SOLUTIONS TO ASSIGNMENT #1 FOR CODING THEORY

I. Let $C \subset \mathbb{B}^n$ be a binary *linear* code. Recall that $d(C) := \min_{v,w \in C} d(v,w)$.

(1) For any $v, w \in \mathbb{B}^n$, show that $d(v,w) = d(v-w, \underline{0})$.

Solution. Let $v := (v_j)_{j=1,\cdots,n}$ and $w := (w_j)_{j=1,\cdots,n}$. Then $d(v,w)$ is the number of j's such that $v_j \neq w_j$. Since the jth bit of $v = w$ is 1 if and only if $v_j \neq w_j$, it follows that $d(v,w) = d(v-w, \underline{0})$ by comparing the definitions of both sides of the equation. □

(2) Show that $d(C) := \min_{v \in C} d(v, \underline{0})$.

Solution. Assume that $v_1, v_2 \in C$ such that $d(v_1, v_2) = d(C)$. Then by the previous part, we have

$$d(C) = d(v_1 - v_2, \underline{0}) \geqslant \min_{v \in C} d(v, \underline{0}),$$

since $v_1 - v_2 \in C$ by linearity. One obtains the opposite inequality

$$d(C) := \min_{v,w \in C} d(v,w) \leqslant \min_{v \in C} d(v, \underline{0})$$

because $\underline{0} \in C$ by linearity. □

(3) Let $H : \mathbb{B}^n \to \mathbb{B}^k$ be a check matrix for C; i.e., $C = \ker(H)$. Show that He_i is the ith column vector of H for any $i = 1, 2, \cdots, n$.

Solution. Define δ_{ai} denote the ath entry of e_i; i.e., $\delta_{ai} = 1$ if $a = i$ and $\delta_{ai} = 0$ if $a \neq i$. Let $H := (H_{ab})$ where $a = 1, \cdots, k$ and $b = 1, \cdots, n$. Then the ath entry of He_i is

$$\sum_{b=1}^{n} H_{ab} \delta_{bi} = H_{ai}.$$

This is precisely the ath entry of the ith column vector of H. □

(4) Show that $d(C)$ is the minimal number of linearly dependent column vectors in H. (*Hint:* **I**.(3) could be useful. Note that $v = (v_1, \cdots, v_n) = v_1 e_1 + \cdots + v_n e_n$. Note also that $H(v+w) = Hv + Hw$ for any $v, w \in F^n$.)

Solution. Let $v = (v_1, \cdots, v_n) = v_1 e_1 + \cdots + v_n e_n$, as in the hint. Then v is a codeword if and only if $Hv = \underline{0}$; i.e., we have

$$Hv = H(v_1 e_1 + \cdots + v_n e_n) = v_1 (He_1) + \cdots + v_n (He_n).$$

Assume that $d(v, \underline{0}) = d$; i.e., exactly d bits among v_j's are not zero. Then by **I**.(3), the above displayed equation is a linearly dependency involving precisely d column vectors of H. Now we are done by **I**.(2). □

II. Let $C \subset \mathbb{B}^n$ be a binary code, not necessarily linear.

(1) Show that if C is r-perfect for some integer r, then $d(C) = 2r + 1$.

Solution. To simplify the notation, put $d := d(C)$. We show in two steps that $d \geqslant 2r + 1$ and $d \leqslant 2r + 1$.

First, you may show $d \geqslant 2r+1$ using that C, being r-perfect, can correct all errors of weight at most r. Alternative, you can proceed as follows. Choose any $v, v' \in C$ such that $d(v, v') = d$. By r-perfectness, we know $d = d(v, v') > r$ (because otherwise $D_r(v)$ contains two distinct codewords v and v'). Write $v - v' = e_{i_1} + e_{i_2} + \cdots + e_{i_d}$ for $i_1, \cdots, i_d$ (i.e., they are the slots where v and v' have different bits). Put $w := v + e_{i_1} + \cdots + e_{i_r}$ (which is possible since $r < d$). Then clearly from the construction we have $d = d(v, v') = d(v, w) + d(w, v')$ and $d(v, w) = r$. Now by r-perfectness, we have that $d(v', w) > r$, hence $d > 2r$.

