Guides (/guides/) / QR Code Tutorial (/qr-code-tutorial/) / Error Correction Coding

Error Correction Coding

Error correction codewords allow QR code readers to detect and correct errors in QR codes. This page explains how to create these error correction codewords after encoding the data (data-encoding).

Step 1: Break Data Codewords into Blocks if Necessary

Before generating error correction codewords, it may be necessary to break up the data codewords into smaller blocks if the QR code is larger than version 2. As an example, if creating a 5-Q QR code, the error correction table (error-correction-table) says that a 5-Q code has 62 data codewords (strings of 8-bit binary numbers). For example:

(codeword #1) 01000011
(codeword #2) 01010101
(codeword #3) 01000110
(codeword #4) 10000110
(codeword #5) 01010111
(codeword #6) 00100110
(codeword #7) 01010101
(codeword #8) 11000010
(codeword #9) 01110111
(codeword #10) 00110010
(codeword #11) 00000110
(codeword #12) 00010010
(codeword #13) 00000110
(codeword #14) 01100111
(codeword #15) 00100110
(codeword #16) 11110110
(codeword #17) 11110110
(codeword #18) 01000010
(codeword #19) 00000111
(codeword #20) 01110110
(codeword #21) 10000110
(codeword #22) 11110010
(codeword #23) 00000111
(codeword #24) 00100110
(codeword #25) 01010110
(codeword #26) 00010110
(codeword #27) 11000110
(codeword #28) 11000111
(codeword #29) 10010010
(codeword #30) 00000110
(codeword #31) 10110110
(codeword #32) 11100110

(codeword #33) 11110111
(codeword #34) 01110111
(codeword #35) 00110010
(codeword #36) 00000111
(codeword #37) 01110110
(codeword #38) 10000110
(codeword #39) 01010111
(codeword #40) 00100110
(codeword #41) 01010010
(codeword #42) 00000110
(codeword #43) 10000110
(codeword #44) 10010111
(codeword #45) 00110010
(codeword #46) 00000111
(codeword #47) 01000110
(codeword #48) 11110111
(codeword #49) 01110110
(codeword #50) 01010110
(codeword #51) 11000010
(codeword #52) 00000110
(codeword #53) 10010111
(codeword #54) 00110010
(codeword #55) 11100000
(codeword #56) 11101100
(codeword #57) 00010001
(codeword #58) 11101100
(codeword #59) 00010001
(codeword #60) 11101100
(codeword #61) 00010001
(codeword #62) 11101100

The error correction table (error-correction-table) mentions "group 1" and "group 2", as well as "number of blocks." This means that the data codewords must be split up into up to two groups, and within each group, the data codewords may be further broken up into blocks. The data codewords are broken up sequentially (i.e. starting with codeword 1, then codeword 2, and so on.)

For a 5-Q code, it says that there are two groups, the first of which should be broken up into 2 blocks that each contain 15 data codewords, and the second of which should be broken up into 2 blocks that each contain 16 data codewords. Notice that 15+15+16+16 = 62, which is the total number of data codewords. Below, are the codewords from above, broken up into the correct groups and blocks to illustrate how to do this step.

Group Number	**Block Number**	**Data Codewords in the Group**
Group 1	Block 1	(codeword #1) 01000011 (codeword #2) 01010101 (codeword #3) 01000110 (codeword #4) 10000110 (codeword #5) 01010111 (codeword #6) 00100110

		(codeword #7) 01010101 (codeword #8) 11000010 (codeword #9) 01110111 (codeword #10) 00110010 (codeword #11) 00000110 (codeword #12) 00010010 (codeword #13) 00000110 (codeword #14) 01100111 (codeword #15) 00100110
	Block 2	(codeword #16) 11110110 (codeword #17) 11110110 (codeword #18) 01000010 (codeword #19) 00000111 (codeword #20) 01110110 (codeword #21) 10000110 (codeword #22) 11110010 (codeword #23) 00000111 (codeword #24) 00100110 (codeword #25) 01010110 (codeword #26) 00010110 (codeword #27) 11000110 (codeword #28) 11000111 (codeword #29) 10010010 (codeword #30) 00000110
Group 2	Block 1	(codeword #31) 10110110 (codeword #32) 11100110 (codeword #33) 11110111 (codeword #34) 01110111 (codeword #35) 00110010 (codeword #36) 00000111 (codeword #37) 01110110 (codeword #38) 10000110 (codeword #39) 01010111 (codeword #40) 00100110 (codeword #41) 01010010 (codeword #42) 00000110 (codeword #43) 10000110 (codeword #44) 10010111 (codeword #45) 00110010 (codeword #46) 00000111
	Block 2	(codeword #47) 01000110 (codeword #48) 11110111 (codeword #49) 01110110 (codeword #50) 01010110 (codeword #51) 11000010

		(codeword #52) 00000110 (codeword #53) 10010111 (codeword #54) 00110010 (codeword #55) 11100000 (codeword #56) 11101100 (codeword #57) 00010001 (codeword #58) 11101100 (codeword #59) 00010001 (codeword #60) 11101100 (codeword #61) 00010001 (codeword #62) 11101100

In the table, notice that the blocks in group 1 consist of 15 data codewords, and the blocks in group 2 consist of 16 data codewords as specified in the error correction table (error-correction-table). Note also that they are broken up sequentially. That is, they are in the same order as they were before being separated into blocks.

The error correction table (error-correction-table) also says that for a 5-Q code, there are 18 error correction codewords per block. In this example, there are four blocks, so there will be four sets of 18 error correction codewords, for a total of 72 error correction codewords.

As shown in the error correction table (error-correction-table), the smaller QR codes do not require the data codewords to be broken up at all, so, for a 1-M code (for example) all 16 data codewords would be used as a single block, and only 10 error correction codewords would need to be generated.

Before continuing the 5-Q example, it is necessary to explain the underlying steps of Reed-Solomon Error correction.

Step 2: Understand Polynomial Long Division

The error correction codewords will be generated using a method called Reed-Solomon Error correction. Part of the process is to perform polynomial long division. That is, dividing one polynomial by another polynomial. This tutorial assumes an understanding of how to do standard long division (i.e. with numbers rather than polyomials) by hand. For more information, read wikiHow's article about how to do long division (http://www.wikihow.com/Do-Long-Division).

Polynomial long division is slightly more complicated than standard long division. To help clarify the process, here is a simple example of polynomial long division.

This example show **$3x^2$ + x - 1** (the dividend) divided by **x + 1** (the divisor). These polynomials are first put into a tableau (see Wikipedia's article on long division (http://en.wikipedia.org/wiki/Long_division#Method) for an explanation of this notation).

$$x + 1 \;\big|\overline{3x^2 + \;\; x - 1}$$

In each step of the long division, multiply **x + 1** by something to make its first term equal that of the polynomial at the bottom of the tableau.

For the first step, **$3x^2$ + x - 1** is at the bottom of the tableau, so multiply **x + 1** by **3x**. This results in **$3x^2$ + 3x**, which has the same first term as that of the polynomial at the bottom of the tableau. Here is the updated tableau.

$$\begin{array}{rl}
 & \underline{3x \qquad\qquad\qquad} \\
x + 1 & |\,3x^2 + \;\; x - 1 \\
 & 3x^2 + 3x
\end{array}$$

Now subtract **$3x^2$ + 3x** from **$3x^2$ + x - 1**. This results in **-2x - 1** (because **x - 3x** is **(1 - 3)x**, or **-2x**). The -1 at the end is carried over from the original polynomial. Here is the updated tableau.

$$\begin{array}{rl}
 & \underline{3x \qquad\qquad\qquad} \\
x + 1 & |\,3x^2 + \;\; x - 1 \\
 & \underline{3x^2 + 3x} \\
 & \qquad - 2x - 1
\end{array}$$

Now, once again, multiply **x + 1** by a term that will result in a polynomial whose first term is the same as that of the bottom polynomial. In this case, do **-2 (x + 1)** to get **-2x - 2**. Here is the updated tableau.

$$\begin{array}{rl}
 & \underline{3x - 2 \qquad} \\
x + 1 & |\,3x^2 + \;\; x - 1 \\
 & \underline{3x^2 + 3x} \\
 & \qquad - 2x - 1 \\
 & \qquad - 2x - 2
\end{array}$$

And now, once again, subtract the two polynomials. The -2x cancels out, but this leaves -1 - 2, which equals 1, as shown in the final tableau below.

$$\begin{array}{rl}
 & \underline{3x - 2 \qquad} \\
x + 1 & |\,3x^2 + \;\; x - 1 \\
 & \underline{3x^2 + 3x} \\
 & \qquad - 2x - 1 \\
 & \qquad \underline{- 2x - 2} \\
 & \qquad\qquad\qquad 1
\end{array}$$

The remainder is 1. Technically it is $1x^0$. It would be necessary to multiply (x + 1) by 1/x to make the first term the same, but since this is a fraction, it means means that no further division is possible. The 1 at the bottom of the tableau is the remainder.

Overall, the steps of polynomial long division are:

1. Find the appropriate term to multiply the divisor by. The result of the multiplication should have the same first term as the the dividend (in the first multiplication step) or remainder (in all subsequent multiplication steps).
2. Subtract the result from the dividend (in the first multiplication step) or remainder (in all subsequent multiplication steps).
3. Repeat steps 1 and 2 until it is no longer possible to multiply by an integer, or in other words, it would be necessary to multiply by a fraction. The number at the bottom of the tableau is the remainder.

The polynomial long division necessary for Reed-Solomon Error Correction is simpler in some ways than this example, because it will not be necessary to deal with the exponents of the polynomial terms.

Step 3: Understand The Galois Field

As mentioned in the previous section, to generate error correction codewords, this process uses a method called Reed-Solomon Error correction. Along with polynomial long division, this method uses a Galois field, which is essentially a restricted set of numbers, as well as some mathematical operations that create numbers that are still in that set.

