Programiz C Online Compiler

main.c

Share Run

```c
#include <stdio.h>
#include <string.h>
#define BLOCK_SIZE 8
void ecb_encrypt(const char *plaintext, char *ciphertext, const char *key)
{
    for (int i = 0; i < strlen(plaintext); i += BLOCK_SIZE)
    {
        for (int j = 0; j < BLOCK_SIZE; j++)
        {
            ciphertext[i + j] = plaintext[i + j] ^ key[j];
        }
    }
}
void pad_plaintext(char *plaintext)
{
    int len = strlen(plaintext);
    plaintext[len] = 1;
    for (int i = len + 1; i < len + BLOCK_SIZE; i++)
    {
        plaintext[i] = 0;
    }
    plaintext[len + BLOCK_SIZE] = '\0';
}
int main() {
    char plaintext[64] = "saveetha";
    char key[BLOCK_SIZE] = "1234567";
    char ciphertext[64] = {0};

    pad_plaintext(plaintext);
    ecb_encrypt(plaintext, ciphertext, key);

    printf("Ciphertext (ECB): ");
    for (int i = 0; i < strlen(plaintext); i++)
    {
        printf("%02X ", (unsigned char)ciphertext[i]);
    }
    printf("\n");
```

Output

```
Ciphertext (ECB): 42 53 45 51 50 42 5F 61 30


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

Csa5113-CNSIP/M.Koushik Red | Online C Compiler - Programiz | New Tab

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

Programiz C Online Compiler

Programiz PRO >

main.c

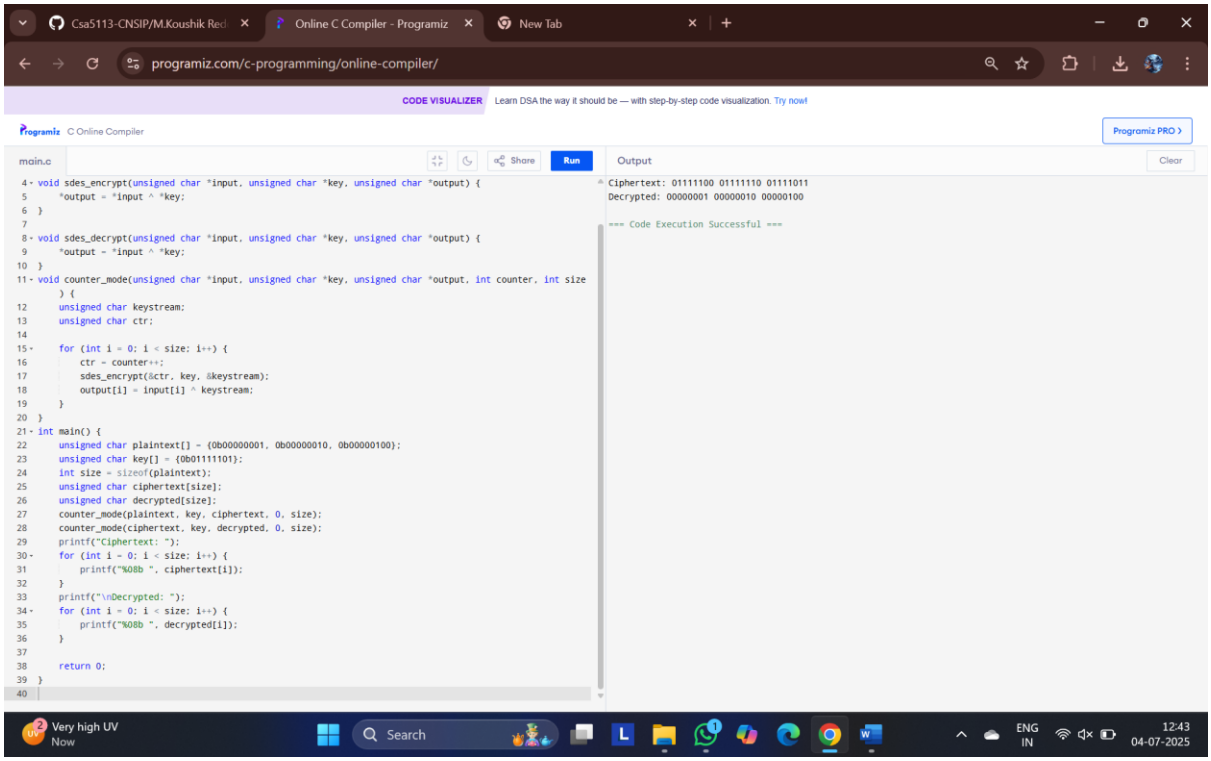
Share Run

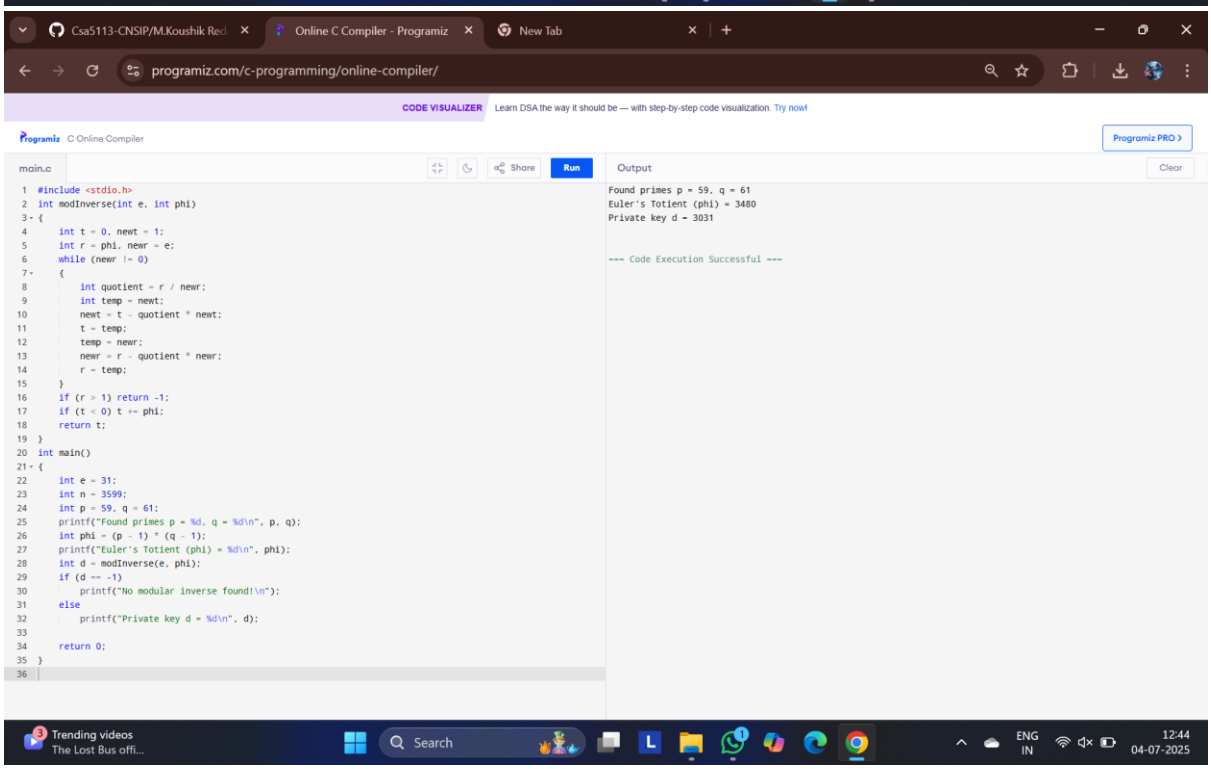
```c
    strcpy(ciphertext, "11110100");
}
void sdes_decrypt(const char *key, const char *ciphertext, char *plaintext)
{
    strcpy(plaintext, "00000001");
}
void cbc_encrypt(const char *iv, const char *plaintext, const char *key, char *ciphertext)
{
    char block[BLOCK_SIZE + 1];
    for (int i = 0; i < strlen(plaintext); i += BLOCK_SIZE) {
        strncpy(block, plaintext + i, BLOCK_SIZE);
        block[BLOCK_SIZE] = '\0';
        sdes_encrypt(key, block, ciphertext + i);
