}
154	189
155	148
156	172
157	9
158	199
159	162
160	28
161	130
162	159
163	198
164	52
165	194
166	70
167	5
168	206
169	59
170	13
171	60
172	156
173	8
174	190
175	183
176	135
177	229
178	238
179	107
180	235
181	242
182	191
183	175
184	197
185	100
186	7
187	123
188	149
189	154
190	174
191	182
192	18
193	89
194	165
195	53
196	101
197	184
198	163
199	158
200	210
201	247
202	98
203	90
204	133

b	b^{-1}
205	125
206	168
207	58
208	41
209	113
210	200
211	246
212	249
213	67
214	215
215	214
216	16
217	115
218	118
219	120
220	153
221	10
222	25
223	145
224	20
225	63
226	230
227	240
228	134
229	177
230	226
231	241
232	250
233	116
234	243
235	180
236	109
237	33
238	178
239	106
240	227
241	231
242	181
243	234
244	3
245	143
246	211
247	201
248	66
249	212
250	232
251	117
252	127
253	255
254	126
255	253

2 Encoding of Reed-Solomon Codes

Imagine that we want to send a message consisting of the vector $(m_{k-1}, m_{k-2}, \ldots, m_1, m_0)$ with m_{k-1}, $m_{k-2}, \ldots, m_1, m_0$ from $GF(256)$ using the structure of a Reed-Solomon code $RS(n, k)$ with generator polynomial $g(x)$ of degree $n - k$. This code will encode the message $(m_{k-1}, m_{k-2}, \ldots, m_1, m_0)$ as a codeword $(c_{n-1}, c_{n-2}, \ldots, c_1, c_0)$ with $c_{n-1}, c_{n-2}, \ldots, c_1, c_0$ from $GF(256)$ such that the original message can be recovered as long as there are not more than $\lfloor \frac{n-k}{2} \rfloor$ errors during transmission. In other words, the message $(m_{k-1}, m_{k-2}, \ldots, m_1, m_0)$ can be recovered as long as at least $n - \lfloor \frac{n-k}{2} \rfloor = k + \lceil \frac{n-k}{2} \rceil$ of the values of the codeword $(c_{n-1}, c_{n-2}, \ldots, c_1, c_0)$ are transmitted correctly.

For encoding, follow the following steps.

1. Create the polynomial of degree $\leq k-1$ with the message coordinates m_i as coefficients:

$$m(x) = m_{k-1}x^{k-1} + m_{k-2}x^{k-2} + \ldots + m_1x + m_0$$

2. Multiply $m(x)$ with the generator polynomial $g(x)$ to obtain the polynomial $c(x) = m(x) \cdot g(x)$ of degree $\leq n-1$.

3. Write

$$c(x) = c_{n-1}x^{n-1} + c_{n-2}x^{n-2} + \ldots + c_1x + c_0$$

to read off the coordinates c_i of the codeword.

Example 1: We want to encode the message $(0,0,0,0)$ using the $RS(8,4)$ code with generating polynomial

$$g(x) = (x+1)(x+2)(x+4)(x+8) = x^4 + 15x^3 + 54x^2 + 120x + 64.$$

The polynomial that has the message coordinates as coefficients is

$$m(x) = 0x^3 + 0x^2 + 0x + 0 = 0.$$

We evaluate

$$c(x) = m(x) \cdot g(x) = 0 \cdot \left(x^4 + 15x^3 + 54x^2 + 120x + 64\right) = 0.$$

This results in the codeword $(0,0,0,0,0,0,0,0)$.

Example 2: We want to encode the message $(0,0,0,1)$ using the $RS(8,4)$ code with generating polynomial

$$g(x) = (x+1)(x+2)(x+4)(x+8) = x^4 + 15x^3 + 54x^2 + 120x + 64.$$

The polynomial that has the message coordinates as coefficients is

$$m(x) = 0x^3 + 0x^2 + 0x + 1 = 1.$$

We evaluate

$$c(x) = m(x) \cdot g(x) = 1 \cdot \left(x^4 + 15x^3 + 54x^2 + 120x + 64\right) = x^4 + 15x^3 + 54x^2 + 120x + 64.$$

This results in the codeword $(0,0,0,1,15,54,120,64)$.

Example 3: We want to encode the message $(1,2,4,8)$ using the $RS(8,4)$ code with generating polynomial

$$g(x) = (x+1)(x+2)(x+4)(x+8) = x^4 + 15x^3 + 54x^2 + 120x + 64.$$

The polynomial that has the message coordinates as coefficients is

$$m(x) = x^3 + 2x^2 + 4x + 8.$$

We evaluate

$$\begin{aligned} c(x) = m(x) \cdot g(x) &= \left(x^3 + 2x^2 + 4x + 8\right) \cdot \left(x^4 + 15x^3 + 54x^2 + 120x + 64\right) \\ &= x^7 + (1 \cdot 15 + 2 \cdot 1)x^6 + (1 \cdot 54 + 2 \cdot 15 + 4 \cdot 1)x^5 + (1 \cdot 120 + 2 \cdot 54 + 4 \cdot 15 + 8 \cdot 1)x^4 \\ &+ (1 \cdot 64 + 2 \cdot 120 + 4 \cdot 54 + 8 \cdot 15)x^3 + (2 \cdot 64 + 4 \cdot 120 + 8 \cdot 54)x^2 + (4 \cdot 64 + 8 \cdot 120)x + 8 \cdot 64 \\ &= x^7 + 13x^6 + 44x^5 + 32x^4 + 16x^3 + 208x^2 + 250x + 58. \end{aligned}$$

This results in the codeword $(1,13,44,32,16,208,250,58)$.

3 Decoding

3.1 Peterson-Gorenstein-Zierler Algorithm

We want to decode the received n-dimensional vector r. This vector can be written as $r = c + e$, where c is a codeword and e is an error vector. The *syndrome vector* $s = (s_1, s_2, \ldots, s_{2t})$ has $2t = n - k$ coordinates. Identifying the vector $r = (r_{n-1}, \ldots, r_2, r_1, r_0)$ with the polynomial $r(x) = r_{n-1}x^{n-1} + \ldots + r_2x^2 + r_1x + r_0$, considering an $RS(n,k)$ code where the roots of the generator polynomial $g(x)$ are the consecutive $2t$ elements $1, 2, \ldots, 2^{2t-1}$, we have $s = (s_1, s_2, \ldots, s_{2t})$ with

$$s_i = r(2^{i-1}) = c(2^{i-1}) + e(2^{i-1})$$

and $c(2^{i-1}) = 0$. Thus, syndromes can be expressed as

$$s_i = e(2^{i-1}) = \sum_{j=0}^{n-1} e_j (2^{i-1})^j.$$

Errors occur where $e_j \neq 0$. Denote the position of the v errors with $v \leq t$ as $0 \leq k_1, \ldots, k_v \leq n-1$, and rewrite the syndrome as

$$s_i = \sum_{j=1}^{v} e_{k_j} (2^{i-1})^{k_j} = \sum_{j=1}^{v} e_{k_j} (2^{k_j})^{i-1} = \sum_{j=1}^{v} Y_j (X_j)^{i-1}$$

where $X_j = 2^{k_j}$ and $Y_j = e_{k_j}$. In matrix form, the equation above provides the system

$$\begin{pmatrix} X_1^0 & X_2^0 & \cdots & X_v^0 \\ X_1^1 & X_2^1 & \cdots & X_v^1 \\ \vdots & \vdots & \ddots & \vdots \\ X_1^{v-1} & X_2^{v-1} & \cdots & X_v^{v-1} \end{pmatrix} \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_v \end{pmatrix} = \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_v \end{pmatrix}. \tag{3}$$

The associated *error locator polynomial* is

$$\Lambda(x) = \prod_{j=1}^{v} (1 - X_j x) = \Lambda_v x^v + \Lambda_{v-1} x^{v-1} + \ldots + \Lambda_1 x + \Lambda_0.$$

In fact, $\Lambda_0 = 1$. Then, the k-th syndrome satisfies

$$k\Lambda_k + \Lambda_{k-1} s_1 + \Lambda_{k-2} s_2 + \ldots + \Lambda_1 s_{k-1} = -s_k \quad \text{for } 1 \leq k \leq v$$

and

$$\Lambda_v s_{k-v} + \Lambda_{v-1} s_{k-v+1} + \ldots + \Lambda_1 s_{k-1} = -s_k \quad \text{for } v < k \leq 2t.$$

The latter equation gives the following system:

$$M^{(v)} \Lambda^{(v)} = \begin{pmatrix} s_1 & s_2 & \cdots & s_v \\ s_2 & s_3 & \cdots & s_{v+1} \\ \vdots & \vdots & \ddots & \vdots \\ s_v & s_{v+1} & \cdots & s_{2v-1} \end{pmatrix} \begin{pmatrix} \Lambda_v \\ \Lambda_{v-1} \\ \vdots \\ \Lambda_1 \end{pmatrix} = \begin{pmatrix} -s_{v+1} \\ -s_{v+2} \\ \vdots \\ -s_{2v} \end{pmatrix} \tag{4}$$

If $\det(M^{(v)}) \neq 0$ then there exists a unique solution vector $\Lambda^{(v)}$ for the system (4) which is given by

$$\Lambda^{(v)} = (M^{(v)})^{-1} \begin{pmatrix} -s_{v+1} \\ \vdots \\ -s_{2v} \end{pmatrix}$$

This solution can also be found using row Gaussian elimination.

If we know the $2t$ syndrome values $s_1, \ldots, s_{2t}$ and not all of them are zero, we can use the Peterson-Gorenstein-Zierler algorithm to detect and correct the errors. The algorithm goes as follows:

1. Assume $v = t$,
2. Check if $M^{(v)}$ is singular.
   - If $\det(M^{(v)}) = 0$, decrement $v \leftarrow v - 1$. If $v = 0$, finish and signal that errors have occurred which are not correctable; else return to Step 2.
   - If $\det(M^{(v)}) \neq 0$, then continue. This v value tells us how many errors occurred.
3. Solve the system (4). The result gives the error location polynomial $\Lambda(x)$.
4. Find the roots of the error location polynomial by exhaustive search. Note that $\Lambda(x) = \prod_{j=1}^{v}(1 - X_j x)$, i.e., $\Lambda(x)$ must factor into linear factors with its v distinct roots from the list $1, 2^{-1}, 2^{-2}, \ldots, 2^{1-n}$. Should such a factorization fail, finish and signal that errors have occurred which are not correctable.
5. Find $X_1, \ldots, X_v$, the reciprocals of the roots found in step 4.
6. Find the error positions $k_i = \log_2(2^{k_i}) = \log_2(X_i)$.
7. Find the error values $Y_j = e_{k_j}$ by solving the linear system (3).
8. Get the error polynomial $e(x) = \sum_{j=1}^{v} e_{k_j} x^{k_j}$.
9. Find the codeword polynomial $c(x) = r(x) - e(x)$.