The QR Code standard says to use bit-wise modulo 2 arithmetic and byte-wise modulo 100011101 arithmetic. This means using Galois Field 2^8, or in other words Galois Field 256, sometimes written as GF(256).

The numbers in GF(256) will all be in the range of 0 through 255 (inclusive). Notice that this is the same range of numbers that can be represented with an eight-bit byte (the largest possible eight-bit byte is 11111111, which equals 255).

This means that all of the mathematical operations in GF(256) will result in numbers that can be represented as eight-bit bytes.

Step 4: Understand Galois Field Arithmetic

As mentioned previously, GF(256) contains the numbers 0 through 255 inclusive. Mathematical operations in GF(256) are cyclical in nature, meaning that if a mathematical operation is performed within GF(256) that results in a number larger than 255, it will be necessary to use the modulo operation to get a number that is still in the Galois Field.

In the Galois Field, negative numbers have the same value as positive numbers, so **-n = n**. In other words, always use the absolute value of the numbers involved in Galois Field arithmetic.

This means that addition and subtraction within the Galois Field are the same thing. Addition and subtraction in the Galois Field are performed by doing the addition or subtraction as normal, but then performing the modulo operation. And since we are using bit-wise modulo 2 arithmetic (as mentioned in the QR code specification), this is the same as performing the XOR operation. For example:
1 + 1 = 2 % 2 = 0
or
1 ^ 1 = 0

and

0 + 1 = 1 % 2 = 1
or
0 ^ 1 = 1

For the purposes of encoding a QR code, all addition and subtraction in GF(256) is performed by XORing the two numbers together.

Step 5: Generate Powers of 2 using Byte-Wise Modulo 100011101

All of the numbers in GF(256) can be represented as a power of 2. Specifically, all numbers in GF(256) can be represented as 2^n, where 0 <= n <= 255. However, as stated before, all numbers in GF(256) must be in the range of 0 through 255 themselves, so 2^8 would seem to be too big for the Galois Field since it is equal to 256.

The powers of 2 from 0 to 8 are:
$2^0 = 1$
$2^1 = 2$
$2^2 = 4$
$2^3 = 8$
$2^4 = 16$
$2^5 = 32$
$2^6 = 64$
$2^7 = 128$
$2^8 = 256$

The QR code specification says to use byte-wise modulo 100011101 arithmetic (where 100011101 is a binary number that is equivalent to 285 in decimal). This means that when a number is 256 or larger, it should be XORed with 285. This leads to unexpected values for 2^8 and larger.

In other words:
$2^8 = 256 \wedge 285 = 29$

Note that when continuing on to 2^9, do not take its usual value of 512 and XOR with 285 (which would result in too large a number anyway). Instead, since $2^9 = 2^8 * 2$, use the value of 2^8 that was calculated in the previous step.

In other words:
$2^9 = 2^8 * 2 = 29 * 2 = 58$

Continue using the previous power of 2 to create the next powers of 2:
$2^{10} = 2^9 * 2 = 58 * 2 = 116$
$2^{11} = 2^{10} * 2 = 116 * 2 = 232$

Whenever a value greater than or equal to 256 is obtained, once again XOR with 285:
$2^{12} = 2^{11} * 2 = 232 * 2 = 464 \wedge 285 = 205$

In general, the values are always equal to 2 times the previous power, and if that value is 256 or greater, it is XORed with 285.

Using this procedure, all numbers in GF(256) can be represented with 2^n, where n is a number in the range 0 <= n <= 255. Using byte-wise modulo 100011101 ensures that all values are within the 0 through 255 range.

Step 6: Understand Multiplication with Logs and Antilogs

Because all of the values can be represented with 2^n as explained above, it is possible to logs and antilogs to simplify multiplication in GF(256). (Incidentally, this is what slide rules (http://en.wikipedia.org/wiki/Slide_rule) do, although they are not restricted to GF(256)).

The simplification is possible because, in general (i.e. not just in Galois Fields), one can multiply two numbers p and q with the following operation:
$b^{(\log_b p + \log_b q)}$

In this case, base 2 is in use, so the above operation becomes
$2^{(\log_2 p + \log_2 q)}$
And the $\log_2 x$ operation tells us how many times to multiply 2 to get an answer of x. In other words, it determines n where $2^n = x$.

This provides a shortcut for multiplying numbers in GF(256). When multiplying two numbers that have the same base, such as $2^2 * 2^8$, this is equivalent to adding the two exponents together, like so:

$2^2 * 2^8 = 2^{(2 + 8)} = 2^{10}$

This means that since all GF(256) values have been expressed as powers of two (explained in the previous section), all multiplication in GF(256) can be performed as addition of base-2 exponents.

For example, to multiply 16 * 32, this is the same as multiplying 2^4 by 2^5. As the previous paragraph explained, this is the same as $2^{(4 + 5)}$ or 2^9. The value of 2^9 was already calculated, therefore in GF(256) with byte-wise modulo 285, 16 * 32 is 2^9, which was determined in the previous section to be equal to 58.

When adding exponents, if the exponent becomes greater than or equal to 256, simply apply modulo 255. In other words:

$2^{170} * 2^{164} = 2^{(170+164)} = 2^{334} \rightarrow 2^{334 \% 255} = 2^{79}$.

Therefore, all that is needed to perform multiplication in GF(256) is to generate all the powers of 2. These values have been calculated already and can be found in the log antilog table (log-antilog-table). The table uses alpha notation, where α = 2. The QR Code specification also uses alpha notation in Annex A.

Step 7: Understanding The Generator Polynomial

We have made a lot of progress toward generating error correction codewords, but we are not quite there yet. The next step is understanding generator polynomials.

As mentioned earlier, error correction coding uses polynomial long division. To do that, two polynomials are needed. The first polynomial to use is called the **message polynomial**. The message polynomial uses the data codewords from the data encoding (data-encoding) step as its coefficients. For example, if the data codewords, converted to integers, were 25, 218, and 35, the message polynomial would be $25x^2 + 218x + 35$. In practice, real message polynomials for standard QR codes are much longer, but this is just an example.

The message polynomial will be divided by a **generator polynomial**. The generator polynomial is a polynomial that is created by multiplying
$(x - \alpha^0) \ldots (x - \alpha^{n-1})$
where n is the number of error correction codewords that must be generated (see the error correction table (error-correction-table)). As mentioned in the previous section, α (alpha) is equal to 2.

It is possible to programmatically create all the generator polynomials that might be needed. During the process, this tutorial will make use of alpha notation and the log antilog (log-antilog-table) multiplication that was described earlier on this page.

The QR code specification lists generator polynomials in Annex A, starting with 2 and ending with 68. Although standard QR codes will always require more than 2 error correction codewords per block, this page will show how to calculate the generator polynomial for 2 error correction codewords, because it illustrates the process of calculating all the rest of the generator polynomials as well.

Generator Polynomial for 2 Error Correction Codewords

First, multiply (x - α^0) and (x - α^1).
Since the coefficient of x is 1, and since $\alpha^0 = 1$, this can be written as
$(\alpha^0x - \alpha^0) * (\alpha^0x - \alpha^1)$

Multiply each term of the first part by each term of the second part to get this:
$(\alpha^0x^1 * \alpha^0x^1) + (\alpha^0x^0 * \alpha^0x^1) + (\alpha^0x^1 * \alpha^1x^0) + (\alpha^0x^0 * \alpha^1x^0)$

Notice that exponent addition can be used here to perform the multiplications:
$(\alpha^{(0+0)}x^{(1+1)}) + (\alpha^{(0+0)}x^{(0+1)}) + (\alpha^{(0+1)}x^{(1+0)}) + (\alpha^{(0+1)}x^{(0+0)})$

The result:
$\alpha^0x^2 + \alpha^0x^1 + \alpha^1x^1 + \alpha^1x^0$

Now, combine like terms. There are two x^1 terms, so add them together.
$\alpha^0x^2 + (\alpha^0+\alpha^1)x^1 + \alpha^1x^0$

Remember that addition in GF(256) is performed by XOR. Convert the alphas to their integer counterparts using the log antilog table (log-antilog-table), then perform the XOR.

$x^2 + (1\text{^}2)x^1 + 2x^0$
$x^2 + 3x^1 + 2x^0$

Back in alpha notation, this is
$\alpha^0x^2 + \alpha^{25}x^1 + \alpha^1x^0$
This is the generator polynomial for 2 error correction codewords.

Generator Polynomial for 3 Error Correction Codewords

All the rest of the generator polynomials can be created the same way, using the generator polynomial from each previous step. Start with the previous generator polynomial and multiply it by the next factor, in this case (x - α^2)

$(\alpha^0x^2 + \alpha^{25}x^1 + \alpha^1x^0) * (\alpha^0x^1 + \alpha^2x^0)$

$(\alpha^0x^2 * \alpha^0x^1) + (\alpha^{25}x^1 * \alpha^0x^1) + (\alpha^1x^0 * \alpha^0x^1) + (\alpha^0x^2 * \alpha^2x^0) + (\alpha^{25}x^1 * \alpha^2x^0) + (\alpha^1x^0 * \alpha^2x^0)$

To multiply, add the exponents together like this:
$(\alpha^{(0+0)}x^{(2+1)}) + (\alpha^{(25+0)}x^{(1+1)}) + (\alpha^{(1+0)}x^{(0+1)}) + (\alpha^{(0+2)}x^{(2+0)}) + (\alpha^{(25+2)}x^{(1+0)}) + (\alpha^{(1+2)}x^{(0+0)})$

After adding the exponents, this is the result:
$\alpha^0x^3 + \alpha^{25}x^2 + \alpha^1x^1 + \alpha^2x^2 + \alpha^{27}x^1 + \alpha^3x^0$

Now combine like terms.
$\alpha^0x^3 + (\alpha^{25} \text{ ^ } \alpha^2)x^2 + (\alpha^1 \text{ ^ } \alpha^{27})x^1 + \alpha^3x^0$

$\alpha^0x^3 + (3 \text{ ^ } 4)x^2 + (2 \text{ ^ } 12)x^1 + \alpha^3x^0$
$\alpha^0x^3 + 7x^2 + 14x^1 + \alpha^3x^0$
$\alpha^0x^3 + \alpha^{198}x^2 + \alpha^{199}x^1 + \alpha^3x^0$

This is the generator polynomial for 3 error correction codewords.

