Programiz C Online Compiler

Programiz PRO >

main.c

Share Run

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

#define BLOCK_SIZE 8   /* block size in bytes (example) */
#define BUF_SIZE 128   /* must be >= original plaintext length + BLOCK_SIZE
    padding */

/* pad buffer with "1 bit then zeros" as in your original style.
   Returns new length after padding, or -1 on error (not enough space). */
static int pad80(unsigned char *buf, int cur_len, int buf_capacity) {
    if (cur_len < 0 || buf_capacity <= 0) return -1;
    int pad_len = BLOCK_SIZE - (cur_len % BLOCK_SIZE);
    if (pad_len == 0) pad_len = BLOCK_SIZE;
    if (cur_len + pad_len > buf_capacity) return -1; /* not enough space */

    buf[cur_len] = 0x80;
    for (int i = 1; i < pad_len; ++i) buf[cur_len + i] = 0x00;
    return cur_len + pad_len;
}

/* Simple XOR "block cipher" primitive for demonstration.
   key must be BLOCK_SIZE bytes (if shorter it will be repeated/truncated). */
static void xor_block(unsigned char *block, const unsigned char *key, int
    block_len) {
```

Output

Clear

```
ECB Encrypted (hex): 39111A16431B16540C59070010064519080A00040417451202OB53000D11170D1D0D1
    A0A0D5CE574
CBC Encrypted (hex): 39111A16431B165435481D16531D534D3D421D12570A165F3F494E125A1B015222445
    4185747E426
CFB Encrypted (hex): 39111A16431B165435481D16531D534D3D421D12570A165F3F494E125A1B015222445
    4185747E426


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