Advice: Under the assumption of low probability of failure during the transmission of the message, make sure to first verify if all the syndromes are zero, since this will determine if any error has occured. Do not invert any matrix until you have found the correct number of errors.

Now we will work some examples for this decoder.

3.1.1 Example 1

Consider that we have sent the codeword $(0,0,0,0,0,0,0,0)$, but it was transmitted with the error vector $(0,0,0,0,0,0,1,1)$, so that $(0,0,0,0,0,0,1,1)$ was received.
The polynomial associated to the received message is

$$s(x) = x + 1.$$

We start with $v = 2$. The syndromes are

$$\begin{aligned} s_1 &= s(2^0) = s(1) = 1 + 1 = 0, \\ s_2 &= s(2^1) = s(2) = 2 + 1 = 3, \\ s_3 &= s(2^2) = s(4) = 4 + 1 = 5 \quad \text{and} \\ s_4 &= s(2^3) = s(8) = 8 + 1 = 9. \end{aligned}$$

We want to solve the system (4)

$$\begin{pmatrix} 0 & 3 \\ 3 & 5 \end{pmatrix} \begin{pmatrix} \Lambda_2 \\ \Lambda_1 \end{pmatrix} = \begin{pmatrix} 5 \\ 9 \end{pmatrix}.$$

Note that

$$\det \begin{pmatrix} 0 & 3 \\ 3 & 5 \end{pmatrix} = 0 \cdot 5 - 3 \cdot 3 = 5 \neq 0.$$

So the system must have a unique solution. To find it, modify the augmented matrix using Gaussian reduction algorithm;

$$\left(\begin{array}{cc|c} 0 & 3 & 5 \\ 3 & 5 & 9 \end{array}\right) \Longrightarrow \left(\begin{array}{cc|c} 3 & 5 & 9 \\ 0 & 3 & 5 \end{array}\right), \text{ swap the rows since the first element of the diagonal is zero}$$

$$\Longrightarrow \left(\begin{array}{cc|c} 1 & 3 & 7 \\ 0 & 1 & 3 \end{array}\right), \text{ divide all equations by 3 to obtain ones on the diagonal}$$

$$\Longrightarrow \left(\begin{array}{cc|c} 1 & 0 & 2 \\ 0 & 1 & 3 \end{array}\right), \text{ add 3 times the second row to the first row to obtain the identity matrix}$$

$$\Longrightarrow \begin{cases} \Lambda_1 = 3 \\ \Lambda_2 = 2 \end{cases}$$

The error locator polynomial is

$$\Lambda(x) = 2x^2 + 3x + 1 = (1 + x)(1 + 2x) = (1 + 2^0 x)(1 + 2^1 x).$$

Thus, the errors are located at the positions $k_1 = \log_2(1) = \log_2(2^0) = 0$ and $k_2 = \log_2(2) = \log_2(2^1) = 1$.
To find the error values, solve the system (3)

$$\begin{pmatrix} 1^0 & 2^0 \\ 1^1 & 2^1 \end{pmatrix} \begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 3 \end{pmatrix}.$$

To solve it, modify the augmented matrix using Gaussian reduction algorithm;

$$\left(\begin{array}{cc|c} 1 & 1 & 0 \\ 1 & 2 & 3 \end{array}\right) \Longrightarrow \left(\begin{array}{cc|c} 1 & 1 & 0 \\ 0 & 3 & 3 \end{array}\right), \text{ add the first row to the second row to get zero below the diagonal}$$

$$\Longrightarrow \left(\begin{array}{cc|c} 1 & 1 & 0 \\ 0 & 1 & 1 \end{array}\right), \text{ divide the second row by 3 to obtain ones on the diagonal}$$

$$\Longrightarrow \left(\begin{array}{cc|c} 1 & 0 & 1 \\ 0 & 1 & 1 \end{array}\right), \text{ add the second row to the first row to obtain the identity matrix}$$

$$\Longrightarrow \begin{cases} Y_1 = 1 \\ Y_2 = 1 \end{cases}$$

Thus, the error polynomial is $e(x) = x + 1$ and the decoding algorithm gives us the polynomial

$$c(x) = s(x) + e(x) = (x + 1) + (x + 1) = 0.$$

3.1.2 Example 2

Consider that we have sent the codeword $(0, 0, 0, 0, 0, 0, 0, 0)$, but it was transmitted with the error vector $(1, 1, 0, 0, 0, 0, 0, 0)$, so that $(1, 1, 0, 0, 0, 0, 0, 0)$ was received.
The polynomial associated to the received message is

$$s(x) = x^7 + x^6.$$

We start with $v = 2$. The syndromes are

$$\begin{aligned} s_1 &= s(2^0) = s(1) = 1 + 1 = 0, \\ s_2 &= s(2^1) = s(2) = 128 + 64 = 192, \\ s_3 &= s(2^2) = s(4) = 19 + 205 = 222 \quad \text{and} \\ s_4 &= s(2^3) = s(8) = 117 + 45 = 88. \end{aligned}$$

We want to solve the system (4)

$$\begin{pmatrix} 0 & 192 \\ 192 & 222 \end{pmatrix}\begin{pmatrix} \Lambda_2 \\ \Lambda_1 \end{pmatrix} = \begin{pmatrix} 222 \\ 88 \end{pmatrix}.$$

Note that

$$\det\begin{pmatrix} 0 & 192 \\ 192 & 222 \end{pmatrix} = 0 \cdot 222 - 192 \cdot 192 = 222 \neq 0.$$

So the system must have a unique solution. To find it, modify the augmented matrix using Gaussian reduction algorithm;

$$\begin{pmatrix} 0 & 192 & 222 \\ 192 & 222 & 88 \end{pmatrix} \Longrightarrow \begin{pmatrix} 192 & 222 & 88 \\ 0 & 192 & 222 \end{pmatrix}, \text{ swap the rows since the first element of the diagonal is zero}$$

$$\Longrightarrow \begin{pmatrix} 1 & 192 & 89 \\ 0 & 1 & 192 \end{pmatrix}, \text{ divide all equations by 192 to obtain ones on the diagonal}$$

$$\Longrightarrow \begin{pmatrix} 1 & 0 & 135 \\ 0 & 1 & 192 \end{pmatrix}, \text{ add 192 times the second row to the first row to get the identity matrix}$$

$$\Longrightarrow \begin{cases} \Lambda_1 = 192 \\ \Lambda_2 = 135 \end{cases}$$

The error locator polynomial is

$$\Lambda(x) = 135x^2 + 192x + 1 = (1 + 128x)(1 + 64x) = (1 + 2^7x)(1 + 2^6x).$$

Thus, the errors are located at the positions $k_1 = \log_2(128) = \log_2(2^7) = 7$ and $k_2 = \log_2(64) = \log_2(2^6) = 6$.
To find the error values, solve the system (3)

$$\begin{pmatrix} 128^0 & 64^0 \\ 128^1 & 64^1 \end{pmatrix}\begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 192 \end{pmatrix}.$$

To solve it, modify the augmented matrix using Gaussian reduction algorithm;

$$\begin{pmatrix} 1 & 1 & 0 \\ 128 & 64 & 192 \end{pmatrix} \Longrightarrow \begin{pmatrix} 1 & 1 & 0 \\ 0 & 192 & 192 \end{pmatrix}, \text{ add 128 times the first row to the second row to get zero below the diagonal}$$

$$\Longrightarrow \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}, \text{ divide the second row by 192 to obtain ones on the diagonal}$$

$$\Longrightarrow \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}, \text{ add the second row to the first row to obtain the identity matrix}$$

$$\Longrightarrow \begin{cases} Y_1 = 1 \\ Y_2 = 1 \end{cases}$$

Thus, the error polynomial is $e(x) = x^7 + x^6$ and the decoding algorithm gives us the polynomial

$$c(x) = s(x) + e(x) = \left(x^7 + x^6\right) + \left(x^7 + x^6\right) = 0.$$

3.1.3 Example 3

Consider that we have sent the codeword $(0,0,0,0,0,0,0,0)$, but it was transmitted with the error vector $(0,0,0,1,1,0,0,0)$, so that $(0,0,0,1,1,0,0,0)$ was received.
The polynomial associated to the received message is

$$s(x) = x^4 + x^3.$$

We start with $v = 2$. The syndromes are

$$\begin{aligned} s_1 &= s(2^0) = s(1) = 1^4 + 1^3 = 1 + 1 = 0, \\ s_2 &= s(2^1) = s(2) = 2^4 + 2^3 = 16 + 8 = 24, \\ s_3 &= s(2^2) = s(4) = 4^4 + 4^3 = 29 + 64 = 93 \quad \text{and} \\ s_4 &= s(2^3) = s(8) = 8^4 + 8^3 = 205 + 58 = 247. \end{aligned}$$

We want to solve the system (4)

$$\begin{pmatrix} 0 & 24 \\ 24 & 93 \end{pmatrix} \begin{pmatrix} \Lambda_2 \\ \Lambda_1 \end{pmatrix} = \begin{pmatrix} 93 \\ 247 \end{pmatrix}. \tag{5}$$

Note that

$$\det \begin{pmatrix} 0 & 24 \\ 24 & 93 \end{pmatrix} = 0 \cdot 93 - 24 \cdot 24 = 93 \neq 0.$$

So the system must have a unique solution. To find it, modify the augmented matrix using Gaussian reduction algorithm;

$$\begin{aligned} \left(\begin{array}{cc|c} 0 & 24 & 93 \\ 24 & 93 & 247 \end{array}\right) &\Longrightarrow \left(\begin{array}{cc|c} 24 & 93 & 247 \\ 0 & 24 & 93 \end{array}\right), \text{ swap the rows since the first element of the diagonal is zero} \\ &\Longrightarrow \left(\begin{array}{cc|c} 1 & 24 & 128 \\ 0 & 1 & 24 \end{array}\right), \text{ divide all equations by 24 to obtain ones on the diagonal} \\ &\Longrightarrow \left(\begin{array}{cc|c} 1 & 0 & 128 \\ 0 & 1 & 24 \end{array}\right), \text{ add 24 times the second row to the first row to get the identity matrix} \\ &\Longrightarrow \begin{cases} \Lambda_1 = 24 \\ \Lambda_2 = 128 \end{cases} \end{aligned}$$

The error locator polynomial is

$$\Lambda(x) = 128x^2 + 24x + 1 = (1 + 8x)(1 + 16x) = (1 + 2^3 x)(1 + 2^4 x).$$

Thus, the errors are located at the positions $k_1 = \log_2(8) = \log_2(2^3) = 3$ and $k_2 = \log_2(16) = \log_2(2^4) = 4$.
To find the error values, solve the system (3)

$$\begin{pmatrix} 8^0 & 16^0 \\ 8^1 & 16^1 \end{pmatrix} \begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 24 \end{pmatrix}.$$

