Many time pad:

Consider the one-time pad encryption where the ciphertext is the XOR of the message and the key. Reusing the key makes the system insecure, as we see in this assignment.

Every ciphertext is the result of encrypting an English sentence using two keys: K_1 and K_2.
K_1 has a fixed length of 16 bytes, denoted as $|K_1|$ = 16 bytes.
The length of K_2, denoted as $|K_2|$, equals the difference of the length of the longest plaintext in the set and16 bytes, i.e., $|K_2|$ = $|M_{max}|$ - 16 bytes, where $|M_{max}|$ is the length of the longest plaintext.

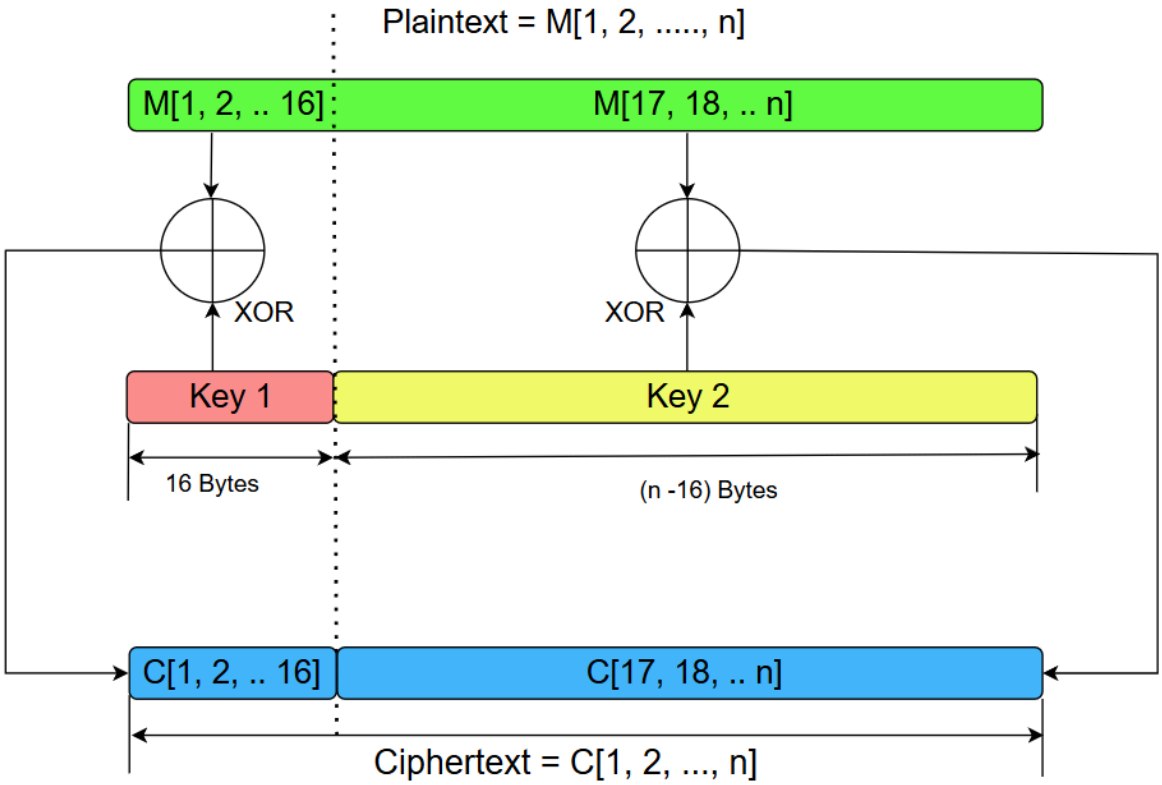
Pseudocode for encryption:

```
ENCRYPT( Messages[12][ ] ):

        Messages[12][ ] := Array of 12 messages
        Ciphertexts[12][ ] := Array to store all 12 ciphertexts

        // n is equal to the maximum length among the twelve messages.
        n := MAX( length of Messages[1], …… , length of Messages[12])

        key_length1 := 16
        key_length2 := n - 16

        // Generate random keys
        K1 := generate random bytes of length key_length1
        K2 := generate random bytes of length key_length2

        FOR i FROM 1 to 12:

                // Encrypt first 16 bytes of the Messages[i]
                FOR j FROM 1 TO 16:
                   Ciphertexts[i][j] = Messages[i][j] ⊕ K1[j]

                // Encrypt remaining bytes of the Messages[i]
                FOR j FROM 17 TO length of Messages[i]
                   Ciphertexts[i][j] = Messages[i][j] ⊕ K2[j - 16]

        RETURN Ciphertexts[12][ ]
```

Provided materials.

- You are provided with four sets, each containing twelve ciphertexts.
- Each of the four sets has its own unique pair of keys, (K_1, K_2).
- A Google Sheet will be provided, containing information about which of the four sets is assigned to each student. You should only solve for the set assigned to you.
- You are also provided with a file named `Dictionary.txt`, which contains a list of possible strings for the first 16 bytes of the plaintext. Utilize this dictionary file to crack the first 16 bytes of the plaintext.

Write a program that outputs the secret key (K_1, K_2) and outputs the plaintext corresponding to the given ciphertext and stores it in a text file.



Deliverables:

1. Your program file should be named as Prog_Asgn_1_<Roll_No>.cpp or Prog_Asgn_1_<Roll_No>.py
2. Text file containing the plaintext and key should be named as Prog_Asgn_1_<Roll_No>.txt
3. Combine your program file and text file into a zip archive file and name it as Prog_Asgn_1_<Roll_No>.zip. Upload this zip file.

Hints:

- Observe the result when you XOR space character with a letter.
- Remember what marks the end of a sentence.
- For the first 16 bytes, you are given ciphertexts, a set of possible plaintexts and you already know what the key space is, look into different ways of using them. Sometimes, changing your approach might give you quicker results!