Now, let us show that $d \leqslant 2r + 1$. Note that $n > r$; if $n \leqslant r$ then we have $d(v, v') \leqslant n \leqslant r$ for any two vectors (hence any two codewords) $v, v' \in \mathbb{B}^n$ so it is impossible to have an r-perfect code. (Note that a code should necessarily have more than one element.) Choose any $v \in C$ and $w \in \mathbb{B}^n$ such that $d(v, w) = r$. Choose i such that $v_i = w_i$ where v_i and w_i are respectively the ith bits of v and w. (This is possible because $n > r$.) Let $w' := w + e_i$ where i is as above. Then clearly $d(v, w') = r + 1$. By r-perfectness, there is a (unique) $v' \in C$ such that $d(v', w') \leqslant r$. Now we have by triangle inequality

$$d(v, v') \leqslant d(v, w') + d(v', w') \leqslant (r + 1) + r$$

We have created v, v' such that $d(v, v') \leqslant 2r + 1$, hence $d \leqslant 2r + 1$. □

(2) Find an example of a binary linear code C such that $d(C) = 3$ but C is not 1-perfect.

Solution. Here is one such example, though there can be many others. Let $C \in \mathbb{B}^6$ be the triple parity check $(6, 3)$-code that we defined in the class; i.e., $C = \{(a, b, c, x, y, z) : \ x = b + c, \ y = c + a, \ z = a + b)\}$. We saw in the lecture that $d(C) = 3$. One can show that C is not 1-perfect in any of the following ways: (1) the dimension of C is 3 but $2^6 \neq 2^3(1 + 6)$; or (2) the vector $(1, 0, 0, 1, 0, 0)$ does not belong to $D_1(v)$ for any $v \in C$. □

III. We define a $(8, 4)$-code $\mathrm{Ham}'(3)$ by adding an "overall parity check bit" to $\mathrm{Ham}(3)$; i.e. a codeword of $\mathrm{Ham}'(3)$ is of the form $(v_1, \cdots, v_7, \sum_{i=1}^{7} v_i)$ where $(v_1, \cdots, v_7)$ is a codeword of $\mathrm{Ham}(3)$.

(1) Write down a generator matrix and a check matrix in standard form. (*Hint*: It is easier to find a generator matrix in standard form.)

Solution. Let G' and H' be the generator and check matrices for $\mathrm{Ham}'(3)$ in standard form. Then,

$$G' = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}, \qquad H' = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

One can obtain them in the following two different ways:
First, following the hint I gave, if $(v_1, \cdots, v_7)$ is a codeword for Ham(3), then there is (x_1, x_2, x_3, x_4) such that $(v_1, \cdots, v_7)^T = G \cdot (x_1, x_2, x_3, x_4)^T$, where G is the generator matrix for Ham(3). Writing out v_i in terms of $(x_1, \cdots, x_4)$, you can find that $\sum_{i=1}^{7} v_i = x_1 + x_3 + x_4$. This gives G' above, and since it's in standard from we obtain H' as well.

Alternatively, since we added a bit v_8 so that $(v_1, \cdots, v_8)$ is a codeword if and only if $H_3(v_1, \cdots, v_7)^T = 0$ and $\sum_{i=1}^{8} v_i = 0$, the following check matrix (in non-standard form) works:

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Note that the upper 3×7-block is exactly H_3. Now we perform elementary *row operations* to reduce it to standard form: namely, subtract each of first three rows from the last row. So we obtain H', and G' can be obtained from H'. □

(2) Show that the minimal distance of $\mathrm{Ham}'(3)$ is 4. (*Hint*: This can be easily deduced from the fact that the minimal distance of Ham(3) is 3, which you can use without proof.)