    }
}
void cbc_decrypt(const char *iv, const char *ciphertext, const char *key, char *plaintext)
{
    char block[BLOCK_SIZE + 1];
    for (int i = 0; i < strlen(ciphertext); i += BLOCK_SIZE) {
        strncpy(block, ciphertext + i, BLOCK_SIZE);
        block[BLOCK_SIZE] = '\0';
        sdes_decrypt(key, block, plaintext + i);
    }
}
int main()
{
    const char *key = "011111101";
    const char *iv = "10101010";
    const char *plaintext = "0000000100100011";
    char ciphertext[BLOCK_SIZE * 2 + 1];
    char decrypted[BLOCK_SIZE * 2 + 1];
    cbc_encrypt(iv, plaintext, key, ciphertext);
    printf("Ciphertext: %s\n", ciphertext);
    cbc_decrypt(iv, ciphertext, key, decrypted);
    printf("Decrypted: %s\n", decrypted);
    return 0;
}
```

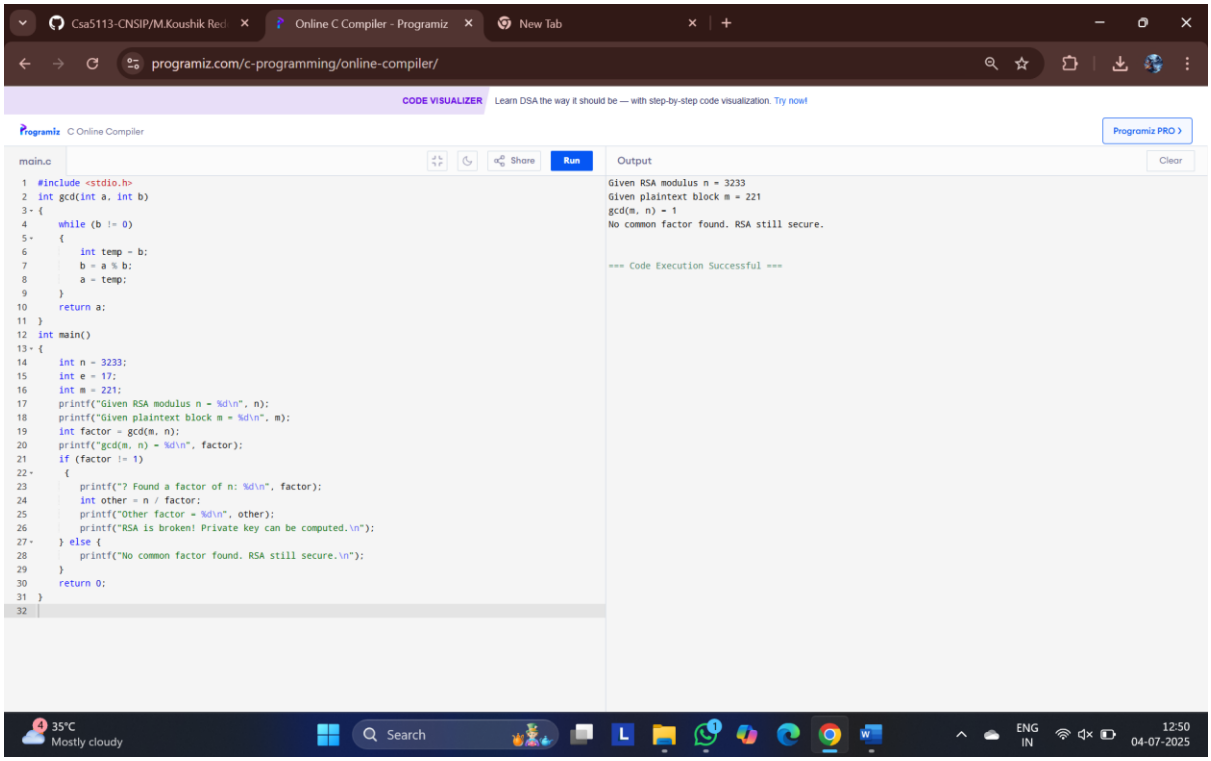
Output Clear

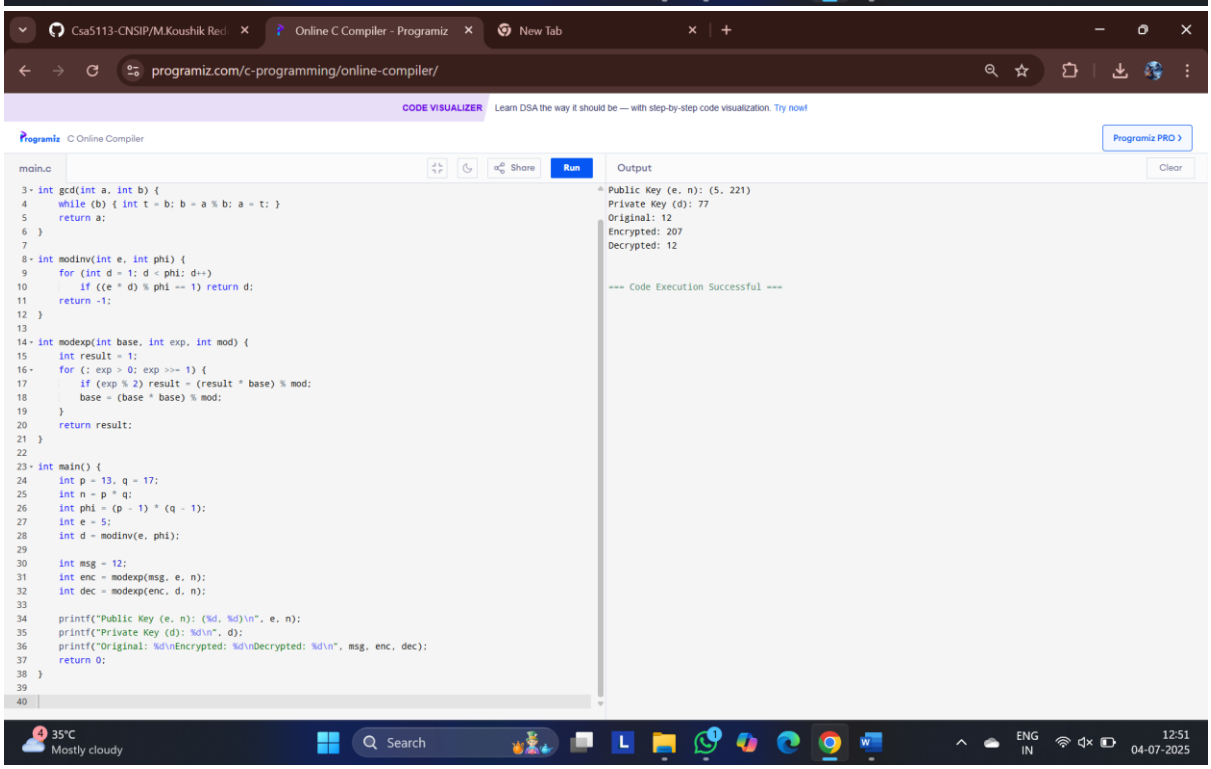
```
Ciphertext: 1111010011110100
Decrypted: 0000000100000001