When the Exponents are Greater than or Equal to 256

When performing multiplication by adding exponents together, sometimes this may result in an exponent that is greater than or equal to 256. In this case, apply modulo 255 BEFORE combining like terms. For example
$\alpha^{257}x^4 = \alpha^{(257\ \%\ 255)}x^4 = \alpha^2x^4$

Other Generator Polynomials

In general, it is possible to create all the generator polynomials g(x) as illustrated below:
(g(x) for 2 error correction codewords) * (x - α^2) = (g(x) for 3 error correction codewords)
(g(x) for 3 error correction codewords) * (x - α^3) = (g(x) for 4 error correction codewords)
(g(x) for 4 error correction codewords) * (x - α^4) = (g(x) for 5 error correction codewords)
...
(g(x) for n-1 error correction codewords) * (x - α^{n-1}) = (g(x) for n error correction codewords)

To obtain a specific generator polynomial without having to calculate it programmatically, the generator polynomial tool (generator-polynomial-tool) can create it.

Step 8: Generating Error Correction Codewords

Now it is finally time to start generating error correction codewords. This part uses the data codewords from the HELLO WORLD example on the alphanumeric mode encoding (alphanumeric-mode-encoding) page. These codewords will be used as the coefficients of the message polynomial. This example uses error correction level M, and version 1, to create a 1-M code.

The Message Polynomial

The data encoding step resulted in the following data codewords for HELLO WORLD as a 1-M code.
00100000 01011011 00001011 01111000 11010001 01110010 11011100 01001101 01000011 01000000 11101100 00010001 11101100 00010001 11101100 00010001

Convert those binary numbers into decimal:
32, 91, 11, 120, 209, 114, 220, 77, 67, 64, 236, 17, 236, 17, 236, 17

These numbers are the coefficients of the **message polynomial**. In other words:
$32x^{15} + 91x^{14} + 11x^{13}$ ... and so on

The Generator Polynomial

First, get the generator polynomial. Since this is a 1-M code, the error correction table (error-correction-table) says to create 10 error correction codewords. Therefore, use the following generator polynomial:
$x^{10} + \alpha^{251}x^9 + \alpha^{67}x^8 + \alpha^{46}x^7 + \alpha^{61}x^6 + \alpha^{118}x^5 + \alpha^{70}x^4 + \alpha^{64}x^3 + \alpha^{94}x^2 + \alpha^{32}x + \alpha^{45}$

(Refer to the generator polynomial tool (generator-polynomial-tool) and the discussion about generator polynomials earlier on this page.)

The Division Steps

Now it is time to divide the message polynomial by the generator polynomial. This is done in much the same way as the polynomial long division (error-correction-coding#polynomial-long-division) shown earlier on the page. Here are the division steps, updated to take Galois Field arithmetic into account:

1. Find the appropriate term to multiply the generator polynomial by. The result of the multiplication should have the same first term as the the message polynomial (in the first multiplication step) or remainder (in all subsequent multiplication steps).
2. XOR the result with the message polynomial (in the first multiplication step) or remainder (in all subsequent multiplication steps).

3. Perform these steps n times, where n is the number of data codewords.

Notice the differences between these steps and those of normal polynomial long division. Instead of subtracting after the multiplication step, we perform an XOR (which, in GF(256), is the same thing).

More importantly, after dividing the two polynomials, there will be a remainder. The coefficients of this remainder are the error correction codewords.

The division will be illustrated step-by-step in the next section.

Step 9: Divide the Message Polynomial by the Generator Polynomial

The first step to the division is to prepare the message polynomial for the division. The full message polynomial is:

$32x^{15} + 91x^{14} + 11x^{13} + 120x^{12} + 209x^{11} + 114x^{10} + 220x^{9} + 77x^{8} + 67x^{7} + 64x^{6} + 236x^{5} + 17x^{4} + 236x^{3} + 17x^{2} + 236x^{1} + 17$

To make sure that the exponent of the lead term doesn't become too small during the division, multiply the message polynomial by x^n where n is the number of error correction codewords that are needed. In this case n is 10, for 10 error correction codewords, so multiply the message polynomial by x^{10}, which gives us:

$32x^{25} + 91x^{24} + 11x^{23} + 120x^{22} + 209x^{21} + 114x^{20} + 220x^{19} + 77x^{18} + 67x^{17} + 64x^{16} + 236x^{15} + 17x^{14} + 236x^{13} + 17x^{12} + 236x^{11} + 17x^{10}$

The lead term of the generator polynomial should also have the same exponent, so multiply by x^{15} to get

$\alpha^{0}x^{25} + \alpha^{251}x^{24} + \alpha^{67}x^{23} + \alpha^{46}x^{22} + \alpha^{61}x^{21} + \alpha^{118}x^{20} + \alpha^{70}x^{19} + \alpha^{64}x^{18} + \alpha^{94}x^{17} + \alpha^{32}x^{16} + \alpha^{45}x^{15}$

Now it is possible to perform the repeated division steps. The number of steps in the division must equal the number of terms in the message polynomial. In this case, the division will take 16 steps to complete. This will result in a remainder that has 10 terms. These terms will be the 10 error correction codewords that are required.

Step 1a: Multiply the Generator Polynomial by the Lead Term of the Message Polynomial

The first step is to multiply the generator polynomial by the lead term of the message polynomial. The lead term in this case is $32x^{25}$. Since alpha notation makes it easier to perform the multiplication, it is recommended to convert $32x^{25}$ to alpha notation. According to the log antilog table, for the integer value 32, the alpha exponent is 5. Therefore 32 = α^{5}. Multiply the generator polynomial by α^{5}:

The exponents of the alphas are added together. In this case, at least one of the exponents is larger than 255, so perform modulo 255 as follows:

$\alpha^{5}x^{25} + \alpha^{(256 \% 255)}x^{24} + \alpha^{72}x^{23} + \alpha^{51}x^{22} + \alpha^{66}x^{21} + \alpha^{123}x^{20} + \alpha^{75}x^{19} + \alpha^{69}x^{18} + \alpha^{99}x^{17} + \alpha^{37}x^{16} + \alpha^{50}x^{15}$

The result is:

$\alpha^{5}x^{25} + \alpha^{1}x^{24} + \alpha^{72}x^{23} + \alpha^{51}x^{22} + \alpha^{66}x^{21} + \alpha^{123}x^{20} + \alpha^{75}x^{19} + \alpha^{69}x^{18} + \alpha^{99}x^{17} + \alpha^{37}x^{16} + \alpha^{50}x^{15}$

Now, convert this to integer notation:

$32x^{25} + 2x^{24} + 101x^{23} + 10x^{22} + 97x^{21} + 197x^{20} + 15x^{19} + 47x^{18} + 134x^{17} + 74x^{16} + 5x^{15}$

Step 1b: XOR the result with the message polynomial

Since this is the first division step, XOR the result from 1a with the message polynomial.

$(32 \oplus 32)x^{25} + (91 \oplus 2)x^{24} + (11 \oplus 101)x^{23} + (120 \oplus 10)x^{22} + (209 \oplus 97)x^{21} + (114 \oplus 197)x^{20} + (220 \oplus 15)x^{19} + (77 \oplus 47)x^{18} + (67 \oplus 134)x^{17} + (64 \oplus 74)x^{16} + (236 \oplus 5)x^{15} + (17 \oplus 0)x^{14} + (236 \oplus 0)x^{13} + (17 \oplus 0)x^{12} + (236 \oplus 0)x^{11} + (17 \oplus 0)x^{10}$

The result is:

$0x^{25} + 89x^{24} + 110x^{23} + 114x^{22} + 176x^{21} + 183x^{20} + 211x^{19} + 98x^{18} + 197x^{17} + 10x^{16} + 233x^{15} + 17x^{14} + 236x^{13} + 17x^{12} + 236x^{11} + 17x^{10}$

Discard the lead 0 term to get:

$89x^{24} + 110x^{23} + 114x^{22} + 176x^{21} + 183x^{20} + 211x^{19} + 98x^{18} + 197x^{17} + 10x^{16} + 233x^{15} + 17x^{14} + 236x^{13} + 17x^{12} + 236x^{11} + 17x^{10}$

Step 2a: Multiply the Generator Polynomial by the Lead Term of the XOR result from the previous step

Next, multiply the generator polynomial by the lead term of the XOR result from the previous step. The lead term in this case is $89x^{24}$. Convert $89x^{24}$ to alpha notation. According to the log antilog table, for the integer value 89, the alpha exponent is 210. Therefore $89 = \alpha^{210}$. Multiply the generator polynomial by α^{210}:

$(\alpha^{210} * \alpha^{0})x^{24} + (\alpha^{210} * \alpha^{251})x^{23} + (\alpha^{210} * \alpha^{67})x^{22} + (\alpha^{210} * \alpha^{46})x^{21} + (\alpha^{210} * \alpha^{61})x^{20} + (\alpha^{210} * \alpha^{118})x^{19} + (\alpha^{210} * \alpha^{70})x^{18} + (\alpha^{210} * \alpha^{64})x^{17} + (\alpha^{210} * \alpha^{94})x^{16} + (\alpha^{210} * \alpha^{32})x^{15} + (\alpha^{210} * \alpha^{45})x^{14}$

The exponents of the alphas are added together. In this case, at least one of the exponents is larger than 255, so perform modulo 255 as follows:

$\alpha^{210}x^{24} + \alpha^{(461 \% 255)}x^{23} + \alpha^{(277 \% 255)}x^{22} + \alpha^{(256 \% 255)}x^{21} + \alpha^{(271 \% 255)}x^{20} + \alpha^{(328 \% 255)}x^{19} + \alpha^{(280 \% 255)}x^{18} + \alpha^{(274 \% 255)}x^{17} + \alpha^{(304 \% 255)}x^{16} + \alpha^{242}x^{15} + \alpha^{255}x^{14}$

The result is:

$\alpha^{210}x^{24} + \alpha^{206}x^{23} + \alpha^{22}x^{22} + \alpha^{1}x^{21} + \alpha^{16}x^{20} + \alpha^{73}x^{19} + \alpha^{25}x^{18} + \alpha^{19}x^{17} + \alpha^{49}x^{16} + \alpha^{242}x^{15} + \alpha^{0}x^{14}$

Now, convert this to integer notation:

$89x^{24} + 83x^{23} + 234x^{22} + 2x^{21} + 76x^{20} + 202x^{19} + 3x^{18} + 90x^{17} + 140x^{16} + 176x^{15} + 1x^{14}$

Step 2b: XOR the result with the result from step 1b

Use the result from step 1b to perform the next XOR.

