Question 5

To clarify the question, we rephrase the erroneous implementation of the McEliece KEM:

Algorithm 1: Key generation

Sample a random Goppa code $\mathcal{C}(g(x), \{\alpha_j\}_{j=1}^n)$ with parity check matrix H;
return $pk = H, sk = \mathcal{C}$

Algorithm 2: Encapsulation

Sample a random e such that $|e| = t$;
$c \leftarrow He$;
$K \leftarrow \text{KDF}(m, c, 0)$ **return** (c, K);

Recall that the parity check matrix is defined by its individual entries:

$$H_{i,j} = \frac{\alpha_j^i}{g(\alpha_j)}$$

We can first recover individual values of α_j using adjacent values of the same column. For each of $1 \leq j \leq n$:

$$\frac{H_{1,j}}{H_{0,j}} = \frac{\alpha_j}{g(\alpha_j)} \frac{g(\alpha_j)}{1} = \alpha_j$$

In addition, we can recover $g(\alpha_j)$ for $1 \leq j \leq n$ by inverting $H_{0,j}$:

$$H_{0,j}^{-1} = g(\alpha_j)$$

From the construction of the Goppa code we know that the degree t of the polynomial $g(x)$ is less than the number of α_j's, which means that with n points on this polynomial we can uniquely determined the polynomial $g(x)$. Thus, we have fully recovered all parameters of the Goppa code from its parity check matrix.

Having recovered the Goppa code $\mathcal{C}$ is equivalent to posessing the secret key since the random permutation and row reduction are both omitted (which is equivalent to having $S = I_{n-k}$ and $P = I_n$). In other words, we have recovered the secret key from the public key, which trivially breaks the KEM.