Solution. One can solve this by showing that (1) all column vectors of H' are non-zero, (2) any two column vectors of H' are distinct, (3) any three column vectors are linearly independent, (4) there are four column vectors that are linearly dependent (i.e. there is a codeword with weight 4). *Note that* just exhibiting a codeword with weight 4 is not enough, and the real hard work of the solution is to show that it is impossible to have 3 linearly dependent column vectors.

Alternatively, one can follow the hint. In Ham(3) a codeword $(v_1, \cdots, v_7)$ "nearest" to $(0, \cdots, 0)$ has precisely three 1's in the entries, so the last parity bit we add is 1. Therefore $(v_1, \cdots, v_7, 1)$ has distance 4 from $(0, \cdots, 0)$. We're done since the distance from $(0, \cdots, 0)$ is either increased or not changed by adding a parity bit. (More precisely, for any a codeword $(v_1, \cdots, v_7)$ of Ham(3) with distance ≥ 4 from $(0, \cdots, 0)$, the distance from $(v_1, \cdots, v_7, \sum_1^7 v_i)$ to $(0, \cdots, 0)$ will be ≥ 4.) □

(3) Is $\mathrm{Ham}'(3)$ r-perfect for any r?

Solution. No. One can see this in one of the following ways.

First, if $\mathrm{Ham}'(3)$ were r-perfect for some r, then the minimal distance would be $2r+1$, in particular an odd number. But the minimal distance of $\mathrm{Ham}'(3)$ is 4.

Alternatively, one can show this by counting. Since $2^8 > 2^4(1+8)$, it follows that $\mathrm{Ham}'(3)$ is not 1-perfect. And it is not 2-perfect because $2^8 < 2^4(1+8+28)$. Now, for any $r \geqslant 2$, we have

$$2^8 < 2^4(1+8+28) = 2^4\Big(\tbinom{8}{0}+\tbinom{8}{1}+\tbinom{8}{2}\Big) \leqslant 2^4\Big(\sum_{i=0}^{r}\tbinom{8}{i}\Big).$$

Therefore, $\mathrm{Ham}'(3)$ is not r-perfect for any r. □

(4) Let p be a probability of an error occurring in a single bit during transmission, and assume $p < 1/2$. We will encode a message of 4000 bits using $\mathrm{Ham}'(3)$ and transmit them. For error processing, we'll choose to correct as many errors as possible (and generate an error message when an error cannot be corrected). Find (i) the probability of correct transmission, i.e. the probability that the corrected message is same as the sent message; and (ii) the probability that all the errors are either corrected or detected. (You do not have to simplify the expressions.)

Solution. Because the minimal distance is 4, $\mathrm{Ham}'(3)$ can correct at most one error in each block, and detect all the weight 2 error patterns at the same time.

Since we are encoding 4000 bits, we obtain 1000 blocks of size 4, and apply our generator matrix to obtain 1000 blocks of size 8.

To compute the probability of correct transmission, one needs to count the probability where there is at most one error in each block:

$$\big((1-p)^8+\tbinom{8}{1}p(1-p)^7\big)^{1000}.$$

To compute the probability that all the errors are either corrected or detected, one needs to count the probability where there are at most two errors in each block:

$$\big((1-p)^8+\tbinom{8}{1}p(1-p)^7+\tbinom{8}{2}p^2(1-p)^6\big)^{1000}$$

□

IV. Let $H_{4,2} : \mathbb{B}^{15} \to \mathbb{B}^8$ be defined by the following matrix.

$$H_{4,2} := \begin{pmatrix} 1&1&1&1&0&1&1&1&1&0&1&1&1&1&0\\ 1&0&1&0&0&1&0&1&0&0&1&0&1&0&0\\ 1&1&0&0&0&1&1&0&0&0&1&1&0&0&0\\ 1&0&0&0&1&1&0&0&0&1&1&0&0&0&1\\ 1&1&1&1&0&1&0&1&1&0&0&1&0&0&0\\ 0&1&1&1&1&0&1&0&1&1&0&0&1&0&0\\ 0&0&1&1&1&1&0&1&0&1&1&0&0&1&0\\ 1&1&1&0&1&0&1&1&0&0&1&0&0&0&1 \end{pmatrix}$$

We denote by $\mathrm{BCH}(4,2)$ the code defined by $H_{4,2}$; i.e., $\mathrm{BCH}(4,2) := \ker H_{4,2}$.

