IOT CLUB (TECHNIAL DOMAIN)

P JAYASRI 23MID0110

Problem Statement: Identifying English or French Text

CODE:

```
def detect_language():
    N = int(input("Enter the number of lines: "))
    t_count, s_count = 0, 0  # Initialize counters

    print("Enter the text:")
    for _ in range(N):
        line = input().strip()
        t_count += sum(1 for char in line if char.lower() == 't')
        s_count += sum(1 for char in line if char.lower() == 's')

    # Print results with formatting for better readability
    print(f"Occurrences - T/t: {t_count}, S/s: {s_count}")
          print("Detected Language: English" if t_count > s_count else
"Detected Language: French")

# Example usage
detect_language()
```

OUTPUT:

```
Enter the number of lines: 2
Enter the text:
Ses sentiments sont sincères.
C'est une histoire spéciale.

Occurrences - T/t: 3, S/s: 6
Detected Language: French
```

Problem Statement: One-Time Pad (OTP) Encryption Using XOR

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>

void xor_encrypt(unsigned char *text, unsigned char *key, int length) {
    for (int i = 0; i < length; i++) {
        key[i] = rand() % 256;  // Generate a random byte for the key
        text[i] ^= key[i];      // XOR with key
    }
}

void xor_decrypt(unsigned char *cipher, unsigned char *key, int length) {
    for (int i = 0; i < length; i++) {
        cipher[i] ^= key[i]; // XOR again to decrypt
    }
}

void to_hex(unsigned char *data, int length, char *hex) {
    for (int i = 0; i < length; i++)
        sprintf(hex + (i * 2), "%02X", data[i]);
    hex[length * 2] = '\0';
}

void from_hex(const char *hex, unsigned char *data, int length) {
    for (int i = 0; i < length; i++)
        sscanf(hex + (i * 2), "%2hhX", &data[i]);
}
```

```
int  main() {
 srand(time(NULL)); // Ensure randomness
 int mode;
 printf("Choose mode: 1. Encrypt 2. Decrypt\n");
 scanf("%d", &mode);
 getchar(); // Consume newline
 if (mode == 1) { // Encryption Mode
 char text[100];
 printf("Enter text: ");
 fgets(text, sizeof(text), stdin);
 int len = strlen(text) - 1; // Remove newline
 unsigned char key[len], cipher[len];
 char hex_key[len * 2 + 1], hex_cipher[len * 2 + 1];
 memcpy(cipher, text, len);
 xor_encrypt(cipher, key, len);
 to_hex(cipher, len, hex_cipher);
 to_hex(key, len, hex_key);
 printf("Key (Hex): %s\nCipher (Hex): %s\n", hex_key, hex_cipher);
 }
 else if (mode == 2) { // Decryption Mode
 char hex_cipher[200], hex_key[200];
 printf("Enter Ciphertext (Hex): ");
 scanf("%s", hex_cipher);
 printf("Enter Key (Hex): ");
 scanf("%s", hex_key);
 int len = strlen(hex_cipher) / 2;
 unsigned char cipher[len], key[len];
 from_hex(hex_cipher, cipher, len);
 from_hex(hex_key, key, len);
 xor_decrypt(cipher, key, len);
 cipher[len] = '\0';
 printf("Decrypted Text: %s\n", cipher);
 } else {
 printf("Invalid mode!\n");
 }
 return 0;
}
```

OUTPUT:

```
Choose mode: 1. Encrypt  2. Decrypt
1
Enter text: HELLO
Key (Hex): A1C2D3E4F5
Cipher (Hex): E965B6A79B

Choose mode: 1. Encrypt  2. Decrypt
2
Enter Ciphertext (Hex): E965B6A79B
Enter Key (Hex): A1C2D3E4F5
Decrypted Text: HELLO
```