$(89 \oplus 89)x^{24} + (110 \oplus 83)x^{23} + (114 \oplus 234)x^{22} + (176 \oplus 2)x^{21} + (183 \oplus 76)x^{20} + (211 \oplus 202)x^{19} + (98 \oplus 3)x^{18} + (197 \oplus 90)x^{17} + (10 \oplus 140)x^{16} + (233 \oplus 176)x^{15} + (17 \oplus 1)x^{14} + (236 \oplus 0)x^{13} + (17 \oplus 0)x^{12} + (236 \oplus 0)x^{11} + (17 \oplus 0)x^{10}$

The result is:

$0x^{24} + 61x^{23} + 152x^{22} + 178x^{21} + 251x^{20} + 25x^{19} + 97x^{18} + 159x^{17} + 134x^{16} + 89x^{15} + 16x^{14} + 236x^{13} + 17x^{12} + 236x^{11} + 17x^{10}$

Discard the lead 0 term to get:

$61x^{23} + 152x^{22} + 178x^{21} + 251x^{20} + 25x^{19} + 97x^{18} + 159x^{17} + 134x^{16} + 89x^{15} + 16x^{14} + 236x^{13} + 17x^{12} + 236x^{11} + 17x^{10}$

Step 3a: Multiply the Generator Polynomial by the Lead Term of the XOR result from the previous step

Next, multiply the generator polynomial by the lead term of the XOR result from the previous step. The lead term in this case is $61x^{23}$. Convert $61x^{23}$ to alpha notation. According to the log antilog table, for the integer value 61, the alpha exponent is 228. Therefore $61 = \alpha^{228}$. Multiply the generator polynomial by α^{228}:

$(\alpha^{228} * \alpha^{0})x^{23} + (\alpha^{228} * \alpha^{251})x^{22} + (\alpha^{228} * \alpha^{67})x^{21} + (\alpha^{228} * \alpha^{46})x^{20} + (\alpha^{228} * \alpha^{61})x^{19} + (\alpha^{228} * \alpha^{118})x^{18} + (\alpha^{228} * \alpha^{70})x^{17} + (\alpha^{228} * \alpha^{64})x^{16} + (\alpha^{228} * \alpha^{94})x^{15} + (\alpha^{228} * \alpha^{32})x^{14} + (\alpha^{228} * \alpha^{45})x^{13}$

The exponents of the alphas are added together. In this case, at least one of the exponents is larger than 255, so perform modulo 255 as follows:

$\alpha^{228}x^{23} + \alpha^{(479 \% 255)}x^{22} + \alpha^{(295 \% 255)}x^{21} + \alpha^{(274 \% 255)}x^{20} + \alpha^{(289 \% 255)}x^{19} + \alpha^{(346 \% 255)}x^{18} + \alpha^{(298 \% 255)}x^{17} + \alpha^{(292 \% 255)}x^{16} + \alpha^{(322 \% 255)}x^{15} + \alpha^{(260 \% 255)}x^{14} + \alpha^{(273 \% 255)}x^{13}$

The result is:

$\alpha^{228}x^{23} + \alpha^{224}x^{22} + \alpha^{40}x^{21} + \alpha^{19}x^{20} + \alpha^{34}x^{19} + \alpha^{91}x^{18} + \alpha^{43}x^{17} + \alpha^{37}x^{16} + \alpha^{67}x^{15} + \alpha^{5}x^{14} + \alpha^{18}x^{13}$

Now, convert this to integer notation:

$61x^{23} + 18x^{22} + 106x^{21} + 90x^{20} + 78x^{19} + 163x^{18} + 119x^{17} + 74x^{16} + 194x^{15} + 32x^{14} + 45x^{13}$

Step 3b: XOR the result with the result from step 2b

Use the result from step 2b to perform the next XOR.

$(61 \oplus 61)x^{23} + (152 \oplus 18)x^{22} + (178 \oplus 106)x^{21} + (251 \oplus 90)x^{20} + (25 \oplus 78)x^{19} + (97 \oplus 163)x^{18} + (159 \oplus 119)x^{17} + (134 \oplus 74)x^{16} + (89 \oplus 194)x^{15} + (16 \oplus 32)x^{14} + (236 \oplus 45)x^{13} + (17 \oplus 0)x^{12} + (236 \oplus 0)x^{11} + (17 \oplus 0)x^{10}$

The result is:

$0x^{23} + 138x^{22} + 216x^{21} + 161x^{20} + 87x^{19} + 194x^{18} + 232x^{17} + 204x^{16} + 155x^{15} + 48x^{14} + 193x^{13} + 17x^{12} + 236x^{11} + 17x^{10}$

Discard the lead 0 term to get:

$138x^{22} + 216x^{21} + 161x^{20} + 87x^{19} + 194x^{18} + 232x^{17} + 204x^{16} + 155x^{15} + 48x^{14} + 193x^{13} + 17x^{12} + 236x^{11} + 17x^{10}$

Step 4a: Multiply the Generator Polynomial by the Lead Term of the XOR result from the previous step

Next, multiply the generator polynomial by the lead term of the XOR result from the previous step. The lead term in this case is $138x^{22}$. Convert $138x^{22}$ to alpha notation. According to the log antilog table, for the integer value 138, the alpha exponent is 222. Therefore $138 = \alpha^{222}$. Multiply the generator polynomial by α^{222}:

$(\alpha^{222} * \alpha^{0})x^{22} + (\alpha^{222} * \alpha^{251})x^{21} + (\alpha^{222} * \alpha^{67})x^{20} + (\alpha^{222} * \alpha^{46})x^{19} + (\alpha^{222} * \alpha^{61})x^{18} + (\alpha^{222} * \alpha^{118})x^{17} + (\alpha^{222} * \alpha^{70})x^{16} + (\alpha^{222} * \alpha^{64})x^{15} + (\alpha^{222} * \alpha^{94})x^{14} + (\alpha^{222} * \alpha^{32})x^{13} + (\alpha^{222} * \alpha^{45})x^{12}$

The exponents of the alphas are added together. In this case, at least one of the exponents is larger than 255, so perform modulo 255 as follows:

$\alpha^{222}x^{22} + \alpha^{(473 \% 255)}x^{21} + \alpha^{(289 \% 255)}x^{20} + \alpha^{(268 \% 255)}x^{19} + \alpha^{(283 \% 255)}x^{18} + \alpha^{(340 \% 255)}x^{17} + \alpha^{(292 \% 255)}x^{16} + \alpha^{(286 \% 255)}x^{15} + \alpha^{(316 \% 255)}x^{14} + \alpha^{254}x^{13} + \alpha^{(267 \% 255)}x^{12}$

The result is:

$\alpha^{222}x^{22} + \alpha^{218}x^{21} + \alpha^{34}x^{20} + \alpha^{13}x^{19} + \alpha^{28}x^{18} + \alpha^{85}x^{17} + \alpha^{37}x^{16} + \alpha^{31}x^{15} + \alpha^{61}x^{14} + \alpha^{254}x^{13} + \alpha^{12}x^{12}$

Now, convert this to integer notation:

$138x^{22} + 43x^{21} + 78x^{20} + 135x^{19} + 24x^{18} + 214x^{17} + 74x^{16} + 192x^{15} + 111x^{14} + 142x^{13} + 205x^{12}$

Step 4b: XOR the result with the result from step 3b

Use the result from step 3b to perform the next XOR.

$(138 \oplus 138)x^{22} + (216 \oplus 43)x^{21} + (161 \oplus 78)x^{20} + (87 \oplus 135)x^{19} + (194 \oplus 24)x^{18} + (232 \oplus 214)x^{17} + (204 \oplus 74)x^{16} + (155 \oplus 192)x^{15} + (48 \oplus 111)x^{14} + (193 \oplus 142)x^{13} + (17 \oplus 205)x^{12} + (236 \oplus 0)x^{11} + (17 \oplus 0)x^{10}$

The result is:

$0x^{22} + 243x^{21} + 239x^{20} + 208x^{19} + 218x^{18} + 62x^{17} + 134x^{16} + 91x^{15} + 95x^{14} + 79x^{13} + 220x^{12} + 236x^{11} + 17x^{10}$

Discard the lead 0 term to get:

$243x^{21} + 239x^{20} + 208x^{19} + 218x^{18} + 62x^{17} + 134x^{16} + 91x^{15} + 95x^{14} + 79x^{13} + 220x^{12} + 236x^{11} + 17x^{10}$

Step 5a: Multiply the Generator Polynomial by the Lead Term of the XOR result from the previous step

Next, multiply the generator polynomial by the lead term of the XOR result from the previous step. The lead term in this case is $243x^{21}$. Convert $243x^{21}$ to alpha notation. According to the log antilog table, for the integer value 243, the alpha exponent is 233. Therefore $243 = \alpha^{233}$. Multiply the generator polynomial by α^{233}:

$(\alpha^{233} * \alpha^{0})x^{21} + (\alpha^{233} * \alpha^{251})x^{20} + (\alpha^{233} * \alpha^{67})x^{19} + (\alpha^{233} * \alpha^{46})x^{18} + (\alpha^{233} * \alpha^{61})x^{17} + (\alpha^{233} * \alpha^{118})x^{16} + (\alpha^{233} * \alpha^{70})x^{15} + (\alpha^{233} * \alpha^{64})x^{14} + (\alpha^{233} * \alpha^{94})x^{13} + (\alpha^{233} * \alpha^{32})x^{12} + (\alpha^{233} * \alpha^{45})x^{11}$

The exponents of the alphas are added together. In this case, at least one of the exponents is larger than 255, so perform modulo 255 as follows:

$\alpha^{233}x^{21} + \alpha^{(484 \% 255)}x^{20} + \alpha^{(300 \% 255)}x^{19} + \alpha^{(279 \% 255)}x^{18} + \alpha^{(294 \% 255)}x^{17} + \alpha^{(351 \% 255)}x^{16} + \alpha^{(303 \% 255)}x^{15} + \alpha^{(297 \% 255)}x^{14} + \alpha^{(327 \% 255)}x^{13} + \alpha^{(265 \% 255)}x^{12} + \alpha^{(278 \% 255)}x^{11}$

The result is:

$\alpha^{233}x^{21} + \alpha^{229}x^{20} + \alpha^{45}x^{19} + \alpha^{24}x^{18} + \alpha^{39}x^{17} + \alpha^{96}x^{16} + \alpha^{48}x^{15} + \alpha^{42}x^{14} + \alpha^{72}x^{13} + \alpha^{10}x^{12} + \alpha^{23}x^{11}$

Now, convert this to integer notation:

$243x^{21} + 122x^{20} + 193x^{19} + 143x^{18} + 53x^{17} + 217x^{16} + 70x^{15} + 181x^{14} + 101x^{13} + 116x^{12} + 201x^{11}$

Step 5b: XOR the result with the result from step 4b

Use the result from step 4b to perform the next XOR.

$(243 \oplus 243)x^{21} + (239 \oplus 122)x^{20} + (208 \oplus 193)x^{19} + (218 \oplus 143)x^{18} + (62 \oplus 53)x^{17} + (134 \oplus 217)x^{16} + (91 \oplus 70)x^{15} + (95 \oplus 181)x^{14} + (79 \oplus 101)x^{13} + (220 \oplus 116)x^{12} + (236 \oplus 201)x^{11} + (17 \oplus 0)x^{10}$

The result is:

$0x^{21} + 149x^{20} + 17x^{19} + 85x^{18} + 11x^{17} + 95x^{16} + 29x^{15} + 234x^{14} + 42x^{13} + 168x^{12} + 37x^{11} + 17x^{10}$

Discard the lead 0 term to get:

$149x^{20} + 17x^{19} + 85x^{18} + 11x^{17} + 95x^{16} + 29x^{15} + 234x^{14} + 42x^{13} + 168x^{12} + 37x^{11} + 17x^{10}$

Step 6a: Multiply the Generator Polynomial by the Lead Term of the XOR result from the previous step

Next, multiply the generator polynomial by the lead term of the XOR result from the previous step. The lead term in this case is $149x^{20}$. Convert $149x^{20}$ to alpha notation. According to the log antilog table, for the integer value 149, the alpha exponent is 184. Therefore 149 = α^{184}. Multiply the generator polynomial by α^{184}:

$(\alpha^{184} * \alpha^{0})x^{20} + (\alpha^{184} * \alpha^{251})x^{19} + (\alpha^{184} * \alpha^{67})x^{18} + (\alpha^{184} * \alpha^{46})x^{17} + (\alpha^{184} * \alpha^{61})x^{16} + (\alpha^{184} * \alpha^{118})x^{15} + (\alpha^{184} * \alpha^{70})x^{14} + (\alpha^{184} * \alpha^{64})x^{13} + (\alpha^{184} * \alpha^{94})x^{12} + (\alpha^{184} * \alpha^{32})x^{11} + (\alpha^{184} * \alpha^{45})x^{10}$

The exponents of the alphas are added together. In this case, at least one of the exponents is larger than 255, so perform modulo 255 as follows:

$\alpha^{184}x^{20} + \alpha^{(435 \% 255)}x^{19} + \alpha^{251}x^{18} + \alpha^{230}x^{17} + \alpha^{245}x^{16} + \alpha^{(302 \% 255)}x^{15} + \alpha^{254}x^{14} + \alpha^{248}x^{13} + \alpha^{(278 \% 255)}x^{12} + \alpha^{216}x^{11} + \alpha^{229}x^{10}$

The result is:

$\alpha^{184}x^{20} + \alpha^{180}x^{19} + \alpha^{251}x^{18} + \alpha^{230}x^{17} + \alpha^{245}x^{16} + \alpha^{47}x^{15} + \alpha^{254}x^{14} + \alpha^{248}x^{13} + \alpha^{23}x^{12} + \alpha^{216}x^{11} + \alpha^{229}x^{10}$

Now, convert this to integer notation:

$149x^{20} + 150x^{19} + 216x^{18} + 244x^{17} + 233x^{16} + 35x^{15} + 142x^{14} + 27x^{13} + 201x^{12} + 195x^{11} + 122x^{10}$

Step 6b: XOR the result with the result from step 5b

Use the result from step 5b to perform the next XOR.

$(149 \oplus 149)x^{20} + (17 \oplus 150)x^{19} + (85 \oplus 216)x^{18} + (11 \oplus 244)x^{17} + (95 \oplus 233)x^{16} + (29 \oplus 35)x^{15} + (234 \oplus 142)x^{14} + (42 \oplus 27)x^{13} + (168 \oplus 201)x^{12} + (37 \oplus 195)x^{11} + (17 \oplus 122)x^{10}$

The result is:

$0x^{20} + 135x^{19} + 141x^{18} + 255x^{17} + 182x^{16} + 62x^{15} + 100x^{14} + 49x^{13} + 97x^{12} + 230x^{11} + 107x^{10}$

Discard the lead 0 term to get:

$135x^{19} + 141x^{18} + 255x^{17} + 182x^{16} + 62x^{15} + 100x^{14} + 49x^{13} + 97x^{12} + 230x^{11} + 107x^{10}$

Step 7a: Multiply the Generator Polynomial by the Lead Term of the XOR result from the previous step

Next, multiply the generator polynomial by the lead term of the XOR result from the previous step. The lead term in this case is $135x^{19}$. Convert $135x^{19}$ to alpha notation. According to the log antilog table, for the integer value 135, the alpha exponent is 13. Therefore 135 = α^{13}. Multiply the generator polynomial by α^{13}:

$(\alpha^{13} * \alpha^{0})x^{19} + (\alpha^{13} * \alpha^{251})x^{18} + (\alpha^{13} * \alpha^{67})x^{17} + (\alpha^{13} * \alpha^{46})x^{16} + (\alpha^{13} * \alpha^{61})x^{15} + (\alpha^{13} * \alpha^{118})x^{14} + (\alpha^{13} * \alpha^{70})x^{13} + (\alpha^{13} * \alpha^{64})x^{12} + (\alpha^{13} * \alpha^{94})x^{11} + (\alpha^{13} * \alpha^{32})x^{10} + (\alpha^{13} * \alpha^{45})x^{9}$

The exponents of the alphas are added together. In this case, at least one of the exponents is larger than 255, so perform modulo 255 as follows:

$\alpha^{13}x^{19} + \alpha^{(264 \% 255)}x^{18} + \alpha^{80}x^{17} + \alpha^{59}x^{16} + \alpha^{74}x^{15} + \alpha^{131}x^{14} + \alpha^{83}x^{13} + \alpha^{77}x^{12} + \alpha^{107}x^{11} + \alpha^{45}x^{10} + \alpha^{58}x^{9}$

The result is:

$\alpha^{13}x^{19} + \alpha^{9}x^{18} + \alpha^{80}x^{17} + \alpha^{59}x^{16} + \alpha^{74}x^{15} + \alpha^{131}x^{14} + \alpha^{83}x^{13} + \alpha^{77}x^{12} + \alpha^{107}x^{11} + \alpha^{45}x^{10} + \alpha^{58}x^{9}$

Now, convert this to integer notation:

$135x^{19} + 58x^{18} + 253x^{17} + 210x^{16} + 137x^{15} + 92x^{14} + 187x^{13} + 60x^{12} + 104x^{11} + 193x^{10} + 105x^{9}$

Step 7b: XOR the result with the result from step 6b

Use the result from step 6b to perform the next XOR.