(1) Only by adding one row to another and swapping two rows on $H_{4,2}$, find a check matrix $H'_{4,2}$ of the following form which defines the same code as :

$$H'_{4,2} := \begin{pmatrix} 0_{(8-m)\times(15-m)} & 0_{(8-m)\times m} \\ A & I_m \end{pmatrix},$$

where m is a suitable integer, A is some $m \times (15-m)$ matrix, and $0_{a\times b}$ is the $a \times b$ matrix with all entries zero.

Solution. Here is an algorithm one may use.

(a) By switching rows if necessary, make sure that the last (15th) entry of the last (8th) row is 1.
(b) Add the last row to all the row with the last entry 1 (so that all the rows except the last row has 0 in the last entry).
(c) By switching rows if necessary, make sure that the 14th entry of the 7th row is 1.
(d) Add the 7th row to all the rows with the 14th entry 1 (so that all the rows except the7th row has 0 in the 14th entry).
(e) Repeat the above.

In general, one might need to switch columns in steps (1) and (3), but for us it is not needed.

The final answer is

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix},$$

and $m = 0$ □

(2) What is the dimension of BCH$(4,2)$? (*Hint*: Rank-nullity theorem.)

Solution. Note that the rank of $H'_{4,2}$ is visibly 8, so by rank-nullity theorem the dimension of BCH$(4,2)$ is 7. □

(3) Write down the generator matrix in standard form.

Solution.

$$G'_{4,2} := \begin{pmatrix} I_7 \\ A \end{pmatrix},$$

where I_7 is a 7×7 diagonal matrix with diagonal entries all 1. □

(4) Find the minimal distance of BCH$(4,2)$. (You may use **I.**)

Solution. First note that the first column of $H'_{4,2}$ is of weight 4, so the first, eighth, ninth, tenth, twelfth columns add up to zero. This shows that the minimal distance is $\leqslant 5$. (*Note that* this is not enough to show that the minimal distance is 5.)

Now, let us show that the minimal distance is 5 by verifying that any smaller number of columns cannot be linearly dependent. Let me just sketch how one can check this. Clearly, there is no zero column and any two columns are distinct, so the minimal distance is at least 3.

Now, compute the sum of any two columns in A. (There are 21 choices.) First, check none of them are of weight $\leqslant 2$. (If this happens, you can form linear dependence relations with $\leqslant 4$ vectors, using columns in I_8.) And them, check that it is possible to obtain any weight 1 vector by adding a column vector of A and one of 21 vectors (obtained from two columns of A).

Finally, it remains to rule out the possibility that 4 column vectors in A adds up to zero. But indeed, one can show an even stronger statement; namely, all 7 column vectors of A are linearly independent. To check this, one may perform column operation to simplify A. If we will eliminate all non-zero entries in the strictly upper triangular entries of A by a similar algorithm to **IV**.(1), we obtain:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

The columns of this matrix are visibly linearly independent. □

(5) Is $\mathrm{BCH}(4,2)$ r-perfect for any r?