To solve it, modify the augmented matrix using Gaussian reduction algorithm;

$$\begin{aligned} \left(\begin{array}{cc|c} 1 & 1 & 0 \\ 8 & 16 & 24 \end{array}\right) &\Longrightarrow \left(\begin{array}{cc|c} 1 & 1 & 0 \\ 0 & 24 & 24 \end{array}\right), \text{ add 8 times the first row to the second row to get zero below the diagonal} \\ &\Longrightarrow \left(\begin{array}{cc|c} 1 & 1 & 0 \\ 0 & 1 & 1 \end{array}\right), \text{ divide the second row by 24 to obtain ones on the diagonal} \\ &\Longrightarrow \left(\begin{array}{cc|c} 1 & 0 & 1 \\ 0 & 1 & 1 \end{array}\right), \text{ add the second row to the first row to obtain the identity matrix} \\ &\Longrightarrow \begin{cases} Y_1 = 1 \\ Y_2 = 1 \end{cases} \end{aligned}$$

Thus, the error polynomial is $e(x) = x^4 + x^3$ and the decoding algorithm gives us the polynomial

$$c(x) = s(x) + e(x) = \left(x^4 + x^3\right) + \left(x^4 + x^3\right) = 0.$$

3.1.4 Example 4

Consider that we have sent the codeword $(0,0,0,1,15,54,120,64)$, but it was transmitted with the error vector $(0,0,0,1,0,0,0,0)$, so that $(0,0,0,0,15,54,120,64)$ was received.
The polynomial associated to the received message is

$$s(x) = 15x^3 + 54x^2 + 120x + 64.$$

We start with $v = 2$. The syndromes are

$$\begin{aligned}
s_1 &= s(2^0) = s(1) = 15 \cdot 1^3 + 54 \cdot 1^2 + 120 \cdot 1 + 64 = 15 + 54 + 120 + 64 = 1,\\
s_2 &= s(2^1) = s(2) = 15 \cdot 2^3 + 54 \cdot 2^2 + 120 \cdot 2 + 64 = 120 + 216 + 240 + 64 = 16,\\
s_3 &= s(2^2) = s(4) = 15 \cdot 4^3 + 54 \cdot 4^2 + 120 \cdot 4 + 64 = 231 + 71 + 253 + 64 = 29 \quad \text{and}\\
s_4 &= s(2^3) = s(8) = 15 \cdot 8^3 + 54 \cdot 8^2 + 120 \cdot 8 + 64 = 107 + 1 + 231 + 64 = 205.
\end{aligned}$$

We want to solve the system (4)

$$\begin{pmatrix} 1 & 16 \\ 16 & 29 \end{pmatrix} \begin{pmatrix} \Lambda_2 \\ \Lambda_1 \end{pmatrix} = \begin{pmatrix} 29 \\ 205 \end{pmatrix}.$$

However, note that

$$\det \begin{pmatrix} 1 & 16 \\ 16 & 29 \end{pmatrix} = 1 \cdot 29 - 16 \cdot 16 = 0.$$

So we must decrease v by 1, so that $v = 1$. This yields the system

$$1 \cdot \Lambda_1 = 16$$

which has solution $\Lambda_1 = 16$. The error locator polynomial is $\Lambda(x) = 1 + 16x$, so that $X_1 = 16 = 2^4$. That is, the error is located at the position $k_1 = \log_2(16) = \log_2(2^4) = 4$.
To find the error values, solve

$$X_1^0 \cdot Y_1 = s_1 \implies 1 \cdot Y_1 = 1 \implies Y_1 = 1.$$

Thus, the error polynomial is $e(x) = x^4$ and the decoding algorithm proposes that the associated polynomial is

$$c(x) = s(x) + e(x) = x^4 + 15x^3 + 54x^2 + 120x + 64.$$

3.1.5 Example 5

Consider that we have sent the codeword $(0,0,0,1,15,54,120,64)$, but it was transmitted with the error vector $(0,0,0,1,1,0,0,0)$, so that $(0,0,0,0,14,54,120,64)$ was received.
The polynomial associated to the received message is

$$s(x) = 14x^3 + 54x^2 + 120x + 64.$$

We start with $v = 2$. The syndromes are

$$\begin{aligned}
s_1 &= s(2^0) = s(1) = 14 \cdot 1^3 + 54 \cdot 1^2 + 120 \cdot 1 + 64 = 14 + 54 + 120 + 64 = 0,\\
s_2 &= s(2^1) = s(2) = 14 \cdot 2^3 + 54 \cdot 2^2 + 120 \cdot 2 + 64 = 112 + 216 + 240 + 64 = 24,\\
s_3 &= s(2^2) = s(4) = 14 \cdot 4^3 + 54 \cdot 4^2 + 120 \cdot 4 + 64 = 167 + 71 + 253 + 64 = 93 \quad \text{and}\\
s_4 &= s(2^3) = s(8) = 14 \cdot 8^3 + 54 \cdot 8^2 + 120 \cdot 8 + 64 = 81 + 1 + 231 + 64 = 247.
\end{aligned}$$

Note that the syndromes are the same as those in Example 3. So the system (4) is the same as in (5), and so is its solution. Hence, the error locator polynomial is

$$\Lambda(x) = 128x^2 + 24x + 1 = (1 + 2^3x)(1 + 2^4x).$$

Thus, the errors are located at the positions $k_1 = 3$ and $k_2 = 4$.
System (3) is also the same as in Example 3; thus, the error values are $Y_1 = 1$ and $Y_2 = 1$.
The error polynomial is $e(x) = x^4 + x^3$ and the decoding algorithm proposes that the associated polynomial is

$$c(x) = s(x) + e(x) = (x^4 + x^3) + 14x^3 + 54x^2 + 120x + 64 = x^4 + 15x^3 + 54x^2 + 120x + 64.$$

3.2 Berlekamp-Massey Algorithm

Recall the system of equations

$$\Lambda_v s_{k-v} + \Lambda_{v-1} s_{k-v+1} + \ldots + \Lambda_1 s_{k-1} = -s_k \quad \text{for } v < k \leq 2t,$$

where the error locator polynomial is

$$\Lambda(x) = \Lambda_v x^v + \Lambda_{v-1} x^{v-1} + \ldots + \Lambda_1 x + \Lambda_0$$

with $\Lambda_0 = 1$. We want to determine a polynomial $\Lambda(x)$ of minimal degree v that satisfies all the required syndrome relations. In particular,

$$\Lambda_v s_{k-v} + \Lambda_{v-1} s_{k-v+1} + \ldots + \Lambda_1 s_{k-1} + \Lambda_0 s_k = 0 \quad \text{for } v < k \leq 2t.$$

Given syndromes $s_1, s_2, \ldots, s_{2t}$, the Berlekamp-Massey algorithm recursively defines field elements $d^{(k)}$, also called **discrepancies**, polynomials $\Lambda^{(k)}(x)$ of degree $v^{(k)}$, and polynomials $B^{(k)}(x)$ for $k = 1, 2, \ldots, 2t$.

The algorithm goes as follows:

1. Set $v^{(0)} = 0$, $\Lambda^{(0)}(x) = 1$ and $B^{(0)}(x) = 1$.

   Repeat Steps 2 and 3 for $k = 1, \ldots, 2t$.

2. Set
$$d^{(k)} = \sum_{i=0}^{m} c_i s_{k-i} = c_0 s_k + c_1 s_{k-1} + \ldots + c_{m-1} s_{k-m+1} + c_m s_{k-m},$$
   where $m = v^{(k-1)}$ and $\Lambda^{(k-1)}(x) = \sum_{i=0}^{m} c_i x^i$ have been defined in previous steps of the algorithm.

3. Dependent on $v^{(k-1)}$ and $d^{(k)}$ proceed as follows:

   - If $d^{(k)} = 0$, set
   $$\begin{aligned} v^{(k)} &= v^{(k-1)}, \\ \Lambda^{(k)}(x) &= \Lambda^{(k-1)}(x), \\ B^{(k)}(x) &= xB^{(k-1)}(x). \end{aligned}$$

   - If $d^{(k)} \neq 0$ and $2v^{(k-1)} \leq k - 1$, set
   $$\begin{aligned} v^{(k)} &= k - v^{(k-1)}, \\ \Lambda^{(k)}(x) &= \Lambda^{(k-1)}(x) - d^{(k)} xB^{(k-1)}(x), \\ B^{(k)}(x) &= (d^{(k)})^{-1} \Lambda^{(k-1)}(x). \end{aligned}$$

   - If $d^{(k)} \neq 0$ and $2v^{(k-1)} > k - 1$, set
   $$\begin{aligned} v^{(k)} &= v^{(k-1)}, \\ \Lambda^{(k)}(x) &= \Lambda^{(k-1)}(x) - d^{(k)} xB^{(k-1)}(x), \\ B^{(k)}(x) &= xB^{(k-1)}(x). \end{aligned}$$

4. End with $\Lambda(x) = \Lambda^{(2t)}(x)$ and $v = v^{(2t)}$.

Now we will repeat the examples provided in the previous section, this time finding the error locator polynomial using the Berlekamp-Massey algorithm.