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

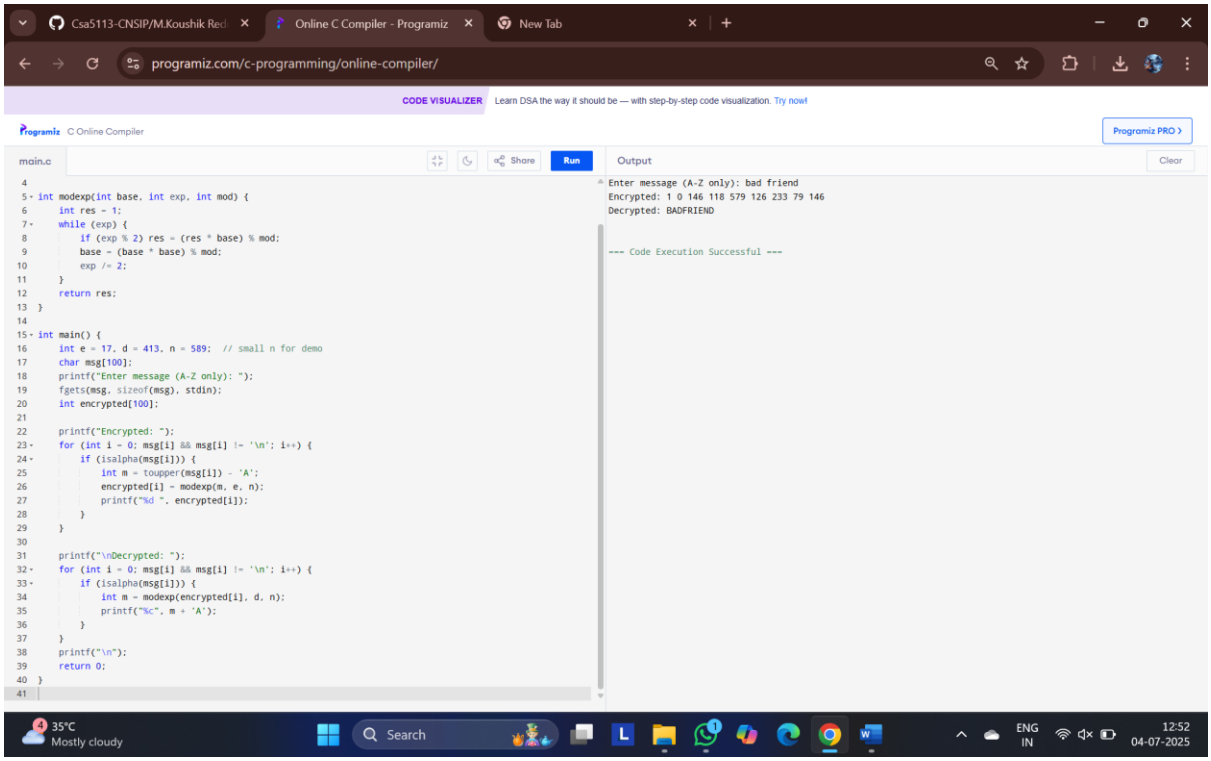
Sports headline Max Verstappen... | Search | ENG IN | 12:42 04-07-2025