Solution. Since the minimal distance is 5, the only candidate for r is 2. On the other hand, one can easily check that

$$2^{15} \neq 2^7 \left(\binom{15}{0} + \binom{15}{1} + \binom{15}{2} \right).$$

Alternatively, one can proceed as the "second solution" of **III**.(3) without using the minimal distance. □

V. With no further mention, all poiynomials here are "binomial polynomials" (i.e.,the coefficients are binary numbers).

(1) For any $a \in \mathbb{B}$ and any $f(X) \in \mathbb{B}[X]$, show that $X - a$ divides $f(X)$ if and only if $f(a) = 0$. (*Hint*: First, show that the remainder of $f(X)$ modulo $X - a$ is $f(a)$. Then, argue that this implies the claim.)

Solution. Let $f(X) \in \mathbb{B}[X]$ and choose $a \in \mathbb{B}$. By division algorithm, one can find $q(X) \in \mathbb{B}[X]$ and $r \in \mathbb{B}$ such that

$$f(X) = q(X)(X - a) + r.$$

If $f(a) = 0$, then $q(a)(a - a) + r = r = 0$ so $(X - a) divides$f(X).$If f$(X-a) divides $f(X)$ (i.e., if $r = 0$), then $f(a) = q(a)(a - a) = 0$. □

(2) Show that X^2 and $X^2 + 1$ are reducible, and $X^2 + X + 1$ is irreducible.

Solution. Note that $X^2 = X \cdot X$ and $X^2 + 1 = (X+1)(X+1)$, so they are reducible.

Let $f(X) := X^2+X+1$. If $f(X)$ were reducible, then $f(X) = g(X)h(X)$ for some $g(X), h(X) \in \mathbb{B}[X]$ where each of them are of degree > 0. This is only possible when both $g(X)$ and $h(X)$ are of degree 1. But there are only two degree-1 binary polynomials; namely, X and $X+1$. On the other hand, $f(0) = f(1) = 1$, so by the previous part neither X nor $X+1$ divides $f(X)$. □

(3) Show that $X^4 + X + 1$ and $X^4 + X^3 + 1$ are irreducible.

Solution. Let $f(X) \in \mathbb{B}[X]$ be any polynomial of degree 4. If $f(X)$ is reducible, then $f(X) = g(X)h(X)$ for some $g(X), h(X) \in \mathbb{B}[X]$ where each of them are of degree > 0. Then either $g(X)$ or $h(X)$ has degree 1 or 2. Therefore, a degree-4 binary polynomial $f(X)$ is irreducible if and only if none of binary polynomials with degree 1 and 2 divides $f(X)$, if and only if none of *irreducible* polynomials with degree 1 and 2 divides $f(X)$. Note that there are only two degree-1 binary polynomials (X and $X+1$) and four degree-2 binary polynomials (X^2, X^2+1, X^2+X, and X^2+X+1), and among them only X, $X+1$, and X^2+X+1 are irreducible.

Now, for $f(X) = X^4 + X + 1$ or $f(X) = X^4 + X^3 + 1$, we just need to check that $f(0) = f(1) = 1$ and that $X^2 + X + 1$ does not divide $f(X)$ (by performing long division). □

(4) Find all the irreducible binary polynomial of degree 4. (*Hint:* There are exactly three.)

Solution. By the solution of the previous part, we just need to find all degree-4 binary polynomials $f(X)$ such that $f(0) = f(1) = 1$ and $f(X)$ is not divisible by X^2+X+1. Let $f(X) = X^4 + aX^3 + bX^2 + cX + d$ be any binary polynomial of degree 4. Then $f(0) = f(1) = 1$ is satisfied if and only if $d = 0$ and $a+b+c = 1$. There are exactly four such binary polynomials; namely, X^4+X^+1, X^4+X^2+1, X^4+X^3+1, and $X^4+X^3+X^2+X+1$. We already checked, in the previous part, that X^4+X^+1 and X^4+X^3+1 are irreducible. We can also check that $X^4+X^3+X^2+X+1$ is not divisible by X^2+X+1 (by performing long division). Finally, $X^4+X^2+1 = (X^2+X+1)^2$ is reducible. □

(5) Compute α^{10} in $\mathbb{B}[\alpha]/\alpha^4 + \alpha + 1$.