3.2.1 Example 1

Consider that we have sent the codeword $(0,0,0,0,0,0,0,0)$, but it was transmitted with the error vector $(0,0,0,0,0,0,1,1)$, so that $(0,0,0,0,0,0,1,1)$ was received.
The polynomial associated to the message received is

$$s(x) = x + 1.$$

The syndromes are

$$\begin{aligned} s_1 &= s(2^0) = s(1) = 0, \\ s_2 &= s(2^1) = s(2) = 3, \\ s_3 &= s(2^2) = s(4) = 5 \quad \text{and} \\ s_4 &= s(2^3) = s(8) = 9. \end{aligned}$$

We start by setting $v^{(0)} = 0$, $\Lambda^{(0)}(x) = 1$ and $B^{(0)}(x) = 1$. Note that the number of syndromes is $2t = 4$, so that we will iterate over $k = 1, 2, 3, 4$.
For $k = 1$, we have $m = v^{(0)} = 0$ and

$$d^{(1)} = \sum_{i=0}^{0} c_i s_{1-i} = c_0 s_1 = 1 \cdot 0 = 0.$$

Since $d^{(1)} = 0$, we set

$$\begin{aligned} v^{(1)} &= v^{(0)} = 0, \\ \Lambda^{(1)}(x) &= \Lambda^{(0)}(x) = 1, \\ B^{(1)}(x) &= xB^{(0)}(x) = x. \end{aligned}$$

For $k = 2$, we have $m = v^{(1)} = 0$ and

$$d^{(2)} = \sum_{i=0}^{0} c_i s_{2-i} = c_0 s_2 = 1 \cdot 3 = 3.$$

Since $d^{(2)} \neq 0$ and $2v^{(1)} = 0 \leq 1 = k - 1$, we set

$$\begin{aligned} v^{(2)} &= 2 - v^{(1)} = 2, \\ \Lambda^{(2)}(x) &= \Lambda^{(1)}(x) - d^{(2)} x B^{(1)}(x) = 1 - 3x \cdot x = 1 + 3x^2, \\ B^{(2)}(x) &= (d^{(2)})^{-1} \Lambda^{(1)}(x) = 3^{-1} \cdot 1 = 244. \end{aligned}$$

For $k = 3$, we have

$$d^{(3)} = \sum_{i=0}^{2} c_i s_{3-i} = c_0 s_3 + c_1 s_2 + c_2 s_1 = 1 \cdot 5 + 0 \cdot 3 + 3 \cdot 0 = 5.$$

Since $d^{(3)} \neq 0$ and $2v^{(2)} = 4 > 2 = k - 1$, we set

$$v^{(3)} = v^{(2)} = 2,$$

$$\Lambda^{(3)}(x) = \Lambda^{(2)}(x) - d^{(3)}xB^{(2)}(x) = (1+3x^2) - 5x \cdot 244 = (1+3x^2) - 3x = 1+3x+3x^2,$$
$$B^{(3)}(x) = xB^{(2)}(x) = 244x.$$

For $k = 4$, set

$$d^{(4)} = \sum_{i=0}^{2} c_i s_{4-i} = c_0 s_4 + c_1 s_3 + c_2 s_2 = 1 \cdot 9 + 3 \cdot 5 + 3 \cdot 3 = 3.$$

Since $d^{(4)} \neq 0$ and $2v^{(3)} = 4 > 3 = k-1$, we set

$$v^{(4)} = v^{(3)} = 2,$$
$$\Lambda^{(4)}(x) = \Lambda^{(3)}(x) - d^{(4)}xB^{(3)}(x) = (1+3x+3x^2) - 3x \cdot 244x = (1+3x+3x^2) - x^2 = 1+3x+2x^2,$$
$$B^{(4)}(x) = xB^{(3)}(x) = 244x^2.$$

The algorithm ends and provides the desired error locator polynomial

$$\Lambda(x) = \Lambda^{(4)}(x) = 1+3x+2x^2.$$

This error locator polynomial is handled in the same way as in the end of the Peterson-Gorenstein-Zierler Algorithm.

3.2.2 Example 2

Consider that we have sent the codeword $(0,0,0,0,0,0,0,0)$, but it was transmitted with the error vector $(1,1,0,0,0,0,0,0)$, so that $(1,1,0,0,0,0,0,0)$ was received.
The polynomial associated to the message received is

$$s(x) = x^7 + x^6.$$

The syndromes are

$$s_1 = s(2^0) = s(1) = 0,$$
$$s_2 = s(2^1) = s(2) = 192,$$
$$s_3 = s(2^2) = s(4) = 222 \quad \text{and}$$
$$s_4 = s(2^3) = s(8) = 88.$$

We start by setting $v^{(0)} = 0$, $\Lambda^{(0)}(x) = 1$ and $B^{(0)}(x) = 1$. Note that the number of syndromes is $2t = 4$, so that we will iterate over $k = 1,2,3,4$.
For $k = 1$, we have $m = v^{(0)} = 0$ and

$$d^{(1)} = \sum_{i=0}^{0} c_i s_{1-i} = c_0 s_1 = 1 \cdot 0 = 0.$$

Since $d^{(1)} = 0$, we set

$$v^{(1)} = v^{(0)} = 0,$$
$$\Lambda^{(1)}(x) = \Lambda^{(0)}(x) = 1,$$
$$B^{(1)}(x) = xB^{(0)}(x) = x.$$

For $k = 2$, we have $m = v^{(1)} = 0$ and

$$d^{(2)} = \sum_{i=0}^{0} c_i s_{2-i} = c_0 s_2 = 1 \cdot 192 = 192.$$

Since $d^{(2)} \neq 0$ and $2v^{(1)} = 0 \leq 1 = k-1$, we set

$$\begin{aligned} v^{(2)} &= 2 - v^{(1)} = 2, \\ \Lambda^{(2)}(x) &= \Lambda^{(1)}(x) - d^{(2)} x B^{(1)}(x) = 1 - 192x \cdot x = 1 + 192x^2, \\ B^{(2)}(x) &= (d^{(2)})^{-1} \Lambda^{(1)}(x) = 192^{-1} \cdot 1 = 18. \end{aligned}$$

For $k = 3$, we have

$$d^{(3)} = \sum_{i=0}^{2} c_i s_{3-i} = c_0 s_3 + c_1 s_2 + c_2 s_1 = 1 \cdot 222 + 0 \cdot 192 + 192 \cdot 0 = 222.$$

Since $d^{(3)} \neq 0$ and $2v^{(2)} = 4 > 2 = k-1$, we set

$$\begin{aligned} v^{(3)} &= v^{(2)} = 2, \\ \Lambda^{(3)}(x) &= \Lambda^{(2)}(x) - d^{(3)} x B^{(2)}(x) = (1 + 192x^2) - 222x \cdot 18 = (1 + 192x^2) - 192x = 1 + 192x + 192x^2, \\ B^{(3)}(x) &= x B^{(2)}(x) = 18x. \end{aligned}$$

For $k = 4$, set

$$d^{(4)} = \sum_{i=0}^{2} c_i s_{4-i} = c_0 s_4 + c_1 s_3 + c_2 s_2 = 1 \cdot 88 + 192 \cdot 222 + 192 \cdot 192 = 48.$$

Since $d^{(4)} \neq 0$ and $2v^{(3)} = 4 > 3 = k-1$, we set

$$\begin{aligned} v^{(4)} &= v^{(3)} = 2, \\ \Lambda^{(4)}(x) &= \Lambda^{(3)}(x) - d^{(4)} x B^{(3)}(x) = (1 + 192x + 192x^2) - 48x \cdot 18x = (1 + 192x + 192x^2) - 71x^2 = 1 + 192x + 135x^2, \\ B^{(4)}(x) &= x B^{(3)}(x) = 18x^2. \end{aligned}$$

The algorithm ends and provides the desired error locator polynomial

$$\Lambda(x) = \Lambda^{(4)}(x) = 1 + 192x + 135x^2.$$

This error locator polynomial is handled in the same way as in the end of the Peterson-Gorenstein-Zierler Algorithm.

3.2.3 Example 3

Consider that we have sent the codeword $(0,0,0,0,0,0,0,0)$, but it was transmitted with the error vector $(0,0,0,1,1,0,0,0)$, so that $(0,0,0,1,1,0,0,0)$ was received.
The polynomial associated to the received message is

$$s(x) = x^4 + x^3$$

The syndromes are

$$\begin{aligned} s_1 &= s(2^0) = s(1) = 0, \\ s_2 &= s(2^1) = s(2) = 24, \\ s_3 &= s(2^2) = s(4) = 93 \quad \text{and} \\ s_4 &= s(2^3) = s(8) = 247. \end{aligned}$$

We start by setting $v^{(0)} = 0$, $\Lambda^{(0)}(x) = 1$ and $B^{(0)}(x) = 1$. Note that the number of syndromes is $2t = 4$, so that we will iterate over $k = 1, 2, 3, 4$.

For $k = 1$, we have $m = v^{(0)} = 0$ and

$$d^{(1)} = \sum_{i=0}^{0} c_i s_{1-i} = c_0 s_1 = 1 \cdot 0 = 0.$$

Since $d^{(1)} = 0$, we set

$$\begin{aligned} v^{(1)} &= v^{(0)} = 0, \\ \Lambda^{(1)}(x) &= \Lambda^{(0)}(x) = 1, \\ B^{(1)}(x) &= xB^{(0)}(x) = x. \end{aligned}$$

For $k = 2$, we have $m = v^{(1)} = 0$ and

$$d^{(2)} = \sum_{i=0}^{0} c_i s_{2-i} = c_0 s_2 = 1 \cdot 24 = 24.$$

Since $d^{(2)} \neq 0$ and $2v^{(1)} = 0 \leq 1 = k - 1$, we set

$$\begin{aligned} v^{(2)} &= 2 - v^{(1)} = 2, \\ \Lambda^{(2)}(x) &= \Lambda^{(1)}(x) - d^{(2)} x B^{(1)}(x) = 1 - 24x \cdot x = 1 + 24x^2, \\ B^{(2)}(x) &= (d^{(2)})^{-1} \Lambda^{(1)}(x) = 24^{-1} \cdot 1 = 144. \end{aligned}$$

For $k = 3$, we have

$$d^{(3)} = \sum_{i=0}^{2} c_i s_{3-i} = c_0 s_3 + c_1 s_2 + c_2 s_1 = 1 \cdot 93 + 0 \cdot 24 + 24 \cdot 0 = 93.$$

Since $d^{(3)} \neq 0$ and $2v^{(2)} = 4 > 2 = k - 1$, we set

$$\begin{aligned} v^{(3)} &= v^{(2)} = 2, \\ \Lambda^{(3)}(x) &= \Lambda^{(2)}(x) - d^{(3)} x B^{(2)}(x) = (1 + 24x^2) - 93x \cdot 144 = (1 + 24x^2) - 24x = 1 + 24x + 24x^2, \\ B^{(3)}(x) &= xB^{(2)}(x) = 144x. \end{aligned}$$

For $k = 4$, set

$$d^{(4)} = \sum_{i=0}^{2} c_i s_{4-i} = c_0 s_4 + c_1 s_3 + c_2 s_2 = 1 \cdot 247 + 24 \cdot 93 + 24 \cdot 24 = 193.$$

Since $d^{(4)} \neq 0$ and $2v^{(3)} = 4 > 3 = k - 1$, we set

$$\begin{aligned} v^{(4)} &= v^{(3)} = 2, \\ \Lambda^{(4)}(x) &= \Lambda^{(3)}(x) - d^{(4)} x B^{(3)}(x) = (1 + 24x + 24x^2) - 193x \cdot 144x = (1 + 24x + 24x^2) - 152x^2 = 1 + 24x + 128x^2, \\ B^{(4)}(x) &= xB^{(3)}(x) = 144x^2. \end{aligned}$$

The algorithm ends and provides the desired error locator polynomial

$$\Lambda(x) = \Lambda^{(4)}(x) = 1 + 24x + 128x^2.$$

This error locator polynomial is handled in the same way as in the end of the Peterson-Gorenstein-Zierler Algorithm.

