main.c

Run

```c
#include <stdio.h>
#include <stdint.h>
#include <string.h>

// Simulated block cipher: simple XOR with key (for demo)
uint64_t block_cipher(uint64_t input, uint64_t key) {
    return input ^ key;
}

// CBC-MAC implementation for fixed block size = 64 bits
uint64_t cbc_mac(uint64_t key, uint64_t *blocks, int block_count) {
    uint64_t iv = 0;
    uint64_t state = iv;

```

Output

Clear

```
Original one-block message X:     0x1122334455667788
CBC-MAC of X (T):                 0xb48796e1f0c3d22d

Forged two-block message:
Block 1: 0x1122334455667788
Block 2: 0xa5a5a5a5a5a5a5a5
CBC-MAC of forged message:        0xb48796e1f0c3d22d

✅ Attack successful: MAC matches original T!


=== Code Execution Successful ===
```

main.c

Run

```
        );
    printf("\nForged two-block message:\n");
    printf("Block 1: 0x%llx\n", forged[0]);
    printf("Block 2: 0x%llx\n", forged[1]);
    printf("CBC-MAC of forged message:
        0x%llx\n", forged_mac);

    // Compare
    if (forged_mac == T)
        printf("\n✅ Attack successful: MAC
            matches original T!\n");
    else
        printf("\n❌ Attack failed: MAC does not
            match T.\n");

    return 0;
}
```

Output

Clear

```
Original one-block message X:     0x1122334455667788
CBC-MAC of X (T):                 0xb48796e1f0c3d22d

Forged two-block message:
Block 1: 0x1122334455667788
Block 2: 0xa5a5a5a5a5a5a5a5
CBC-MAC of forged message:        0xb48796e1f0c3d22d

✅ Attack successful: MAC matches original T!


=== Code Execution Successful ===
```