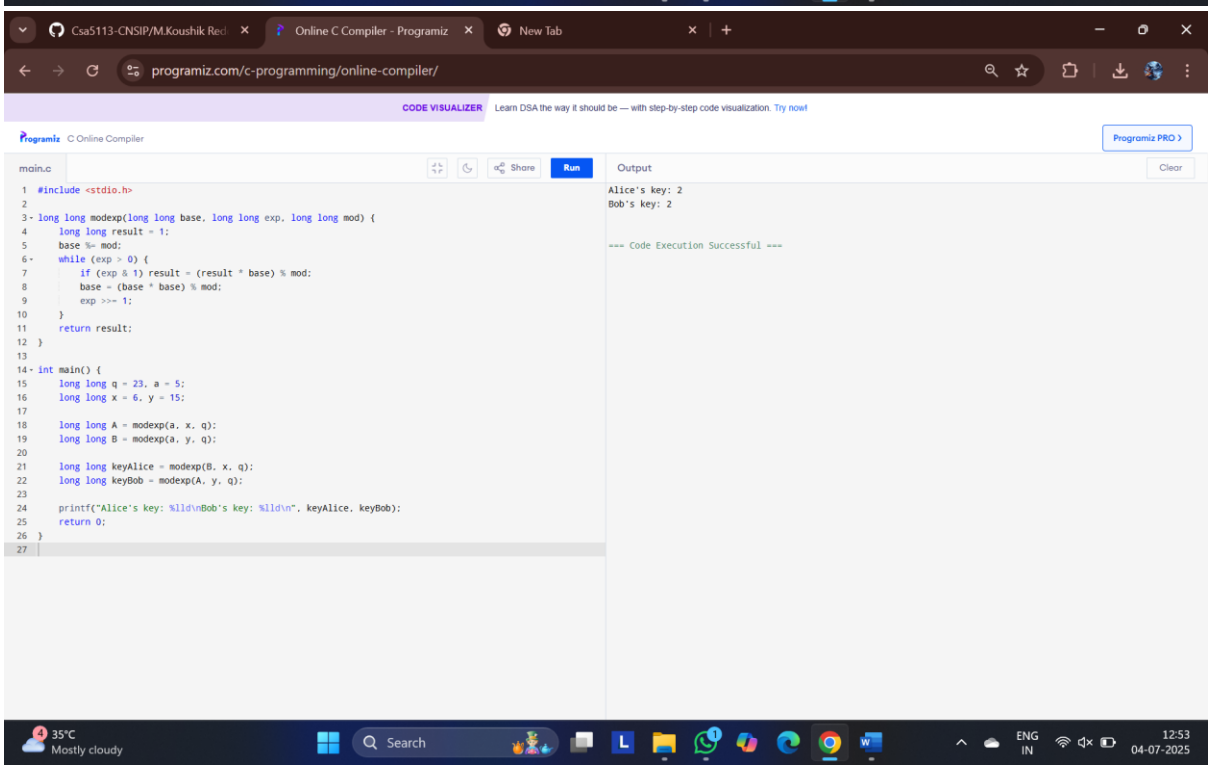

Csa5113-CNSIP/M.Koushik Red
Online C Compiler - Programiz
New Tab
programiz.com/c-programming/online-compiler/
CODE VISUALIZER
Learn DSA the way it should be — with step-by-step code visualization. Try now!
Programiz C Online Compiler
Programiz PRO
main.c
Share
Run
Output
Clear
void sdes_encrypt(unsigned char *input, unsigned char *key, unsigned char *output) {
*output = *input ^ *key;
}
void sdes_decrypt(unsigned char *input, unsigned char *key, unsigned char *output) {
*output = *input ^ *key;
}
void counter_mode(unsigned char *input, unsigned char *key, unsigned char *output, int counter, int size
) {
unsigned char keystream;
unsigned char ctr;
for (int i = 0; i < size; i++) {
ctr = counter++;
sdes_encrypt(&ctr, key, &keystream);
output[i] = input[i] ^ keystream;
}
}
int main() {
unsigned char plaintext[] = {0b00000001, 0b00000010, 0b00000100};
unsigned char key[] = {0b01111101};
int size = sizeof(plaintext);
unsigned char ciphertext[size];
unsigned char decrypted[size];
counter_mode(plaintext, key, ciphertext, 0, size);
counter_mode(ciphertext, key, decrypted, 0, size);
printf("Ciphertext: ");
for (int i = 0; i < size; i++) {
printf("%08b ", ciphertext[i]);
}
printf("\nDecrypted: ");
for (int i = 0; i < size; i++) {
printf("%08b ", decrypted[i]);
}
return 0;
}
Ciphertext: 01111100 01111110 01111011
Decrypted: 00000001 00000010 00000100
=== Code Execution Successful ===
Very high UV
Now
Search
ENG
IN
12:43
04-07-2025