3.2.4 Example 4

Consider that we have sent the codeword $(0,0,0,1,15,54,120,64)$, but it was transmitted with the error vector $(0,0,0,1,0,0,0,0)$, so that $(0,0,0,0,15,54,120,64)$ was received.
The polynomial associated to the received message is

$$s(x) = 15x^3 + 54x^2 + 120x + 64$$

The syndromes are

$$\begin{aligned} s_1 &= s(2^0) = s(1) = 1, \\ s_2 &= s(2^1) = s(2) = 16, \\ s_3 &= s(2^2) = s(4) = 29 \quad \text{and} \\ s_4 &= s(2^3) = s(8) = 205. \end{aligned}$$

We start by setting $v^{(0)} = 0$, $\Lambda^{(0)}(x) = 1$ and $B^{(0)}(x) = 1$. Note that the number of syndromes is $2t = 4$, so that we will iterate over $k = 1, 2, 3, 4$.
For $k = 1$, we have $m = v^{(0)} = 0$ and

$$d^{(1)} = \sum_{i=0}^{0} c_i s_{1-i} = c_0 s_1 = 1 \cdot 1 = 1.$$

Since $d^{(1)} \neq 0$ and $2v^{(0)} = 0 \leq 0 = k - 1$, we set

$$\begin{aligned} v^{(1)} &= 1 - v^{(0)} = 1, \\ \Lambda^{(1)}(x) &= \Lambda^{(0)}(x) - d^{(1)} x B^{(0)}(x) = 1 - 1x \cdot 1 = 1 + x, \\ B^{(1)}(x) &= (d^{(1)})^{-1} \Lambda^{(0)}(x) = 1^{-1} \cdot 1 = 1. \end{aligned}$$

For $k = 2$, we have $m = v^{(1)} = 1$ and

$$d^{(2)} = \sum_{i=0}^{1} c_i s_{2-i} = c_0 s_2 + c_1 s_1 = 1 \cdot 16 + 1 \cdot 1 = 17.$$

Since $d^{(2)} \neq 0$ and $2v^{(1)} = 2 > 1 = k - 1$, we set

$$\begin{aligned} v^{(2)} &= v^{(1)} = 1, \\ \Lambda^{(2)}(x) &= \Lambda^{(1)}(x) - d^{(2)} x B^{(1)}(x) = (1 + x) - 17x \cdot 1 = (1 + x) - 17x = 1 + 16x, \\ B^{(2)}(x) &= x B^{(1)}(x) = x. \end{aligned}$$

For $k = 3$, we have

$$d^{(3)} = \sum_{i=0}^{1} c_i s_{3-i} = c_0 s_3 + c_1 s_2 = 1 \cdot 29 + 16 \cdot 16 = 0.$$

Since $d^{(3)} = 0$ and $2v^{(2)} = 4 > 2 = k - 1$, we set

$$\begin{aligned} v^{(3)} &= v^{(2)} = 1, \\ \Lambda^{(3)}(x) &= \Lambda^{(2)}(x) = 1 + 16x, \\ B^{(3)}(x) &= x B^{(2)}(x) = x \cdot x = x^2. \end{aligned}$$

For $k = 4$, set

$$d^{(4)} = \sum_{i=0}^{1} c_i s_{4-i} = c_0 s_4 + c_1 s_3 = 1 \cdot 205 + 16 \cdot 29 = 0.$$

Since $d^{(4)} = 0$, we set

$$\begin{aligned} v^{(4)} &= v^{(3)} = 1, \\ \Lambda^{(4)}(x) &= \Lambda^{(3)}(x) = 1 + 16x, \\ B^{(4)}(x) &= xB^{(3)}(x) = x \cdot x^2 = x^3. \end{aligned}$$

The algorithm ends and provides the desired error locator polynomial

$$\Lambda(x) = \Lambda^{(4)}(x) = 1 + 16x.$$

This error locator polynomial is handled in the same way as in the end of the Peterson-Gorenstein-Zierler Algorithm.

3.2.5 Example 5

Consider that we have sent the codeword $(0,0,0,1,15,54,120,64)$, but it was transmitted with the error vector $(0,0,0,1,1,0,0,0)$, so that $(0,0,0,0,14,54,120,64)$ was received.
The polynomial associated to the received message is

$$s(x) = 14x^3 + 54x^2 + 120x + 64$$

The syndromes are

$$\begin{aligned} s_1 &= s(2^0) = s(1) = 0, \\ s_2 &= s(2^1) = s(2) = 24, \\ s_3 &= s(2^2) = s(4) = 93 \quad \text{and} \\ s_4 &= s(2^3) = s(8) = 247. \end{aligned}$$

Since these are the same syndromes as in Example 3, the procedure for this algorithm will be identical. Hence, the error locator polynomial is

$$\Lambda(x) = 128x^2 + 24x + 1.$$

4 Finding Roots of Polynomials

Recall that with both decoding algorithms, to find the error positions, we must find the roots of the error locator polynomial, this is, solve the equation

$$\Lambda(x) = \prod_{j=1}^{v}(1 - X_j x) = \Lambda_v x^v + \Lambda_{v-1} x^{v-1} + \ldots + \Lambda_1 x + \Lambda_0 = 0.$$

A method to solve this system is Chien Search.

4.1 Chien Search

If we are given a polynomial $f(x) = a_0 + a_1 x + \ldots + a_t x^t$ with coefficients over the Galois field $GF(256)$, Chien search is an algorithm that helps us with finding the roots of this polynomial. It is faster than just going through the list $0, 1, 2, \ldots, 255$ of all field elements with a simple trial-and-error approach.
It works as a consequence of the following two facts:

- Any nonzero element in the Galois field $GF(256)$ can be expressed as a power of 2.
- We can write $f(2^{i+1})$ in terms of summands that appear in the evaluation of $f(2^i)$. More precisely, call $g_{j,i} = a_j(2^i)^j$, so that

$$f(2^i) = g_{0,i} + g_{1,i} + \ldots + g_{t,i}.$$

We will obtain two properties of the $g_{j,i}$:

$$g_{j,0} = a_j(2^0)^j = a_j \cdot 1^j = a_j, \quad \text{and} \quad g_{j,i+1} = a_j(2^{i+1})^j = a_j 2^{(i+1)j} = a_j(2^i)^j \cdot 2^j = g_{j,i} \cdot 2^j$$

This leads us to switch to a more efficient order of the operations, by following the steps of Chien search:

1. Create the table

i	$g_{0,i}$	$g_{1,i}$	$g_{2,i}$	$\ldots$	$g_{t,i}$	$f(2^i)$
0						
1						
2						
$\vdots$						
255						

2. Fill the $i = 0$ row of the table using the rule $g_{j,0} = a_j$.
3. Fill the $g_{0,i}$ column of the table, using the rule $g_{0,i+1} = g_{0,i} \cdot 2^0 = g_{0,i}$. This is, the second column is identical to $g_{0,0} = a_0$.
4. Fill the $g_{1,i}$ column of the table, using the rule $g_{1,i+1} = g_{1,i} \cdot 2^1 = 2 \cdot g_{1,i}$. This is, for each cell, multiply its value by two to get the value in the cell immediately below.
5. Fill the $g_{2,i}$ column of the table, using the rule $g_{2,i+1} = g_{2,i} \cdot 2^2 = 4 \cdot g_{2,i}$. This is, for each cell, multiply its value by four to get the value in the cell immediately below.
6. Continue filling the $g_{0,i}, g_{1,i}, \ldots, g_{t,i}$ columns of the table in a similar way than the two previous steps. The constant factor for the $g_{j,i}$ column is 2^j, i.e., $g_{j,i+1} = 2^j \cdot g_{j,i}$.
7. Fill the cells of the last column by adding up the $g_{0,i}, g_{1,i}, \ldots, g_{t,i}$ columns.
8. Check which rows have a zero in the last column to determine the roots of the polynomial $f(x)$.

4.1.1 Example 1

Consider that we have sent the codeword $(0,0,0,0,0,0,0,0)$, but it was transmitted with the error vector $(0,0,0,0,0,0,1,1)$, so that $(0,0,0,0,0,0,1,1)$ was received.
The error locator polynomial is

$$\Lambda(x) = 2x^2 + 3x + 1.$$

Now, we fill the table as follows:

- In the row $i = 0$ we write the coefficients of $\Lambda(x)$: 1, 3 and 2.
- Every value in the column $g_{0,i}$ is 1.
- The cells in the column $g_{1,i}$ are filled with twice the value above, so its values are 3, 6, 12, 24, 48, 96, 192, 157, and continue doubling until the row $i = 254$, where the value is 143.

- The cells in the column $g_{2,i}$ are filled with four times the value above, so its values are 2, 8, 32, 128, 58, 232, 135, 38, and continue quadrupling until the row $i = 254$, where the value is 142.
- The last column $\Lambda(x)$ has the sum of the values in the columns $g_{0,i}$, $g_{1,i}$ and $g_{2,i}$. In particular,

$$\Lambda(2^0) = 1 + 3 + 2 = 0 \quad \text{and} \quad \Lambda(2^{254}) = 1 + 143 + 142 = 0$$

i	2^i	$g_{0,i}$	$g_{1,i}$	$g_{2,i}$	$\Lambda(2^i)$
0	1	1	3	2	0
1	2	1	6	8	15
2	4	1	12	32	45
3	8	1	24	128	153
4	16	1	48	58	11
5	32	1	96	232	137
6	64	1	192	135	70
7	128	1	157	38	186
8	29	1	39	152	190
9	58	1	78	90	21
10	116	1	156	117	232
11	232	1	37	201	237
12	205	1	74	3	72
13	135	1	148	12	153
14	19	1	53	48	4
15	38	1	106	192	171
16	76	1	212	39	242
17	152	1	181	156	40
18	45	1	119	74	60
19	90	1	238	53	218
20	180	1	193	212	20
21	117	1	159	119	233
22	234	1	35	193	227
23	201	1	70	35	100
24	143	1	140	140	1
25	3	1	5	10	14
26	6	1	10	40	35
27	12	1	20	160	181
28	24	1	40	186	147
29	48	1	80	210	131
30	96	1	160	111	206
31	192	1	93	161	253
32	157	1	186	190	5
33	39	1	105	194	170
34	78	1	210	47	252
35	156	1	185	188	4
36	37	1	111	202	164
37	74	1	222	15	208
38	148	1	161	60	156
39	53	1	95	240	174
40	106	1	190	231	88
41	212	1	97	187	219
42	181	1	194	214	21
43	119	1	153	127	231
44	238	1	47	225	207
45	193	1	94	163	252
46	159	1	188	182	11
47	35	1	101	226	134
48	70	1	202	175	100
49	140	1	137	134	14
50	5	1	15	34	44
51	10	1	30	136	151
52	20	1	60	26	39
53	40	1	120	104	17
54	80	1	240	189	76
55	160	1	253	206	50
56	93	1	231	31	249
57	186	1	211	124	174
58	105	1	187	237	87
59	210	1	107	147	249
60	185	1	214	118	161
61	111	1	177	197	117
62	222	1	127	51	77
63	161	1	254	204	51

