Error detection and correction:
Standard arrays and table look up for decoding:

A standard array for an [*n*,*k*]-code is a q^{n-k} by q^k array where:

1. The first row lists all codewords (with the 0 codeword on the extreme left)
2. Each row is a coset with the coset leader in the first column
3. The entry in the i-th row and j-th column is the sum of the i-th coset leader and the j-th codeword.

For example, the [5,2]-code C_3 = {0, 01101, 10110, 11011} has a standard array as follows:

0	01101	10110	11011
10000	11101	00110	01011
01000	00101	11110	10011
00100	01001	10010	11111
00010	01111	10100	11001
00001	01100	10111	11010
11000	10101	01110	00011
10001	11100	00111	01010

Note that the above is only one possibility for the standard array; had 00011 been chosen as the first coset leader of weight two, another standard array representing the code would have been constructed.

Note that the first row contains the 0 vector and the codewords of C_3 (0 itself being a codeword). Also, the leftmost column contains the vectors of minimum weight enumerating vectors of weight 1 first and then using vectors of weight 2. Note also that each possible vector in the vector space appears exactly once.

Steps of Constructing a standard array

Because each possible vector can appear only once in a standard array some care must be taken during construction. A standard array can be created as follows:

1. List the codewords of C, starting with 0, as the first row
2. Choose any vector of minimum weight not already in the array. Write this as the first entry of the next row. This vector is denoted the '**coset leader**'.
3. Fill out the row by adding the coset leader to the codeword at the top of each column. The sum of the i-th coset leader and the j-th codeword becomes the entry in row i, column j.
4. Repeat steps 2 and 3 until all rows/cosets are listed and each vector appears exactly once.

Note that adding vectors is done mod q. For example, binary codes are added mod 2 (which equivalent to bit-wise XOR addition). For example, in Z_2, 11000 + 11011 = 00011.

Note also that selecting different coset leaders will create a slightly different but equivalent standard array, and will not affect results when decoding.

Definitions:

- **Given X ∈ {0,1}ⁿ, the Hamming Weight of X is the number of 1's in X**

$$\begin{array}{cccc} C_1 & C_2 & .... & C_M \\ e_1 & e_1 + C_2 & & e_1 + C_M \\ : & e_2 + C_2 & & e_2 + C_M \end{array}$$

Each vector having minimum weight in a coset is called a coset leader.

Example:

Let C be the binary [4,2]-code. i.e. C = {0000, 1011, 0101, 1110}. To construct the standard array, we first list the codewords in a row.

0000	1011	0101	1110

We then select a vector of minimum weight (in this case, weight 1) that has not been used. This vector becomes the coset leader for the second row.

0000	1011	0101	1110
1000			

Following step 3, we complete the row by adding the coset leader to each codeword.

0000	1011	0101	1110
1000	0011	1101	0110

We then repeat steps 2 and 3 until we have completed all rows. We stop when we have reached $q^{n-k} = 2^{4-2} = 2^2 = 4$ rows.

0000	1011	0101	1110
1000	0011	1101	0110
0100	1111	0001	1010
0010	1001	0111	1100

Note that in this example we could not have chosen the vector 0001 as the coset leader of the final row, even though it meets the critedia of having minimal weight (1), because the vector was already present in the array We could, however, have chosen it as the first coset leader and constructed a different standard array.

Decoding via standard array

Error vectors which will be corrected are precisely coset leaders!
In practice, this decoding method is too slow and requires too much memory.

A Standard Array for a $(6, 3)$ code

$$\begin{aligned} \boldsymbol{C} &= \{000000, 100110, 010101, 001011, \\ &\quad 110011, 101101, 011110, 111000\} \end{aligned}$$

000000	100110	010101	001011	110011	101101	011110	111000
000001	100111	010100	001010	110010	101100	011111	111001
000010	100100	010111	001001	110001	101111	011100	111010
000100	100010	010001	001111	110111	101001	011010	111100
001000	101110	011101	000011	111011	100101	010110	110000
010000	110110	000101	011011	100011	111101	001110	101000
100000	000110	110101	101011	010011	001101	111110	011000
100001	000111	110100	101010	010010	001100	111111	011001