$(135 \oplus 135)x^{19} + (141 \oplus 58)x^{18} + (255 \oplus 253)x^{17} + (182 \oplus 210)x^{16} + (62 \oplus 137)x^{15} + (100 \oplus 92)x^{14} + (49 \oplus 187)x^{13} + (97 \oplus 60)x^{12} + (230 \oplus 104)x^{11} + (107 \oplus 193)x^{10} + (0 \oplus 105)x^{9}$

The result is:

$0x^{19} + 183x^{18} + 2x^{17} + 100x^{16} + 183x^{15} + 56x^{14} + 138x^{13} + 93x^{12} + 142x^{11} + 170x^{10} + 105x^{9}$

Discard the lead 0 term to get:

$183x^{18} + 2x^{17} + 100x^{16} + 183x^{15} + 56x^{14} + 138x^{13} + 93x^{12} + 142x^{11} + 170x^{10} + 105x^{9}$

Step 8a: Multiply the Generator Polynomial by the Lead Term of the XOR result from the previous step

Next, multiply the generator polynomial by the lead term of the XOR result from the previous step. The lead term in this case is $183x^{18}$. Convert $183x^{18}$ to alpha notation. According to the log antilog table, for the integer value 183, the alpha exponent is 158. Therefore 183 = α^{158}. Multiply the generator polynomial by α^{158}:

$(\alpha^{158} * \alpha^{0})x^{18} + (\alpha^{158} * \alpha^{251})x^{17} + (\alpha^{158} * \alpha^{67})x^{16} + (\alpha^{158} * \alpha^{46})x^{15} + (\alpha^{158} * \alpha^{61})x^{14} + (\alpha^{158} * \alpha^{118})x^{13} + (\alpha^{158} * \alpha^{70})x^{12} + (\alpha^{158} * \alpha^{64})x^{11} + (\alpha^{158} * \alpha^{94})x^{10} + (\alpha^{158} * \alpha^{32})x^{9} + (\alpha^{158} * \alpha^{45})x^{8}$

The exponents of the alphas are added together. In this case, at least one of the exponents is larger than 255, so perform modulo 255 as follows:

$\alpha^{158}x^{18} + \alpha^{(409 \% 255)}x^{17} + \alpha^{225}x^{16} + \alpha^{204}x^{15} + \alpha^{219}x^{14} + \alpha^{(276 \% 255)}x^{13} + \alpha^{228}x^{12} + \alpha^{222}x^{11} + \alpha^{252}x^{10} + \alpha^{190}x^{9} + \alpha^{203}x^{8}$

The result is:

$\alpha^{158}x^{18} + \alpha^{154}x^{17} + \alpha^{225}x^{16} + \alpha^{204}x^{15} + \alpha^{219}x^{14} + \alpha^{21}x^{13} + \alpha^{228}x^{12} + \alpha^{222}x^{11} + \alpha^{252}x^{10} + \alpha^{190}x^{9} + \alpha^{203}x^{8}$

Now, convert this to integer notation:

$183x^{18} + 57x^{17} + 36x^{16} + 221x^{15} + 86x^{14} + 117x^{13} + 61x^{12} + 138x^{11} + 173x^{10} + 174x^{9} + 224x^{8}$

Step 8b: XOR the result with the result from step 7b

Use the result from step 7b to perform the next XOR.

$(183 \oplus 183)x^{18} + (2 \oplus 57)x^{17} + (100 \oplus 36)x^{16} + (183 \oplus 221)x^{15} + (56 \oplus 86)x^{14} + (138 \oplus 117)x^{13} + (93 \oplus 61)x^{12} + (142 \oplus 138)x^{11} + (170 \oplus 173)x^{10} + (105 \oplus 174)x^{9} + (0 \oplus 224)x^{8}$

The result is:

$0x^{18} + 59x^{17} + 64x^{16} + 106x^{15} + 110x^{14} + 255x^{13} + 96x^{12} + 4x^{11} + 7x^{10} + 199x^{9} + 224x^{8}$

Discard the lead 0 term to get:

$59x^{17} + 64x^{16} + 106x^{15} + 110x^{14} + 255x^{13} + 96x^{12} + 4x^{11} + 7x^{10} + 199x^{9} + 224x^{8}$

Step 9a: Multiply the Generator Polynomial by the Lead Term of the XOR result from the previous step

Next, multiply the generator polynomial by the lead term of the XOR result from the previous step. The lead term in this case is $59x^{17}$. Convert $59x^{17}$ to alpha notation. According to the log antilog table, for the integer value 59, the alpha exponent is 120. Therefore $59 = \alpha^{120}$. Multiply the generator polynomial by α^{120}:

$(\alpha^{120} * \alpha^{0})x^{17} + (\alpha^{120} * \alpha^{251})x^{16} + (\alpha^{120} * \alpha^{67})x^{15} + (\alpha^{120} * \alpha^{46})x^{14} + (\alpha^{120} * \alpha^{61})x^{13} + (\alpha^{120} * \alpha^{118})x^{12} + (\alpha^{120} * \alpha^{70})x^{11} + (\alpha^{120} * \alpha^{64})x^{10} + (\alpha^{120} * \alpha^{94})x^{9} + (\alpha^{120} * \alpha^{32})x^{8} + (\alpha^{120} * \alpha^{45})x^{7}$

The exponents of the alphas are added together. In this case, at least one of the exponents is larger than 255, so perform modulo 255 as follows:

$\alpha^{120}x^{17} + \alpha^{(371 \% 255)}x^{16} + \alpha^{187}x^{15} + \alpha^{166}x^{14} + \alpha^{181}x^{13} + \alpha^{238}x^{12} + \alpha^{190}x^{11} + \alpha^{184}x^{10} + \alpha^{214}x^{9} + \alpha^{152}x^{8} + \alpha^{165}x^{7}$

The result is:

$\alpha^{120}x^{17} + \alpha^{116}x^{16} + \alpha^{187}x^{15} + \alpha^{166}x^{14} + \alpha^{181}x^{13} + \alpha^{238}x^{12} + \alpha^{190}x^{11} + \alpha^{184}x^{10} + \alpha^{214}x^{9} + \alpha^{152}x^{8} + \alpha^{165}x^{7}$

Now, convert this to integer notation:

$59x^{17} + 248x^{16} + 220x^{15} + 63x^{14} + 49x^{13} + 11x^{12} + 174x^{11} + 149x^{10} + 249x^{9} + 73x^{8} + 145x^{7}$

Step 9b: XOR the result with the result from step 8b

Use the result from step 8b to perform the next XOR.

$(59 \oplus 59)x^{17} + (64 \oplus 248)x^{16} + (106 \oplus 220)x^{15} + (110 \oplus 63)x^{14} + (255 \oplus 49)x^{13} + (96 \oplus 11)x^{12} + (4 \oplus 174)x^{11} + (7 \oplus 149)x^{10} + (199 \oplus 249)x^{9} + (224 \oplus 73)x^{8} + (0 \oplus 145)x^{7}$

The result is:

$0x^{17} + 184x^{16} + 182x^{15} + 81x^{14} + 206x^{13} + 107x^{12} + 170x^{11} + 146x^{10} + 62x^{9} + 169x^{8} + 145x^{7}$

Discard the lead 0 term to get:

$184x^{16} + 182x^{15} + 81x^{14} + 206x^{13} + 107x^{12} + 170x^{11} + 146x^{10} + 62x^{9} + 169x^{8} + 145x^{7}$

Step 10a: Multiply the Generator Polynomial by the Lead Term of the XOR result from the previous step

Next, multiply the generator polynomial by the lead term of the XOR result from the previous step. The lead term in this case is $184x^{16}$. Convert $184x^{16}$ to alpha notation. According to the log antilog table, for the integer value 184, the alpha exponent is 132. Therefore $184 = \alpha^{132}$. Multiply the generator polynomial by α^{132}:

$(\alpha^{132} * \alpha^{0})x^{16} + (\alpha^{132} * \alpha^{251})x^{15} + (\alpha^{132} * \alpha^{67})x^{14} + (\alpha^{132} * \alpha^{46})x^{13} + (\alpha^{132} * \alpha^{61})x^{12} + (\alpha^{132} * \alpha^{118})x^{11} + (\alpha^{132} * \alpha^{70})x^{10} + (\alpha^{132} * \alpha^{64})x^{9} + (\alpha^{132} * \alpha^{94})x^{8} + (\alpha^{132} * \alpha^{32})x^{7} + (\alpha^{132} * \alpha^{45})x^{6}$

The exponents of the alphas are added together. In this case, at least one of the exponents is larger than 255, so perform modulo 255 as follows:

$\alpha^{132}x^{16} + \alpha^{(383 \% 255)}x^{15} + \alpha^{199}x^{14} + \alpha^{178}x^{13} + \alpha^{193}x^{12} + \alpha^{250}x^{11} + \alpha^{202}x^{10} + \alpha^{196}x^{9} + \alpha^{226}x^{8} + \alpha^{164}x^{7} + \alpha^{177}x^{6}$

The result is:

$\alpha^{132}x^{16} + \alpha^{128}x^{15} + \alpha^{199}x^{14} + \alpha^{178}x^{13} + \alpha^{193}x^{12} + \alpha^{250}x^{11} + \alpha^{202}x^{10} + \alpha^{196}x^{9} + \alpha^{226}x^{8} + \alpha^{164}x^{7} + \alpha^{177}x^{6}$

Now, convert this to integer notation:

$184x^{16} + 133x^{15} + 14x^{14} + 171x^{13} + 25x^{12} + 108x^{11} + 112x^{10} + 200x^{9} + 72x^{8} + 198x^{7} + 219x^{6}$

Step 10b: XOR the result with the result from step 9b

Use the result from step 9b to perform the next XOR.

$(184 \oplus 184)x^{16} + (182 \oplus 133)x^{15} + (81 \oplus 14)x^{14} + (206 \oplus 171)x^{13} + (107 \oplus 25)x^{12} + (170 \oplus 108)x^{11} + (146 \oplus 112)x^{10} + (62 \oplus 200)x^{9} + (169 \oplus 72)x^{8} + (145 \oplus 198)x^{7} + (0 \oplus 219)x^{6}$

The result is:

$0x^{16} + 51x^{15} + 95x^{14} + 101x^{13} + 114x^{12} + 198x^{11} + 226x^{10} + 246x^{9} + 225x^{8} + 87x^{7} + 219x^{6}$

Discard the lead 0 term to get:

$51x^{15} + 95x^{14} + 101x^{13} + 114x^{12} + 198x^{11} + 226x^{10} + 246x^{9} + 225x^{8} + 87x^{7} + 219x^{6}$

Step 11a: Multiply the Generator Polynomial by the Lead Term of the XOR result from the previous step

Next, multiply the generator polynomial by the lead term of the XOR result from the previous step. The lead term in this case is $51x^{15}$. Convert $51x^{15}$ to alpha notation. According to the log antilog table, for the integer value 51, the alpha exponent is 125. Therefore $51 = \alpha^{125}$. Multiply the generator polynomial by α^{125}:

$(\alpha^{125} * \alpha^{0})x^{15} + (\alpha^{125} * \alpha^{251})x^{14} + (\alpha^{125} * \alpha^{67})x^{13} + (\alpha^{125} * \alpha^{46})x^{12} + (\alpha^{125} * \alpha^{61})x^{11} + (\alpha^{125} * \alpha^{118})x^{10} + (\alpha^{125} * \alpha^{70})x^{9} + (\alpha^{125} * \alpha^{64})x^{8} + (\alpha^{125} * \alpha^{94})x^{7} + (\alpha^{125} * \alpha^{32})x^{6} + (\alpha^{125} * \alpha^{45})x^{5}$

The exponents of the alphas are added together. In this case, at least one of the exponents is larger than 255, so perform modulo 255 as follows:

$\alpha^{125}x^{15} + \alpha^{(376 \% 255)}x^{14} + \alpha^{192}x^{13} + \alpha^{171}x^{12} + \alpha^{186}x^{11} + \alpha^{243}x^{10} + \alpha^{195}x^{9} + \alpha^{189}x^{8} + \alpha^{219}x^{7} + \alpha^{157}x^{6} + \alpha^{170}x^{5}$

The result is:

$\alpha^{125}x^{15} + \alpha^{121}x^{14} + \alpha^{192}x^{13} + \alpha^{171}x^{12} + \alpha^{186}x^{11} + \alpha^{243}x^{10} + \alpha^{195}x^{9} + \alpha^{189}x^{8} + \alpha^{219}x^{7} + \alpha^{157}x^{6} + \alpha^{170}x^{5}$

Now, convert this to integer notation:

$51x^{15} + 118x^{14} + 130x^{13} + 179x^{12} + 110x^{11} + 125x^{10} + 100x^{9} + 87x^{8} + 86x^{7} + 213x^{6} + 215x^{5}$

Step 11b: XOR the result with the result from step 10b

Use the result from step 10b to perform the next XOR.

$(51 \oplus 51)x^{15} + (95 \oplus 118)x^{14} + (101 \oplus 130)x^{13} + (114 \oplus 179)x^{12} + (198 \oplus 110)x^{11} + (226 \oplus 125)x^{10} + (246 \oplus 100)x^{9} + (225 \oplus 87)x^{8} + (87 \oplus 86)x^{7} + (219 \oplus 213)x^{6} + (0 \oplus 215)x^{5}$

The result is:

$0x^{15} + 41x^{14} + 231x^{13} + 193x^{12} + 168x^{11} + 159x^{10} + 146x^{9} + 182x^{8} + 1x^{7} + 14x^{6} + 215x^{5}$

Discard the lead 0 term to get:

$41x^{14} + 231x^{13} + 193x^{12} + 168x^{11} + 159x^{10} + 146x^{9} + 182x^{8} + 1x^{7} + 14x^{6} + 215x^{5}$

Step 12a: Multiply the Generator Polynomial by the Lead Term of the XOR result from the previous step

Next, multiply the generator polynomial by the lead term of the XOR result from the previous step. The lead term in this case is $41x^{14}$. Convert $41x^{14}$ to alpha notation. According to the log antilog table, for the integer value 41, the alpha exponent is 147. Therefore $41 = \alpha^{147}$. Multiply the generator polynomial by α^{147}:

$(\alpha^{147} * \alpha^{0})x^{14} + (\alpha^{147} * \alpha^{251})x^{13} + (\alpha^{147} * \alpha^{67})x^{12} + (\alpha^{147} * \alpha^{46})x^{11} + (\alpha^{147} * \alpha^{61})x^{10} + (\alpha^{147} * \alpha^{118})x^{9} + (\alpha^{147} * \alpha^{70})x^{8} + (\alpha^{147} * \alpha^{64})x^{7} + (\alpha^{147} * \alpha^{94})x^{6} + (\alpha^{147} * \alpha^{32})x^{5} + (\alpha^{147} * \alpha^{45})x^{4}$

The exponents of the alphas are added together. In this case, at least one of the exponents is larger than 255, so perform modulo 255 as follows:

$\alpha^{147}x^{14} + \alpha^{(398\ \%\ 255)}x^{13} + \alpha^{214}x^{12} + \alpha^{193}x^{11} + \alpha^{208}x^{10} + \alpha^{(265\ \%\ 255)}x^{9} + \alpha^{217}x^{8} + \alpha^{211}x^{7} + \alpha^{241}x^{6} + \alpha^{179}x^{5} + \alpha^{192}x^{4}$

The result is:

$\alpha^{147}x^{14} + \alpha^{143}x^{13} + \alpha^{214}x^{12} + \alpha^{193}x^{11} + \alpha^{208}x^{10} + \alpha^{10}x^{9} + \alpha^{217}x^{8} + \alpha^{211}x^{7} + \alpha^{241}x^{6} + \alpha^{179}x^{5} + \alpha^{192}x^{4}$

Now, convert this to integer notation:

$41x^{14} + 84x^{13} + 249x^{12} + 25x^{11} + 81x^{10} + 116x^{9} + 155x^{8} + 178x^{7} + 88x^{6} + 75x^{5} + 130x^{4}$

Step 12b: XOR the result with the result from step 11b

Use the result from step 11b to perform the next XOR.

$(41 \oplus 41)x^{14} + (231 \oplus 84)x^{13} + (193 \oplus 249)x^{12} + (168 \oplus 25)x^{11} + (159 \oplus 81)x^{10} + (146 \oplus 116)x^{9} + (182 \oplus 155)x^{8} + (1 \oplus 178)x^{7} + (14 \oplus 88)x^{6} + (215 \oplus 75)x^{5} + (0 \oplus 130)x^{4}$

The result is:

$0x^{14} + 179x^{13} + 56x^{12} + 177x^{11} + 206x^{10} + 230x^{9} + 45x^{8} + 179x^{7} + 86x^{6} + 156x^{5} + 130x^{4}$

Discard the lead 0 term to get:

$179x^{13} + 56x^{12} + 177x^{11} + 206x^{10} + 230x^{9} + 45x^{8} + 179x^{7} + 86x^{6} + 156x^{5} + 130x^{4}$

Step 13a: Multiply the Generator Polynomial by the Lead Term of the XOR

result from the previous step

Next, multiply the generator polynomial by the lead term of the XOR result from the previous step. The lead term in this case is $179x^{13}$. Convert $179x^{13}$ to alpha notation. According to the log antilog table, for the integer value 179, the alpha exponent is 171. Therefore $179 = \alpha^{171}$. Multiply the generator polynomial by α^{171}:

$(\alpha^{171} * \alpha^{0})x^{13} + (\alpha^{171} * \alpha^{251})x^{12} + (\alpha^{171} * \alpha^{67})x^{11} + (\alpha^{171} * \alpha^{46})x^{10} + (\alpha^{171} * \alpha^{61})x^{9} + (\alpha^{171} * \alpha^{118})x^{8} +$
$(\alpha^{171} * \alpha^{70})x^{7} + (\alpha^{171} * \alpha^{64})x^{6} + (\alpha^{171} * \alpha^{94})x^{5} + (\alpha^{171} * \alpha^{32})x^{4} + (\alpha^{171} * \alpha^{45})x^{3}$

The exponents of the alphas are added together. In this case, at least one of the exponents is larger than 255, so perform modulo 255 as follows:

$\alpha^{171}x^{13} + \alpha^{(422 \% 255)}x^{12} + \alpha^{238}x^{11} + \alpha^{217}x^{10} + \alpha^{232}x^{9} + \alpha^{(289 \% 255)}x^{8} + \alpha^{241}x^{7} + \alpha^{235}x^{6} + \alpha^{(265 \% 255)}x^{5}$
$+ \alpha^{203}x^{4} + \alpha^{216}x^{3}$

The result is:

$\alpha^{171}x^{13} + \alpha^{167}x^{12} + \alpha^{238}x^{11} + \alpha^{217}x^{10} + \alpha^{232}x^{9} + \alpha^{34}x^{8} + \alpha^{241}x^{7} + \alpha^{235}x^{6} + \alpha^{10}x^{5} + \alpha^{203}x^{4} + \alpha^{216}x^{3}$

Now, convert this to integer notation:

$179x^{13} + 126x^{12} + 11x^{11} + 155x^{10} + 247x^{9} + 78x^{8} + 88x^{7} + 235x^{6} + 116x^{5} + 224x^{4} + 195x^{3}$

Step 13b: XOR the result with the result from step 12b

Use the result from step 12b to perform the next XOR.

$(179 \oplus 179)x^{13} + (56 \oplus 126)x^{12} + (177 \oplus 11)x^{11} + (206 \oplus 155)x^{10} + (230 \oplus 247)x^{9} + (45 \oplus 78)x^{8} + (179$
$\oplus 88)x^{7} + (86 \oplus 235)x^{6} + (156 \oplus 116)x^{5} + (130 \oplus 224)x^{4} + (0 \oplus 195)x^{3}$

The result is:

$0x^{13} + 70x^{12} + 186x^{11} + 85x^{10} + 17x^{9} + 99x^{8} + 235x^{7} + 189x^{6} + 232x^{5} + 98x^{4} + 195x^{3}$

Discard the lead 0 term to get:

$70x^{12} + 186x^{11} + 85x^{10} + 17x^{9} + 99x^{8} + 235x^{7} + 189x^{6} + 232x^{5} + 98x^{4} + 195x^{3}$

Step 14a: Multiply the Generator Polynomial by the Lead Term of the XOR result from the previous step

Next, multiply the generator polynomial by the lead term of the XOR result from the previous step. The lead term in this case is $70x^{12}$. Convert $70x^{12}$ to alpha notation. According to the log antilog table, for the integer value 70, the alpha exponent is 48. Therefore $70 = \alpha^{48}$. Multiply the generator polynomial by α^{48}:

$(\alpha^{48} * \alpha^{0})x^{12} + (\alpha^{48} * \alpha^{251})x^{11} + (\alpha^{48} * \alpha^{67})x^{10} + (\alpha^{48} * \alpha^{46})x^{9} + (\alpha^{48} * \alpha^{61})x^{8} + (\alpha^{48} * \alpha^{118})x^{7} + (\alpha^{48} *$
$\alpha^{70})x^{6} + (\alpha^{48} * \alpha^{64})x^{5} + (\alpha^{48} * \alpha^{94})x^{4} + (\alpha^{48} * \alpha^{32})x^{3} + (\alpha^{48} * \alpha^{45})x^{2}$