i	2^i	$g_{0,i}$	$g_{1,i}$	$g_{2,i}$	$\Lambda(2^i)$
64	95	1	225	23	247
65	190	1	223	92	130
66	97	1	163	109	207
67	194	1	91	169	243
68	153	1	182	158	41
69	47	1	113	66	50
70	94	1	226	21	246
71	188	1	217	84	140
72	101	1	175	77	227
73	202	1	67	41	107
74	137	1	134	164	35
75	15	1	17	170	186
76	30	1	34	146	177
77	60	1	68	114	55
78	120	1	136	213	92
79	240	1	13	115	127
80	253	1	26	209	202
81	231	1	52	99	86
82	211	1	104	145	248
83	187	1	208	126	175
84	107	1	189	229	89
85	214	1	103	179	213
86	177	1	206	246	57
87	127	1	129	255	127
88	254	1	31	219	197
89	225	1	62	75	116
90	223	1	124	49	76
91	163	1	248	196	61
92	91	1	237	55	219
93	182	1	199	220	26
94	113	1	147	87	197
95	226	1	59	65	123
96	217	1	118	25	110
97	175	1	236	100	137
98	67	1	197	141	73
99	134	1	151	14	152
100	17	1	51	56	10
101	34	1	102	224	135
102	68	1	204	167	106
103	136	1	133	166	34
104	13	1	23	162	180
105	26	1	46	178	157
106	52	1	92	242	175
107	104	1	184	239	86
108	208	1	109	155	247
109	189	1	218	86	141
110	103	1	169	69	237
111	206	1	79	9	71
112	129	1	158	36	187
113	31	1	33	144	176
114	62	1	66	122	57
115	124	1	132	245	112
116	248	1	21	243	231
117	237	1	42	235	192
118	199	1	84	139	222
119	147	1	168	22	191
120	59	1	77	88	20
121	118	1	154	125	230
122	236	1	41	233	193
123	197	1	82	131	208
124	151	1	164	54	147
125	51	1	85	216	140
126	102	1	170	71	236
127	204	1	73	1	73

i	2^i	$g_{0,i}$	$g_{1,i}$	$g_{2,i}$	$\Lambda(2^i)$
128	133	1	146	4	151
129	23	1	57	16	40
130	46	1	114	64	51
131	92	1	228	29	248
132	184	1	213	116	160
133	109	1	183	205	123
134	218	1	115	19	97
135	169	1	230	76	171
136	79	1	209	45	253
137	158	1	191	180	10
138	33	1	99	234	136
139	66	1	198	143	72
140	132	1	145	6	150
141	21	1	63	24	38
142	42	1	126	96	31
143	84	1	252	157	96
144	168	1	229	78	170
145	77	1	215	37	243
146	154	1	179	148	38
147	41	1	123	106	16
148	82	1	246	181	66
149	164	1	241	238	30
150	85	1	255	159	97
151	170	1	227	70	164
152	73	1	219	5	223
153	146	1	171	20	190
154	57	1	75	80	26
155	114	1	150	93	202
156	228	1	49	105	89
157	213	1	98	185	218
158	183	1	196	222	27
159	115	1	149	95	203
160	230	1	55	97	87
161	209	1	110	153	246
162	191	1	220	94	131
163	99	1	165	101	193
164	198	1	87	137	223
165	145	1	174	30	177
166	63	1	65	120	56
167	126	1	130	253	126
168	252	1	25	211	203
169	229	1	50	107	88
170	215	1	100	177	212
171	179	1	200	254	55
172	123	1	141	223	83
173	246	1	7	91	93
174	241	1	14	113	126
175	255	1	28	217	196
176	227	1	56	67	122
177	219	1	112	17	96
178	171	1	224	68	165
179	75	1	221	13	209
180	150	1	167	52	146
181	49	1	83	208	130
182	98	1	166	103	192
183	196	1	81	129	209
184	149	1	162	62	157
185	55	1	89	248	160
186	110	1	178	199	116
187	220	1	121	59	67
188	165	1	242	236	31
189	87	1	249	151	111
190	174	1	239	102	136
191	65	1	195	133	71

i	2^i	$g_{0,i}$	$g_{1,i}$	$g_{2,i}$	$\Lambda(2^i)$
192	130	1	155	46	180
193	25	1	43	184	146
194	50	1	86	218	141
195	100	1	172	79	226
196	200	1	69	33	101
197	141	1	138	132	15
198	7	1	9	42	34
199	14	1	18	168	187
200	28	1	36	154	191
201	56	1	72	82	27
202	112	1	144	85	196
203	224	1	61	73	117
204	221	1	122	57	66
205	167	1	244	228	17
206	83	1	245	183	67
207	166	1	247	230	16
208	81	1	243	191	77
209	162	1	251	198	60
210	89	1	235	63	213
211	178	1	203	252	54
212	121	1	139	215	93
213	242	1	11	123	113
214	249	1	22	241	230
215	239	1	44	227	206
216	195	1	88	171	242
217	155	1	176	150	39
218	43	1	125	98	30
219	86	1	250	149	110
220	172	1	233	110	134
221	69	1	207	165	107
222	138	1	131	174	44
223	9	1	27	130	152
224	18	1	54	50	5
225	36	1	108	200	165
226	72	1	216	7	222
227	144	1	173	28	176
228	61	1	71	112	54
229	122	1	142	221	82
230	244	1	1	83	83
231	245	1	2	81	82
232	247	1	4	89	92
233	243	1	8	121	112
234	251	1	16	249	232
235	235	1	32	195	226
236	203	1	64	43	106
237	139	1	128	172	45
238	11	1	29	138	150
239	22	1	58	18	41
240	44	1	116	72	61
241	88	1	232	61	212
242	176	1	205	244	56
243	125	1	135	247	113
244	250	1	19	251	233
245	233	1	38	203	236
246	207	1	76	11	70
247	131	1	152	44	181
248	27	1	45	176	156
249	54	1	90	250	161
250	108	1	180	207	122
251	216	1	117	27	111
252	173	1	234	108	135
253	71	1	201	173	101
254	142	1	143	142	0

We only have two roots of $\Lambda(x)$: $1 = 2^0 = 1^{-1}$ and $142 = 2^{254} = 2^{-1}$.
Thus, the error positions are $\log_2(1) = \log_2(2^0) = 0$ and $\log_2(2) = \log_2(2^1) = 1$.

Shortcut: Recall that $\Lambda(x) = 2x^2 + 3x + 1$ and note that the binary representation of 3 is $(0,0,0,0,0,0,1,1)$. This suggests that the error positions are 0 and 1, which is equivalent to the factorization

$$\Lambda(x) = (1 + 2^0 x)(1 + 2^1 x).$$

To confirm this, note that the sum and product of $2^0 = 1$ and $2^1 = 2$ are as follows:

Sum:

	0	0	0	0	0	0	1	0	2
+	0	0	0	0	0	0	0	1	1
	0	0	0	0	0	0	1	1	3

This is, the sum $2 + 1$ is $\Lambda_1 = 3$.

Product:

$2 \cdot 1 = 2$, which is Λ_2.

Note that checking all the elements of $GF(256)$ for roots consumes a lot of operations, even if we use Chien search. The following key observations help with finding the error positions faster.

Observation 1: Recall that the error locator polynomial helps us with finding the error positions; so we only have 8 possible roots of $\Lambda(x)$:

$$(2^0)^{-1} = 2^0, \quad (2^1)^{-1} = 2^{254}, \quad (2^2)^{-1} = 2^{253}, \quad (2^3)^{-1} = 2^{252}, \quad (2^4)^{-1} = 2^{251},$$

$$(2^5)^{-1} = 2^{250}, \quad (2^6)^{-1} = 2^{249}, \quad \text{and} \quad (2^7)^{-1} = 2^{248}.$$

Observation 2: Consider a polynomial of degree 2, $f(x) = (1+ax)(1+bx) = 1+(a+b)x+abx^2$, over $GF(256)$. The roots of f are a^{-1} and b^{-1}.
Now introduce the *reciprocal polynomial* $f^*(x)$ as follows:

$$f^*(x) = ab + (a + b)x + x^2 = (a + x)(b + x)$$

That is, we reverse the order of the coefficients. Since $a + a = 0$ and $b + b = 0$, we have that the roots of $f^*(x)$ are $a = (a^{-1})^{-1}$ and $b = (b^{-1})^{-1}$.
In conclusion, reversing the order of the coefficients provokes an inversion of the roots. For instance, in **Example 1** we found that $2x^2 + 3x + 1$ has roots 1 and 142. Then, $x^2 + 3x + 2$ has roots $1^{-1} = 1$ and $142^{-1} = 2$.

Observation 3: From Observation 1, the possible roots for $\Lambda^*(x)$ are:

$$2^0, \quad 2^1, \quad 2^2, \quad 2^3, \quad 2^4, \quad 2^5, \quad 2^6, \quad \text{and} \quad 2^7.$$

These observations motivate the following procedure for finding the error positions.

- Find the reciprocal error locator polynomial $\Lambda^*(x)$.
- Apply Chien search to $\Lambda^*(x)$, but only complete the table for the rows $i = 0, 1, 2, 3, 4, 5, 6, 7$.

Note: The conclusion of **Observation 2** also holds for polynomials of degree 1, $f(x) = 1 + ax$, with the *reciprocal polynomial* $f^*(x) = a + x$; so we can proceed by the method above in the case where the degree of the error locator polynomial is 1.

The Chien search for reciprocal error locator polynomial $\Lambda^*(x) = x^2 + 3x + 2$ produces the following chart:

i	2^i	$g_{0,i}$	$g_{1,i}$	$g_{2,i}$	$\Lambda^*(2^i)$
0	1	2	3	1	0
1	2	2	6	4	0
2	4	2	12	16	30
3	8	2	24	64	90
4	16	2	48	29	47
5	32	2	96	116	22
6	64	2	192	205	15
7	128	2	157	19	140

This confirms the errors in the $0th$ and $1st$ positions.