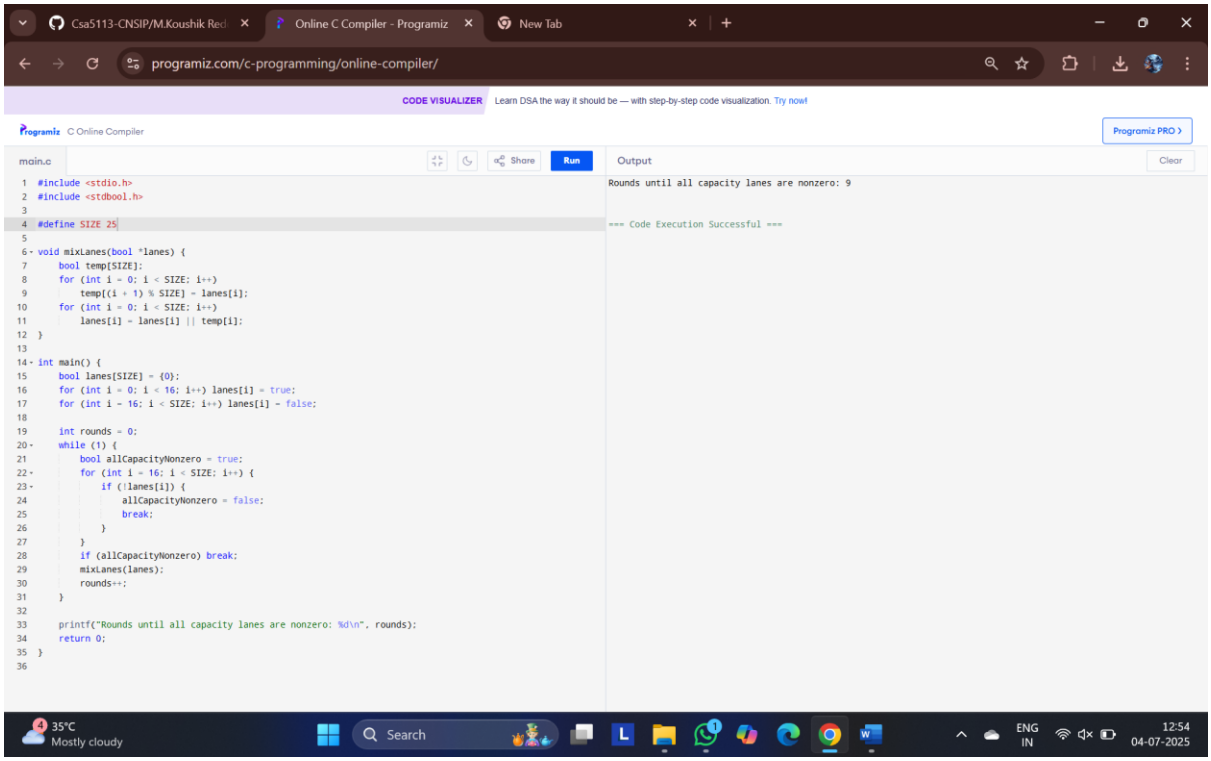

Csa5113-CNSIP/M.Koushik Red
Online C Compiler - Programiz
New Tab
programiz.com/c-programming/online-compiler/
CODE VISUALIZER
Learn DSA the way it should be — with step-by-step code visualization. Try now!
Programiz C Online Compiler
Programiz PRO
main.c
Share
Run
Output
Clear
#include <stdio.h>
int modInverse(int e, int phi)
{
int t = 0, newt = 1;
int r = phi, newr = e;
while (newr != 0)
{
int quotient = r / newr;
int temp = newt;
newt = t - quotient * newt;
t = temp;
temp = newr;
newr = r - quotient * newr;
r = temp;
}
if (r > 1) return -1;
if (t < 0) t += phi;
return t;
}
int main()
{
int e = 31;
int n = 3599;
int p = 59, q = 61;
printf("Found primes p = %d, q = %d\n", p, q);
int phi = (p - 1) * (q - 1);
printf("Euler's Totient (phi) = %d\n", phi);
int d = modInverse(e, phi);
if (d == -1)
printf("No modular inverse found!\n");
else
printf("Private key d = %d\n", d);
return 0;
}
Found primes p = 59, q = 61
Euler's Totient (phi) = 3480
Private key d = 3031
=== Code Execution Successful ===
Trending videos
The Lost Bus offi...
Search
ENG
IN
12:44
04-07-2025

