

Online C Compiler - Programiz

Upload to GitHub

Upload files - deepapreya-h/CNS

+

programiz.com/c-programming/online-compiler/

☆

Verify it's you

Learn DSA the way it should be – with step-by-step code visualization. [Try now!](#)

Programiz C Online Compiler

Programiz PRO

main.c

🔍

⚙️

🔗 Share

🏃 Run

```
23- for (int i = 0; i < len; i += BLOCK_SIZE) {
24-     for (int j = 0; j < BLOCK_SIZE; j++)
25-         temp[j] = data[i + j] ^ iv[j];
26-     xor_cipher(temp, key, output + i);
27-     memcpy(iv, output + i, BLOCK_SIZE);
28- }
29- }
30- void cfb_encrypt(uint8_t *data, int len, uint8_t *key, uint8_t *iv, uint8_t
    *output) {
31-     uint8_t cipher_out[BLOCK_SIZE];
32-     for (int i = 0; i < len; i += BLOCK_SIZE) {
33-         xor_cipher(iv, key, cipher_out);
34-         for (int j = 0; j < BLOCK_SIZE; j++)
35-             output[i + j] = data[i + j] ^ cipher_out[j];
36-         memcpy(iv, output + i, BLOCK_SIZE);
37-     }
38- }
39- void print_hex(const char *label, uint8_t *data, int len) {
40-     printf("%s: ", label);
41-     for (int i = 0; i < len; i++)
42-         printf("%02X ", data[i]);
43-     printf("\n");
44- }
45- int main() {
```

Output

Clear

Plaintext: 48 45 4C 4C 4F 20 42 4C 4F 43 4B 20 57 4F 52 4C 44 21 80 00 00 00 00 00

ECB Cipher: 52 6E 70 01 11 4F 32 CD 55 68 77 6D 09 20 22 CD 5E 0A BC 4D 5E 6F 70 81

CBC Cipher: 52 6E 70 01 11 4F 32 CD 07 06 07 6C 18 6F 10 00 59 0C BB 21 46 00 60 81

CFB Cipher: 52 6E 70 01 11 4F 32 CD 07 06 07 6C 18 6F 10 00 59 0C BB 21 46 00 60 81

=== Code Execution Successful ===

USD/INR

+0.48%

Search

🐱

📁

🌐

📧

📅

📶

ENG

IN

21:04

29.07.2025

Online C Compiler - Programiz

Upload to GitHub

Upload files - deepapreya-h/CNS

programiz.com/c-programming/online-compiler/

Verify it's you

Learn DSA the way it should be – with step-by-step code visualization. Try now!

Programiz C Online Compiler

Programiz PRO

main.c

Share

Run

```
46  uint8_t key[BLOCK_SIZE] = {0x1A, 0x2B, 0x3C, 0x4D, 0x5E, 0x6F, 0x70,
47      0x81};
48  uint8_t iv1[BLOCK_SIZE] = {0x00};
49  uint8_t iv2[BLOCK_SIZE] = {0x00};
50  uint8_t iv3[BLOCK_SIZE] = {0x00};
51  char message[] = "HELLO BLOCK WORLD!";
52  uint8_t data[64];
53  memset(data, 0, sizeof(data));
54  memcpy(data, message, strlen(message));
55  int padded_len = add_padding(data, strlen(message));
56  uint8_t ecb_out[64], cbc_out[64], cfb_out[64];
57  memset(ecb_out, 0, 64);
58  memset(cbc_out, 0, 64);
59  memset(cfb_out, 0, 64);
60  ecb_encrypt(data, padded_len, key, ecb_out);
61  cbc_encrypt(data, padded_len, key, iv1, cbc_out);
62  cfb_encrypt(data, padded_len, key, iv2, cfb_out);
63  print_hex("Plaintext", data, padded_len);
64  print_hex("ECB Cipher", ecb_out, padded_len);
65  print_hex("CBC Cipher", cbc_out, padded_len);
66  print_hex("CFB Cipher", cfb_out, padded_len);
67  return 0;
```

Output

Clear

```
Plaintext: 48 45 4C 4C 4F 20 42 4C 4F 43 4B 20 57 4F 52 4C 44 21 80 00 00 00 00 00
ECB Cipher: 52 6E 70 01 11 4F 32 CD 55 68 77 6D 09 20 22 CD 5E 0A BC 4D 5E 6F 70 81
CBC Cipher: 52 6E 70 01 11 4F 32 CD 07 06 07 6C 18 6F 10 00 59 0C BB 21 46 00 60 81
CFB Cipher: 52 6E 70 01 11 4F 32 CD 07 06 07 6C 18 6F 10 00 59 0C BB 21 46 00 60 81

--- Code Execution Successful ---
```

USD/INR

+0.48%

Search

21:04

29-07-2025