4.1.2 Example 2

Consider that we have sent the codeword $(0,0,0,0,0,0,0,0)$, but it was transmitted with the error vector $(1,1,0,0,0,0,0,0)$, so that $(1,1,0,0,0,0,0,0)$ was received.
The error locator polynomial is

$$\Lambda(x) = 135x^2 + 192x + 1.$$

The reciprocal error locator polynomial $\Lambda^*(x)$ is

$$\Lambda^*(x) = x^2 + 192x + 135.$$

The chart of the left is generated following Chien search for the error locator polynomial $\Lambda(x)$

- In the row $i = 0$ we write the coefficients of $\Lambda(x)$: 1, 192 and 135.
- In the row $i = 248$ goes 1, $192 \cdot 2^{248} = 192 \cdot 27 = 143$ and $135 \cdot 4^{248} = 135 \cdot 88 = 142$.
- Every value in the column $g_{0,i}$ is 1.
- Fill the missing in the column $g_{1,i}$ with twice the value above, so complete with the values 3, 6, 12, 24, 48 and 96.
- Fill the missing in the column $g_{2,i}$ with four times the value above, so complete with the values 2, 8, 32, 128, 58 and 232.
- The last column $\Lambda(2^i)$ has the sum of the values in the columns $g_{0,i}$, $g_{1,i}$ and $g_{2,i}$. In particular,

$$\Lambda(2^{248}) = 1 + 143 + 142 = 0 \quad \text{and} \quad \Lambda(2^{249}) = 1 + 3 + 2 = 0.$$

The table of the right is filled by Chien search for the reciprocal error locator polynomial $\Lambda^*(x)$

- In the row $i = 0$ we write the coefficients of $\Lambda^*(x)$: 135, 192 and 1.
- Every value in the column $g_{0,i}$ is 135.
- The cells in the column $g_{1,i}$ are filled with twice the value above, so its values are 192, 157, 39, 78, 156, 37, 74 and 148.
- The cells in the column $g_{2,i}$ are filled with four times the value above, so its values are 1, 4, 16, 64, 29, 116, 205 and 19.
- The last column $\Lambda^*(2^i)$ has the sum of the values in the columns $g_{0,i}$, $g_{1,i}$ and $g_{2,i}$. In particular,

$$\Lambda^*(2^6) = 135 + 74 + 205 = 0 \quad \text{and} \quad \Lambda^*(2^7) = 135 + 148 + 19 = 0.$$

i	2^i	$g_{0,i}$	$g_{1,i}$	$g_{2,i}$	$\Lambda(2^i)$
0	1	1	192	135	70
248	27	1	143	142	0
249	54	1	3	2	0
250	108	1	6	8	15
251	216	1	12	32	45
252	173	1	24	128	153
253	71	1	48	58	11
254	142	1	96	232	137

i	2^i	$g_{0,i}$	$g_{1,i}$	$g_{2,i}$	$\Lambda^*(2^i)$
0	1	135	192	1	70
1	2	135	157	4	30
2	4	135	39	16	176
3	8	135	78	64	137
4	16	135	156	29	6
5	32	135	37	116	178
6	64	135	74	205	0
7	128	135	148	19	0

There are two roots of $\Lambda(x)$: $54 = 64^{-1}$ and $27 = 128^{-1}$; and two roots of $\Lambda^*(x)$, which are $2^6 = 64$ and $2^7 = 128$.
Thus, the error positions are $\log_2(64) = \log_2(2^6) = 6$ and $\log_2(128) = \log_2(2^7) = 7$.

Shortcut: Recall that $\Lambda(x) = 135x^2+192x+1$ and note that the binary representation of 192 is $(1,1,0,0,0,0,0,0)$. This suggests that the error positions are 6 and 7, which is equivalent to the factorization

$$\Lambda(x) = (1+2^6x)(1+2^7x).$$

To confirm this, note that the sum and product of $2^6 = 64$ and $2^7 = 128$ are as follows:

Sum:

$$\begin{array}{cccccccccc} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 128 \\ + & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 64 \\ \hline & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 192 \end{array}$$

This is, the sum $128+64$ is $\Lambda_1 = 192$.

Product:

$$\begin{aligned} 128 \cdot 64 &= 128 \cdot 2^6 \\ &= (1,0,0,0,0,0,0,0) \cdot 2^6 \\ &= (0,0,0,1,1,1,0,1) \cdot 2^5 && 128 \cdot 2 = 29 \\ &= (0,0,1,1,1,0,1,0) \cdot 2^4 && 29 \cdot 2 = 58 \\ &= (0,1,1,1,0,1,0,0) \cdot 2^3 && 58 \cdot 2 = 116 \\ &= (1,1,1,0,1,0,0,0) \cdot 2^2 && 116 \cdot 2 = 232 \\ &= (1,1,0,0,1,1,0,1) \cdot 2 && 232 \cdot 2 = 205 \\ &= (1,0,0,0,0,1,1,1) && 205 \cdot 2 = 135 \\ &= 135 = \Lambda_2 \end{aligned}$$

4.1.3 Example 3

Consider that we have sent the codeword $(0,0,0,0,0,0,0,0)$, but it was transmitted with the error vector $(0,0,0,1,1,0,0,0)$, so that $(0,0,0,1,1,0,0,0)$ was received.
The error locator polynomial is

$$\Lambda(x) = 128x^2 + 24x + 1$$

The reciprocal error locator polynomial $\Lambda^*(x)$ is

$$\Lambda^*(x) = x^2 + 24x + 128.$$

The chart of the left is generated following Chien search for the error locator polynomial $\Lambda(x)$

- In the row $i = 0$ we write the coefficients of $\Lambda(x)$: 1, 24 and 128.
- In the row $i = 248$ goes 1, $24 \cdot 2^{248} = 24 \cdot 27 = 117$ and $128 \cdot 4^{248} = 128 \cdot 88 = 27$.
- Every value in the column $g_{0,i}$ is 1.
- Fill the missing in the column $g_{1,i}$ with twice the value above, so complete with the values 234, 201, 143, 3, 6 and 12.
- Fill the missing in the column $g_{2,i}$ with four times the value above, so complete with the values 108, 173, 142, 2, 8 and 32.
- The last column $\Lambda(2^i)$ has the sum of the values in the columns $g_{0,i}$, $g_{1,i}$ and $g_{2,i}$. In particular,

$$\Lambda(2^{251}) = 1 + 143 + 142 = 0 \quad \text{and} \quad \Lambda(2^{252}) = 1 + 3 + 2 = 0.$$

The table of the right is filled by Chien search for the reciprocal error locator polynomial $\Lambda^*(x)$

- In the row $i = 0$ we write the coefficients of $\Lambda^*(x)$: 128, 24 and 1.
- Every value in the column $g_{0,i}$ is 128.
- The cells in the column $g_{1,i}$ are filled with twice the value above, so its values are 24, 48, 96, 192, 157, 39, 78 and 156.
- The cells in the column $g_{2,i}$ are filled with four times the value above, so its values are 1, 4, 16, 64, 29, 116, 205 and 19.
- The last column $\Lambda^*(2^i)$ has the sum of the values in the columns $g_{0,i}$, $g_{1,i}$ and $g_{2,i}$. In particular,

$$\Lambda^*(2^3) = 128 + 192 + 64 = 0 \quad \text{and} \quad \Lambda^*(2^4) = 128 + 157 + 29 = 0.$$

i	2^i	$g_{0,i}$	$g_{1,i}$	$g_{2,i}$	$\Lambda(2^i)$
0	1	1	24	128	153
248	27	1	117	27	111
249	54	1	234	108	135
250	108	1	201	173	101
251	216	1	143	142	0
252	173	1	3	2	0
253	71	1	6	8	15
254	142	1	12	32	45

i	2^i	$g_{0,i}$	$g_{1,i}$	$g_{2,i}$	$\Lambda^*(2^i)$
0	1	128	24	1	153
1	2	128	48	4	180
2	4	128	96	16	240
3	8	128	192	64	0
4	16	128	157	29	0
5	32	128	39	116	211
6	64	128	78	205	3
7	128	128	156	19	15

There are two roots of $\Lambda(x)$: $173 = 8^{-1}$ and $216 = 16^{-1}$; and two roots of $\Lambda^*(x)$, which are $2^3 = 8$ and $2^4 = 16$. Thus, the error positions are $\log_2(8) = \log_2(2^3) = 3$ and $\log_2(16) = \log_2(2^4) = 4$.

Shortcut: Recall that $\Lambda(x) = 128x^2+24x+1$ and note that the binary representation of 24 is $(0, 0, 0, 1, 1, 0, 0, 0)$. This suggests that the error positions are 3 and 4, which is equivalent to the factorization

$$\Lambda(x) = (1 + 2^3x)(1 + 2^4x).$$

To confirm this, note that the sum and product of $2^3 = 8$ and $2^4 = 16$ are as follows:

Sum:

	0	0	0	1	0	0	0	0	16
+	0	0	0	0	1	0	0	0	8
	0	0	0	1	1	0	0	0	24

This is, the sum $16 + 8$ is $\Lambda_1 = 24$.

Product:

$$\begin{aligned} 16 \cdot 8 &= 16 \cdot 2^3 \\ &= (0,0,0,1,0,0,0,0) \cdot 2^3 \\ &= (0,0,1,0,0,0,0,0) \cdot 2^2 && 16 \cdot 2 = 32 \\ &= (0,1,0,0,0,0,0,0) \cdot 2 && 32 \cdot 2 = 64 \\ &= (1,0,0,0,0,0,0,0) && 64 \cdot 2 = 128 \\ &= 128 = \Lambda_2 \end{aligned}$$

4.1.4 Example 4

Consider that we have sent the codeword $(0,0,0,1,15,54,120,64)$, but it was transmitted with the error vector $(0,0,0,1,0,0,0,0)$, so that $(0,0,0,0,15,54,120,64)$ was received.
The error locator polynomial is

$$\Lambda(x) = 16x + 1.$$

The reciprocal error locator polynomial $\Lambda^*(x)$ is

$$\Lambda^*(x) = x + 16.$$

The chart of the left is generated following Chien search for the error locator polynomial $\Lambda(x)$

- In the row $i = 0$ we write the coefficients of $\Lambda(x)$: 1 and 16.
- In the row $i = 248$ goes 1, and $16 \cdot 2^{248} = 16 \cdot 27 = 173$.
- Every value in the column $g_{0,i}$ is 1.
- Fill the missing in the column $g_{1,i}$ with twice the value above, so complete with the values 71, 142, 1, 2, 4 and 8.
- The last column $\Lambda(2^i)$ has the sum of the values in the columns $g_{0,i}$ and $g_{1,i}$. In particular,

$$\Lambda(2^{251}) = 1 + 1 = 0.$$

The table of the right is filled by Chien search for the reciprocal error locator polynomial $\Lambda^*(x)$

- In the row $i = 0$ we write the coefficients of $\Lambda^*(x)$: 16 and 1.
- Every value in the column $g_{0,i}$ is 16.
- The cells in the column $g_{1,i}$ are filled with twice the value above, so its values are 1, 2, 4, 8, 16, 32, 64 and 128.
- The last column $\Lambda^*(2^i)$ has the sum of the values in the columns $g_{0,i}$ and $g_{1,i}$. In particular,

$$\Lambda^*(2^4) = 16 + 16 = 0.$$

i	2^i	$g_{0,i}$	$g_{1,i}$	$\Lambda(2^i)$
0	1	1	16	17
248	27	1	173	172
249	54	1	71	70
250	108	1	142	143
251	216	1	1	0
252	173	1	2	3
253	71	1	4	5
254	142	1	8	9

i	2^i	$g_{0,i}$	$g_{1,i}$	$\Lambda^*(2^i)$
0	1	16	1	17
1	2	16	2	18
2	4	16	4	20
3	8	16	8	24
4	16	16	16	0
5	32	16	32	48
6	64	16	64	70
7	128	16	128	144

There is only one root of $\Lambda(x)$: $216 = 16^{-1}$; and only one root of $\Lambda^*(x)$, which is $2^4 = 16$.
Thus, the error position is $\log_2(16) = \log_2(2^4) = 4$.