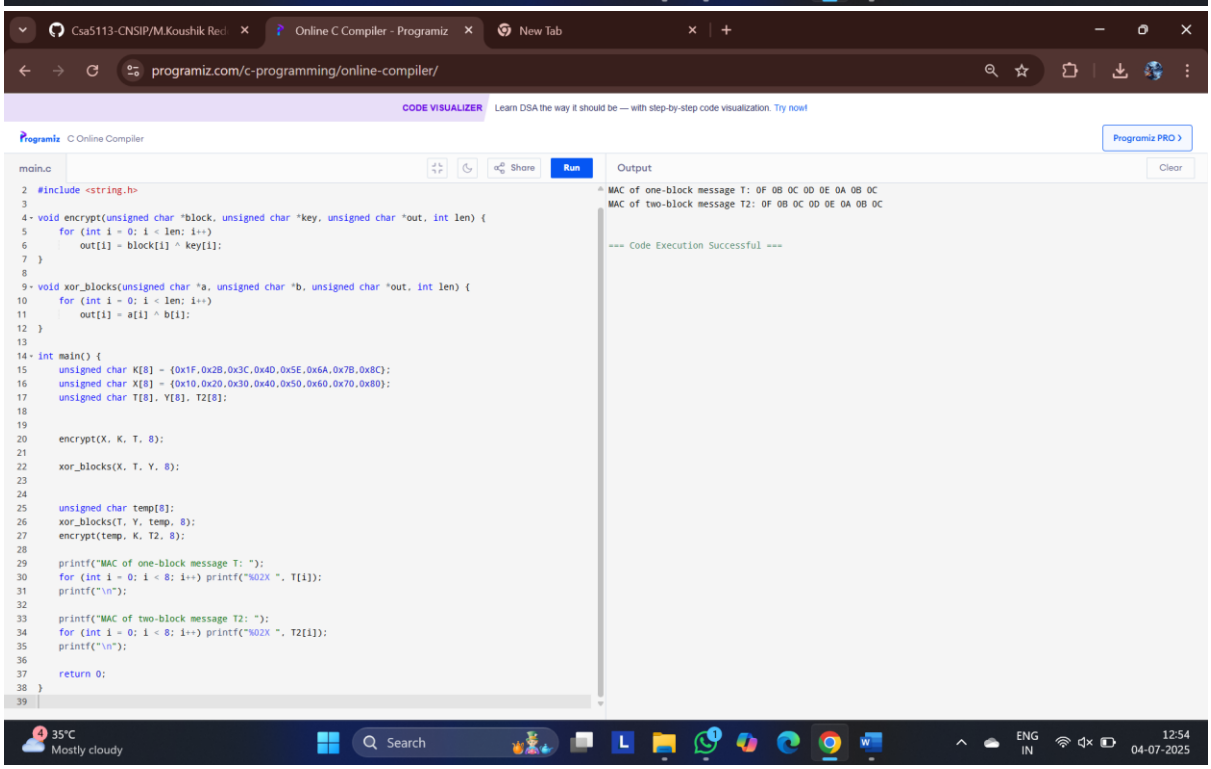

Csa5113-CNSIP/M.Koushik Red
Online C Compiler - Programiz
New Tab
programiz.com/c-programming/online-compiler/
CODE VISUALIZER
Learn DSA the way it should be — with step-by-step code visualization. Try now!
Programiz C Online Compiler
Programiz PRO
main.c
Share
Run
Output
Clear
#include <stdio.h>
int gcd(int a, int b)
{
while (b != 0)
{
int temp = b;
b = a % b;
a = temp;
}
return a;
}
int main()
{
int n = 3233;
int e = 17;
int m = 221;
printf("Given RSA modulus n = %d\n", n);
printf("Given plaintext block m = %d\n", m);
int factor = gcd(m, n);
printf("gcd(m, n) = %d\n", factor);
if (factor != 1)
{
printf("? Found a factor of n: %d\n", factor);
int other = n / factor;
printf("Other factor = %d\n", other);
printf("RSA is broken! Private key can be computed.\n");
} else {
printf("No common factor found. RSA still secure.\n");
}
return 0;
}
Given RSA modulus n = 3233
Given plaintext block m = 221
gcd(m, n) = 1
No common factor found. RSA still secure.
=== Code Execution Successful ===
35°C
Mostly cloudy
Search
ENG
IN
12:50
04-07-2025


Csa5113-CNSIP/M.Koushik Red
Online C Compiler - Programiz
New Tab
programiz.com/c-programming/online-compiler/
CODE VISUALIZER
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
Output
Clear
int gcd(int a, int b) {
while (b) { int t = b; b = a % b; a = t; }
return a;
}
int modinv(int e, int phi) {
for (int d = 1; d < phi; d++)
if ((e * d) % phi == 1) return d;
return -1;
}
int modexp(int base, int exp, int mod) {
int result = 1;
for (; exp > 0; exp >>= 1) {
if (exp % 2) result = (result * base) % mod;
base = (base * base) % mod;
}
return result;
}
int main() {
int p = 13, q = 17;
int n = p * q;
int phi = (p - 1) * (q - 1);
int e = 5;
int d = modinv(e, phi);
int msg = 12;
int enc = modexp(msg, e, n);
int dec = modexp(enc, d, n);
printf("Public Key (e, n): (%d, %d)\n", e, n);
printf("Private Key (d): %d\n", d);
printf("Original: %d\nEncrypted: %d\nDecrypted: %d\n", msg, enc, dec);
return 0;
}
Public Key (e, n): (5, 221)
Private Key (d): 77
Original: 12
Encrypted: 207
Decrypted: 12
=== Code Execution Successful ===
35°C
Mostly cloudy
Search
ENG
IN
12:51
04-07-2025