Solution.

$$\alpha^{10} = (\alpha^4)^2\alpha^2 = (\alpha+1)^2\alpha^2 = (\alpha^2+1)\alpha^2 = \alpha^4 + \alpha^2 = \alpha^2 + \alpha + 1$$

In the above equation, we just used $\alpha^4 = \alpha + 1$.

Alternatively, one can perform long division and find the remainder of α^{10} modulo $\alpha^4 + \alpha + 1$. □

(6) Compute β^{10} in $\mathbb{B}[\beta]/\beta^4 + \beta^3 + 1$.

Solution.

$$\begin{aligned}\beta^{10} &= (\beta^4)^2\beta^2 = (\beta^3+1)^2\beta^2 = \beta^8+\beta^2 = (\beta^4)^2+\beta^2 = (\beta^3+1)^2+\beta^2\\ &= \beta^6+\beta^2+1 = \beta^4\beta^2+\beta^2+1 = (\beta^3+1)\beta^2+\beta^2+1\\ &= \beta^5+1 = \beta^4\beta+1 = (\beta^3+1)\beta+1 = \beta^4+\beta+1 = \beta^3+\beta\end{aligned}$$

In the above equation, we just used $\beta^4 = \beta^3+1$.

Alternatively, one can perform long division and find the remainder of α^{10} modulo $\alpha^4+\alpha+1$. □

(7) Find the multiplicative inverse of $\alpha^3+\alpha+1$ in $\mathbb{B}[\alpha]/\alpha^4+\alpha+1$.

Solution. Euclid's algorithm gives you

$$1 = (\alpha^2+\alpha+1)(\alpha^4+\alpha+1)+(\alpha^3+\alpha^2)(\alpha^3+\alpha+1)$$

so the multiplicative inverse of $\alpha^3+\alpha+1$ is $\alpha^3+\alpha^2$.

If you cannot remember the formulae for Euclid's algorithm (like me), I'd do the following:

$$\begin{aligned}\alpha^4+\alpha+1 &= \alpha(\alpha^3+\alpha+1)+\alpha^2+1 \quad \text{...(1)}\\ \alpha^3+\alpha+1 &= (\alpha+1)(\alpha^2+1)+\alpha \quad \text{...(2)}\\ \alpha^2+1 &= \alpha\cdot\alpha+1 \quad \text{...(3)}\end{aligned}$$

Now, we rewrite the above as follows:

$$\begin{aligned}1 &= \alpha^2+1+\alpha\cdot\alpha \quad \text{...by (3)}\\ &= \alpha^2+1+\alpha\Big(\alpha^3+\alpha+1+(\alpha+1)(\alpha^2+1)\Big) \quad \text{...by substituting (2)}\\ &= (\alpha^2+1)+\alpha(\alpha^3+\alpha+1)+(\alpha^2+\alpha)(\alpha^2+1) \quad \text{...distribute the big parenthesis}\\ &= (\alpha^2+\alpha+1)(\alpha^2+1)+\alpha(\alpha^3+\alpha+1) \quad \text{...put together } \alpha^2+1 \text{ terms}\\ &= (\alpha^2+\alpha+1)\Big(\alpha^4+\alpha+1+\alpha(\alpha^3+\alpha+1)\Big)+\alpha(\alpha^3+\alpha+1) \quad \text{...by substituting (1)}\\ &= (\alpha^2+\alpha+1)(\alpha^4+\alpha+1)+(\alpha^3+\alpha^2+\alpha)(\alpha^3+\alpha+1)+\alpha(\alpha^3+\alpha+1)\\ &\qquad \text{...distribute the big parenthesis}\\ &= (\alpha^2+\alpha+1)(\alpha^4+\alpha+1)+(\alpha^3+\alpha^2)(\alpha^3+\alpha+1) \quad \text{...put together } \alpha^3+\alpha+1 \text{ terms}\end{aligned}$$

□