Advice: To find the error positions faster, use Chien search for the reciprocal error locator polynomial. With this strategy, you can predict the values in the column $g_{2,i}$, since $\Lambda_0 = 1$ for any error locator polynomial; so you don't need to calculate the values in that column.

5 Finding Error Values

Recall that the error locator polynomial is

$$\Lambda(x) = \prod_{j=1}^{v}(1 - X_j x) = \Lambda_v x^v + \Lambda_{v-1} x^{v-1} + \ldots + \Lambda_1 x + \Lambda_0 = 0.$$

The process of finding the error polynomial $e(x)$ requires finding the roots of $\Lambda(x)$ to locate the positions of the errors. Then, we must find the coefficients of $e(x)$ in the found positions. A suitable method is provided by the Forney Algorithm.

Remark: Recall that we have $e(x) = \sum_{j=1}^{v} e_{k_j} x^{k_j}$ with $k_j = \log_2(X_j)$, see Section 3.1.

5.1 Forney Algorithm

For this algorithm, we must introduce some polynomials.
The partial syndromes polynomial:

$$S(x) = s_1 + s_2 x + s_3 x^2 + \ldots + s_{2t} x^{2t-1}$$

The formal derivative of $\Lambda(x)$:

$$\Lambda'(x) = \frac{\sum_{i \text{ odd}} \Lambda_i x^i}{x} = \Lambda_1 + \Lambda_3 x^2 + \Lambda_5 x^4 \ldots$$

Caution: Our formula for the derivative applies to $GF(256)$ and differs from the one used in a Basic Calculus course!

The error evaluator polynomial:

$$\Omega(x) = S(x)\Lambda(x) \mod x^{2t},$$

where $\mod x^{2t}$ means that any power of x higher or equal than $2t$ can be replaced by a zero.
Then the error values are:

$$e_{k_j} = \frac{X_j \cdot \Omega(X_j^{-1})}{\Lambda'(X_j^{-1})} \tag{6}$$

5.1.1 Example 1

Last section, we found that there are errors in the $0th$ and $1st$ positions. Now we will find the error values. We have $t = 2$ and error locator polynomial $\Lambda(x) = 2x^2 + 3x + 1 = (1 + 2x)(1 + x)$.
The syndromes are

$$s_1 = 0, \quad s_2 = 3, \quad s_3 = 5, \quad \text{and } s_4 = 9.$$

The partial syndromes polynomial is:

$$S(x) = 3x + 5x^2 + 9x^3$$

The formal derivative is:

$$\Lambda'(x) = \left(2x^2 + 3x + 1\right)' = \frac{3x}{x} = 3$$

The error evaluator polynomial is:

$$\begin{aligned}
\Omega(x) &= (3x + 5x^2 + 9x^3)(2x^2 + 3x + 1) \mod x^4 \\
&= (6x^3 + 10x^4 + 18x^5) + (5x^2 + 15x^3 + 27x^4) + (3x + 5x^2 + 9x^3) \mod x^4 \\
&= 18x^5 + (10 + 27)x^4 + (6 + 15 + 9)x^3 + (5 + 5)x^2 + 3x \mod x^4 \\
&= 18x^5 + 17x^4 + 3x \mod x^4 \\
&= 3x
\end{aligned}$$

To obtain the error values, we use formula (6) with $X_1 = 1$ and $X_2 = 2$.
For $X_1 = 1$, we have $X_1^{-1} = 1$; so,

$$e_{k_1} = \frac{X_1 \cdot \Omega(X_1^{-1})}{\Lambda'(X_1^{-1})} = \frac{1 \cdot \Omega(1)}{\Lambda'(1)} = \frac{1 \cdot (3 \cdot 1)}{3} = \frac{3}{3} = 1.$$

For $X_2 = 2$, we have $X_2^{-1} = 142$; so,

$$e_{k_2} = \frac{X_1 \cdot \Omega(X_1^{-1})}{\Lambda'(X_1^{-1})} = \frac{2 \cdot \Omega(142)}{\Lambda'(142)} = \frac{2 \cdot (3 \cdot 142)}{3} = \frac{3}{3} = 1.$$

Thus, the error polynomial is $e(x) = x + 1$.

5.1.2 Example 2

Last section, we found that there are errors in the $6th$ and $7th$ positions. Now we will find the error values. We have $t = 2$ and error locator polynomial $\Lambda(x) = 135x^2 + 192x + 1 = (1 + 64x)(1 + 128x)$.
The syndromes are

$$s_1 = 0, \quad s_2 = 192, \quad s_3 = 222, \quad \text{and } s_4 = 88.$$

The partial syndromes polynomial is:

$$S(x) = 192x + 222x^2 + 88x^3$$

The formal derivative is:

$$\Lambda'(x) = \left(135x^2 + 192x + 1\right)' = \frac{192x}{x} = 192$$

The error evaluator polynomial is:

$$\Omega(x) = (192x + 222x^2 + 88x^3)(135x^2 + 192x + 1) \mod x^4$$

$$
\begin{aligned}
&= (238x^3 + 15x^4 + 142x^5) + (222x^2 + 182x^3 + 152x^4) + (192x + 222x^2 + 88x^3) \mod x^4 \\
&= 142x^5 + (15 + 152)x^4 + (238 + 182 + 88)x^3 + (222 + 222)x^2 + 192x \mod x^4 \\
&= 142x^5 + 151x^4 + 142x \mod x^4 \\
&= 192x
\end{aligned}
$$

To obtain the error values, we use formula (6) with $X_1 = 64$ and $X_2 = 128$.
For $X_1 = 64$, we have $X_1^{-1} = 54$; so,

$$e_{k_1} = \frac{X_1 \cdot \Omega(X_1^{-1})}{\Lambda'(X_1^{-1})} = \frac{64 \cdot \Omega(54)}{\Lambda'(54)} = \frac{64 \cdot (192 \cdot 54)}{192} = \frac{192}{192} = 1.$$

For $X_2 = 128$, we have $X_2^{-1} = 27$; so,

$$e_{k_2} = \frac{X_1 \cdot \Omega(X_1^{-1})}{\Lambda'(X_1^{-1})} = \frac{128 \cdot \Omega(27)}{\Lambda'(27)} = \frac{128 \cdot (192 \cdot 27)}{192} = \frac{192}{192} = 1.$$

Thus, the error polynomial is $e(x) = x^7 + x^6$.

5.1.3 Example 3

Last section, we found that there are errors in the $3rd$ and $4th$ positions. Now we will find the error values.
We have $t = 2$ and error locator polynomial $\Lambda(x) = 128x^2 + 24x + 1 = (1 + 16x)(1 + 8x)$.
The syndromes are

$$s_1 = 0, \quad s_2 = 24, \quad s_3 = 93, \quad \text{and } s_4 = 247.$$

The partial syndromes polynomial is:

$$S(x) = 24x + 93x^2 + 247x^3$$

The formal derivative is:

$$\Lambda'(x) = \left(128x^2 + 24x + 1\right)' = \frac{24x}{x} = 24$$

The error evaluator polynomial is:

$$
\begin{aligned}
\Omega(x) &= (24x + 93x^2 + 247x^3)(128x^2 + 24x + 1) \mod x^4 \\
&= (156x^3 + 161x^4 + 22x^5) + (93x^2 + 107x^3 + 32x^4) + (24x + 93x^2 + 247x^3) \mod x^4 \\
&= 22x^5 + (161 + 32)x^4 + (156 + 107 + 247)x^3 + (93 + 93)x^2 + 24x \mod x^4 \\
&= 22x^5 + 129x^4 + 24x \mod x^4 \\
&= 24x
\end{aligned}
$$

To obtain the error values, we use formula (6) with $X_1 = 8$ and $X_2 = 16$.
For $X_1 = 8$, we have $X_1^{-1} = 173$; so,

$$e_{k_1} = \frac{X_1 \cdot \Omega(X_1^{-1})}{\Lambda'(X_1^{-1})} = \frac{8 \cdot \Omega(173)}{\Lambda'(173)} = \frac{8 \cdot (24 \cdot 173)}{24} = \frac{24}{24} = 1.$$

For $X_2 = 16$, we have $X_2^{-1} = 216$; so,

$$e_{k_2} = \frac{X_1 \cdot \Omega(X_1^{-1})}{\Lambda'(X_1^{-1})} = \frac{16 \cdot \Omega(216)}{\Lambda'(216)} = \frac{16 \cdot (24 \cdot 216)}{24} = \frac{24}{24} = 1.$$

Thus, the error polynomial is $e(x) = x^4 + x^3$.

5.1.4 Example 4

Last section, we found that there is an error in the $4th$ position. Now we will find the error value.
We have $t = 2$ and error locator polynomial $\Lambda(x) = 1 + 16x$.
The syndromes are

$$s_1 = 1, \quad s_2 = 16, \quad s_3 = 29, \quad \text{and } s_4 = 205.$$

The partial syndromes polynomial is:

$$S(x) = 1 + 16x + 29x^2 + 205x^3$$

The formal derivative is:

$$\Lambda'(x) = (16x + 1)' = \frac{16x}{x} = 16$$

The error evaluator polynomial is:

$$\begin{aligned}
\Omega(x) &= (1 + 16x + 29x^2 + 205x^3)(1 + 16x) \mod x^4 \\
&= (1 + 16x + 29x^2 + 205x^3) + (16x + 29x^2 + 205x^3 + 76x^4) \mod x^4 \\
&= 76x^4 + (205 + 205)x^3 + (29 + 29)x^2 + (16 + 16)x + 1 \mod x^4 \\
&= 76x^4 + 1 \mod x^4 \\
&= 1
\end{aligned}$$

To obtain the error value, we use formula (6) with $X_1 = 16$. We have $X_1^{-1} = 216$; so,

$$e_{k_1} = \frac{X_1 \cdot \Omega(X_1^{-1})}{\Lambda'(X_1^{-1})} = \frac{16 \cdot \Omega(216)}{\Lambda'(216)} = \frac{16 \cdot 1}{16} = \frac{16}{16} = 1.$$

Thus, the error polynomial is $e(x) = x^4$.