Question 3

We are to prove that Elias Omega code is prefix-free, i.e. no Elias codeword of any $n \in Z^+$ can be a prefix of any other codeword for $m \neq n \in Z^+$. A prefix-free codeword implies that, for any two codewords *s1* and *s2* where len(*s1*) < len(*s2*), *s1* is not a prefix of *s2*.

The Elias Omega code consists of two parts, the length encoding (coloured blue) and the integer encoding (coloured red):

E.g. Elias(123) = 0000101111011

To encode an integer into Elias Omega code, we first generate the binary representation of the integer (the red part). This component always starts with a 1-bit. Then we encode the length of the binary representation after a decrement by 1 and flip the first bit, then the length of that after decrementing by 1 and flip the first bit, all the way until we reach an encoding of length 1. All length components always start with a 0-bit.

Suppose Elias Omega code is not prefix-free, there exists two Elias Omega codewords for integers n (s1) and m (s2), in which *s1* is the prefix of s2, therefore len(*s1*) <= len(*s2*), and n < m.

$$s1 = b_1 b_2 b_3 \ldots b_j b_{j+1} b_{j+2} \ldots b_{j+x} = d_n n_2$$

$$\text{and } s2 = b'_1 b'_2 b'_3 \ldots b'_k b'_{k+1} b'_{k+2} \ldots b'_{k+y} = d_m m_2$$

It is impossible for Elias Omega codes of the same length to be identical, since the binary representation component is unique for all numbers. Therefore in the case where len(s1) = len(s2) and n < m, Elias Omega codes are prefix-free by contradiction.

In the scenario where len(s1) < len(s2), this implies that len(n_2) < len(m_2), and len(d_m) <= len(d_n). A total of 4 cases will arise:

1. len(d_n) = len(d_m)

Since len(n_2) < len(m_2), ($b_1 b_2 b_3 \ldots b_j$) ≠ ($b'_1 b'_2 b'_3 \ldots b'_k$), s1 is not a prefix of s2.

2. len(d_n) + len(n_2) = len(d_m)

This indicates that s1 is equal to the length encoding of m_2, which is impossible. This is due to the fact that the binary representation component n_2 can't be present in the length component of m_2 because it would have its MSB flipped to a 0 to encode the length. Therefore s1 is not a prefix of s2.

3. len(d_n) + len(n_2) < len(d_m)

This is similar to case 2 where we will encounter the binary representation component n_2 which is present in the length component of m_2. This is impossible because it would have its MSB flipped to a 0 to encode the length. Therefore s1 is not a prefix of s2.

4. len(d_n) + len(n_2) > len(d_m)

This is the mirror of cases 2 and 3 where we encounter the 1-bit in the binary representation component m_2 which is present in the length component of n_2. This is impossible because it would have its MSB flipped to a 0 to encode the length. Therefore s1 is not a prefix of s2.

In all 5 cases, we have proved that Elias Omega code is prefix-free by contradiction.