Csa5113-CNSIP/M.Koushik Red
Online C Compiler - Programiz
New Tab
programiz.com/c-programming/online-compiler/
CODE VISUALIZER
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
Output
Clear
int modexp(int base, int exp, int mod) {
int res = 1;
while (exp) {
if (exp % 2) res = (res * base) % mod;
base = (base * base) % mod;
exp /= 2;
}
return res;
}
int main() {
int e = 17, d = 413, n = 589; // small n for demo
char msg[100];
printf("Enter message (A-Z only): ");
fgets(msg, sizeof(msg), stdin);
int encrypted[100];
printf("Encrypted: ");
for (int i = 0; msg[i] && msg[i] != '\n'; i++) {
if (isalpha(msg[i])) {
int m = toupper(msg[i]) - 'A';
encrypted[i] = modexp(m, e, n);
printf("%d ", encrypted[i]);
}
}
printf("\nDecrypted: ");
for (int i = 0; msg[i] && msg[i] != '\n'; i++) {
if (isalpha(msg[i])) {
int m = modexp(encrypted[i], d, n);
printf("%c", m + 'A');
}
}
printf("\n");
return 0;
}
Enter message (A-Z only): bad friend
Encrypted: 1 0 146 118 579 126 233 79 146
Decrypted: BADFRIEND
--- Code Execution Successful ---
35°C
Mostly cloudy
Search
ENG
IN
12:52
04-07-2025


Csa5113-CNSIP/M.Koushik Red
Online C Compiler - Programiz
New Tab
programiz.com/c-programming/online-compiler/
CODE VISUALIZER
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
Output
Clear
#include <stdio.h>
long long modexp(long long base, long long exp, long long mod) {
long long result = 1;
base %= mod;
while (exp > 0) {
if (exp & 1) result = (result * base) % mod;
base = (base * base) % mod;
exp >>= 1;
}
return result;
}
int main() {
long long q = 23, a = 5;
long long x = 6, y = 15;
long long A = modexp(a, x, q);
long long B = modexp(a, y, q);
long long keyAlice = modexp(B, x, q);
long long keyBob = modexp(A, y, q);
printf("Alice's key: %lld\nBob's key: %lld\n", keyAlice, keyBob);
return 0;
}
Alice's key: 2
Bob's key: 2
--- Code Execution Successful ---
35°C
Mostly cloudy
Search
ENG
IN
12:53
04-07-2025


Csa5113-CNSIP/M.Koushik Red
Online C Compiler - Programiz
New Tab
programiz.com/c-programming/online-compiler/
CODE VISUALIZER
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
Output
Clear
#include <stdio.h>
#include <stdbool.h>
#define SIZE 25
void mixLanes(bool *lanes) {
bool temp[SIZE];
for (int i = 0; i < SIZE; i++)
temp[(i + 1) % SIZE] = lanes[i];
for (int i = 0; i < SIZE; i++)
lanes[i] = lanes[i] || temp[i];
}
int main() {
bool lanes[SIZE] = {0};
for (int i = 0; i < 16; i++) lanes[i] = true;
for (int i = 16; i < SIZE; i++) lanes[i] = false;
int rounds = 0;
while (1) {
bool allCapacityNonzero = true;
for (int i = 16; i < SIZE; i++) {
if (!lanes[i]) {
allCapacityNonzero = false;
break;
}
}
if (allCapacityNonzero) break;
mixLanes(lanes);
rounds++;
}
printf("Rounds until all capacity lanes are nonzero: %d\n", rounds);
return 0;
}
Rounds until all capacity lanes are nonzero: 9
=== Code Execution Successful ===
35°C
Mostly cloudy
Search
ENG
IN
12:54
04-07-2025


Csa5113-CNSIP/M.Koushik Red
Online C Compiler - Programiz
New Tab
programiz.com/c-programming/online-compiler/
CODE VISUALIZER
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
Output
Clear
#include <string.h>
void encrypt(unsigned char *block, unsigned char *key, unsigned char *out, int len) {
for (int i = 0; i < len; i++)
out[i] = block[i] ^ key[i];
}
void xor_blocks(unsigned char *a, unsigned char *b, unsigned char *out, int len) {
for (int i = 0; i < len; i++)
out[i] = a[i] ^ b[i];
}
int main() {
unsigned char K[8] = {0x1F,0x2B,0x3C,0x4D,0x5E,0x6A,0x7B,0x8C};
unsigned char X[8] = {0x10,0x20,0x30,0x40,0x50,0x60,0x70,0x80};
unsigned char T[8], Y[8], T2[8];
encrypt(X, K, T, 8);
xor_blocks(X, T, Y, 8);
unsigned char temp[8];
xor_blocks(T, Y, temp, 8);
encrypt(temp, K, T2, 8);
printf("MAC of one-block message T: ");
for (int i = 0; i < 8; i++) printf("%02X ", T[i]);
printf("\n");
printf("MAC of two-block message T2: ");
for (int i = 0; i < 8; i++) printf("%02X ", T2[i]);
printf("\n");
return 0;
}
MAC of one-block message T: 0F 0B 0C 0D 0E 0A 0B 0C
MAC of two-block message T2: 0F 0B 0C 0D 0E 0A 0B 0C
=== Code Execution Successful ===
35°C
Mostly cloudy
Search
ENG
IN
12:54
04-07-2025