Block ciphers usually process 64 or 128-bit blocks at a time. To illustrate how their modes of operation work, we can use instead a pseudo-random permutation that operates on the 26 letters of the English alphabet:

| 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 |
|--------------------------|--------------------------|
| A B C D E F G H I J K L M N O P Q R S T U V W X Y Z                   |

As the XOR operation is not defined on the set \{A, \ldots, Z\}, we replace it here during encryption with modulo-26 addition (e.g., C \oplus D = F and Y \oplus C = A).

(a) Encrypt the plaintext “TRIPOS” using:

(i) electronic codebook mode; [2 marks]

(ii) cipher-block chaining (using IV \(c_0 = K\)); [4 marks]

(iii) output feedback mode (using IV \(c_0 = K\)). [4 marks]

(b) Decrypt the ciphertext “BSMILVO” using cipher-block chaining. What operation should replace XOR? [4 marks]

(c) Your opponent is allowed to send you two plaintext messages \(M_0\) and \(M_1\), each \(n\) letters long. You now pick a new private key \(K\), resulting in a new pseudo-random permutation \(E_K : \{A, \ldots, Z\} \leftrightarrow \{A, \ldots, Z\}\). You also pick uniformly at random a private bit \(b \in \{0,1\}\) and return a ciphertext \(C = c_0c_1 \ldots c_n\), namely the message \(M_b\) encrypted with cipher-block chaining using the fresh \(E_K\). Finally, your opponent has to guess your bit \(b\).

Approximately how large must \(n\) be at least for your opponent to have a greater than 75% chance of guessing \(b\) correctly? Outline a strategy that your opponent can use to achieve this. [6 marks]