The exponents of the alphas are added together. In this case, at least one of the exponents is larger than 255, so perform modulo 255 as follows:

$\alpha^{48}x^{12} + \alpha^{(299 \% 255)}x^{11} + \alpha^{115}x^{10} + \alpha^{94}x^{9} + \alpha^{109}x^{8} + \alpha^{166}x^{7} + \alpha^{118}x^{6} + \alpha^{112}x^{5} + \alpha^{142}x^{4} + \alpha^{80}x^{3} + \alpha^{93}x^{2}$

The result is:

$\alpha^{48}x^{12} + \alpha^{44}x^{11} + \alpha^{115}x^{10} + \alpha^{94}x^{9} + \alpha^{109}x^{8} + \alpha^{166}x^{7} + \alpha^{118}x^{6} + \alpha^{112}x^{5} + \alpha^{142}x^{4} + \alpha^{80}x^{3} + \alpha^{93}x^{2}$

Now, convert this to integer notation:

$70x^{12} + 238x^{11} + 124x^{10} + 113x^{9} + 189x^{8} + 63x^{7} + 199x^{6} + 129x^{5} + 42x^{4} + 253x^{3} + 182x^{2}$

Step 14b: XOR the result with the result from step 13b

Use the result from step 13b to perform the next XOR.

$(70 \oplus 70)x^{12} + (186 \oplus 238)x^{11} + (85 \oplus 124)x^{10} + (17 \oplus 113)x^{9} + (99 \oplus 189)x^{8} + (235 \oplus 63)x^{7} + (189 \oplus 199)x^{6} + (232 \oplus 129)x^{5} + (98 \oplus 42)x^{4} + (195 \oplus 253)x^{3} + (0 \oplus 182)x^{2}$

The result is:

$0x^{12} + 84x^{11} + 41x^{10} + 96x^{9} + 222x^{8} + 212x^{7} + 122x^{6} + 105x^{5} + 72x^{4} + 62x^{3} + 182x^{2}$

Discard the lead 0 term to get:

$84x^{11} + 41x^{10} + 96x^{9} + 222x^{8} + 212x^{7} + 122x^{6} + 105x^{5} + 72x^{4} + 62x^{3} + 182x^{2}$

Step 15a: Multiply the Generator Polynomial by the Lead Term of the XOR result from the previous step

Next, multiply the generator polynomial by the lead term of the XOR result from the previous step. The lead term in this case is $84x^{11}$. Convert $84x^{11}$ to alpha notation. According to the log antilog table, for the integer value 84, the alpha exponent is 143. Therefore $84 = \alpha^{143}$. Multiply the generator polynomial by α^{143}:

$(\alpha^{143} * \alpha^{0})x^{11} + (\alpha^{143} * \alpha^{251})x^{10} + (\alpha^{143} * \alpha^{67})x^{9} + (\alpha^{143} * \alpha^{46})x^{8} + (\alpha^{143} * \alpha^{61})x^{7} + (\alpha^{143} * \alpha^{118})x^{6} + (\alpha^{143} * \alpha^{70})x^{5} + (\alpha^{143} * \alpha^{64})x^{4} + (\alpha^{143} * \alpha^{94})x^{3} + (\alpha^{143} * \alpha^{32})x^{2} + (\alpha^{143} * \alpha^{45})x^{1}$

The exponents of the alphas are added together. In this case, at least one of the exponents is larger than 255, so perform modulo 255 as follows:

$\alpha^{143}x^{11} + \alpha^{(394 \% 255)}x^{10} + \alpha^{210}x^{9} + \alpha^{189}x^{8} + \alpha^{204}x^{7} + \alpha^{(261 \% 255)}x^{6} + \alpha^{213}x^{5} + \alpha^{207}x^{4} + \alpha^{237}x^{3} + \alpha^{175}x^{2} + \alpha^{188}x^{1}$

The result is:

$\alpha^{143}x^{11} + \alpha^{139}x^{10} + \alpha^{210}x^{9} + \alpha^{189}x^{8} + \alpha^{204}x^{7} + \alpha^{6}x^{6} + \alpha^{213}x^{5} + \alpha^{207}x^{4} + \alpha^{237}x^{3} + \alpha^{175}x^{2} + \alpha^{188}x^{1}$

Now, convert this to integer notation:

$84x^{11} + 66x^{10} + 89x^{9} + 87x^{8} + 221x^{7} + 64x^{6} + 242x^{5} + 166x^{4} + 139x^{3} + 255x^{2} + 165x^{1}$

Step 15b: XOR the result with the result from step 14b

Use the result from step 14b to perform the next XOR.

$(84 \oplus 84)x^{11} + (41 \oplus 66)x^{10} + (96 \oplus 89)x^{9} + (222 \oplus 87)x^{8} + (212 \oplus 221)x^{7} + (122 \oplus 64)x^{6} + (105 \oplus 242)x^{5} + (72 \oplus 166)x^{4} + (62 \oplus 139)x^{3} + (182 \oplus 255)x^{2} + (0 \oplus 165)x^{1}$

The result is:

$0x^{11} + 107x^{10} + 57x^{9} + 137x^{8} + 9x^{7} + 58x^{6} + 155x^{5} + 238x^{4} + 181x^{3} + 73x^{2} + 165x^{1}$

Discard the lead 0 term to get:

$107x^{10} + 57x^{9} + 137x^{8} + 9x^{7} + 58x^{6} + 155x^{5} + 238x^{4} + 181x^{3} + 73x^{2} + 165x^{1}$

Step 16a: Multiply the Generator Polynomial by the Lead Term of the XOR

result from the previous step

Next, multiply the generator polynomial by the lead term of the XOR result from the previous step. The lead term in this case is $107x^{10}$. Convert $107x^{10}$ to alpha notation. According to the log antilog table, for the integer value 107, the alpha exponent is 84. Therefore $107 = \alpha^{84}$. Multiply the generator polynomial by α^{84}:

$(\alpha^{84} * \alpha^{0})x^{10} + (\alpha^{84} * \alpha^{251})x^{9} + (\alpha^{84} * \alpha^{67})x^{8} + (\alpha^{84} * \alpha^{46})x^{7} + (\alpha^{84} * \alpha^{61})x^{6} + (\alpha^{84} * \alpha^{118})x^{5} + (\alpha^{84} * \alpha^{70})x^{4} + (\alpha^{84} * \alpha^{64})x^{3} + (\alpha^{84} * \alpha^{94})x^{2} + (\alpha^{84} * \alpha^{32})x^{1} + (\alpha^{84} * \alpha^{45})x^{0}$

The exponents of the alphas are added together. In this case, at least one of the exponents is larger than 255, so perform modulo 255 as follows:

$\alpha^{84}x^{10} + \alpha^{(335 \% 255)}x^{9} + \alpha^{151}x^{8} + \alpha^{130}x^{7} + \alpha^{145}x^{6} + \alpha^{202}x^{5} + \alpha^{154}x^{4} + \alpha^{148}x^{3} + \alpha^{178}x^{2} + \alpha^{116}x^{1} + \alpha^{129}x^{0}$

The result is:

$\alpha^{84}x^{10} + \alpha^{80}x^{9} + \alpha^{151}x^{8} + \alpha^{130}x^{7} + \alpha^{145}x^{6} + \alpha^{202}x^{5} + \alpha^{154}x^{4} + \alpha^{148}x^{3} + \alpha^{178}x^{2} + \alpha^{116}x^{1} + \alpha^{129}x^{0}$

Now, convert this to integer notation:

$107x^{10} + 253x^{9} + 170x^{8} + 46x^{7} + 77x^{6} + 112x^{5} + 57x^{4} + 82x^{3} + 171x^{2} + 248x^{1} + 23$

Step 16b: XOR the result with the result from step 15b

Use the result from step 15b to perform the next XOR.

$(107 \oplus 107)x^{10} + (57 \oplus 253)x^{9} + (137 \oplus 170)x^{8} + (9 \oplus 46)x^{7} + (58 \oplus 77)x^{6} + (155 \oplus 112)x^{5} + (238 \oplus 57)x^{4} + (181 \oplus 82)x^{3} + (73 \oplus 171)x^{2} + (165 \oplus 248)x^{1} + (0 \oplus 23)x^{0}$

The result is:

$0x^{10} + 196x^{9} + 35x^{8} + 39x^{7} + 119x^{6} + 235x^{5} + 215x^{4} + 231x^{3} + 226x^{2} + 93x^{1} + 23$

Discard the lead 0 term to get:

$196x^{9} + 35x^{8} + 39x^{7} + 119x^{6} + 235x^{5} + 215x^{4} + 231x^{3} + 226x^{2} + 93x^{1} + 23$

Use the terms of the remainder as the error correction codewords

The division has been performed 16 times, which is the number of terms in the message polynomial. This means that the division is complete and the terms of the above polynomial are the error correction codewords to use for the original message polynomial:

196 35 39 119 235 215 231 226 93 23

Next: Structure Final Message

The data codewords and error correction codewords have now been generated. For larger QR codes, the QR code specification requires that the codewords be interleaved according to a particular pattern. The next section, structure final message (structure-final-message) explains the interleaving process.

(http://creativecommons.org/licenses/by-nc/4.0/)

Thonky.com's QR Code Tutorial by Thonky (http://www.thonky.com/) is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (http://creativecommons.org/licenses/by-nc/4.0/).

QR Code is registered trademark of DENSO WAVE (http://www.denso-wave.com/en/adcd/) INCORPORATED.

Page last updated 2017-04-14T15:02:18-05:00

© 2018 Thonky.com (/) | Privacy Policy (/globalpriv.php) | About Thonky.com (/about/) | Support Thonky.com (/support-thonky/)