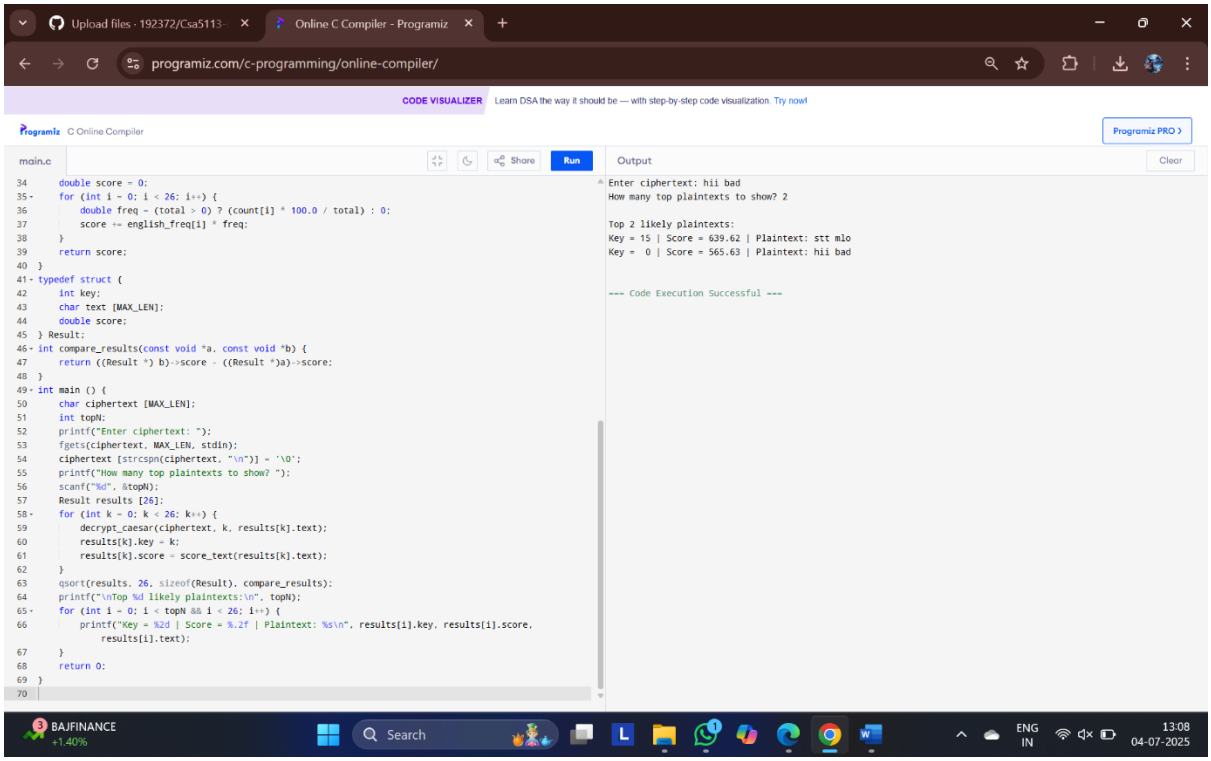
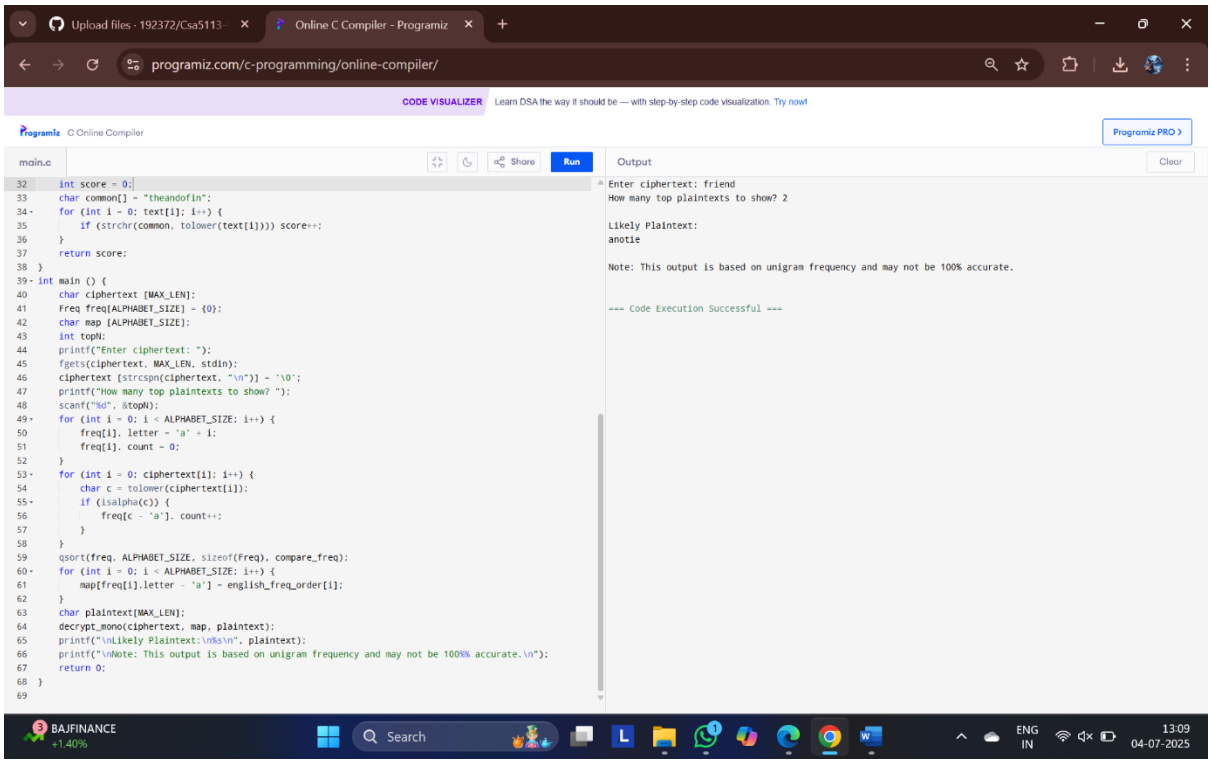
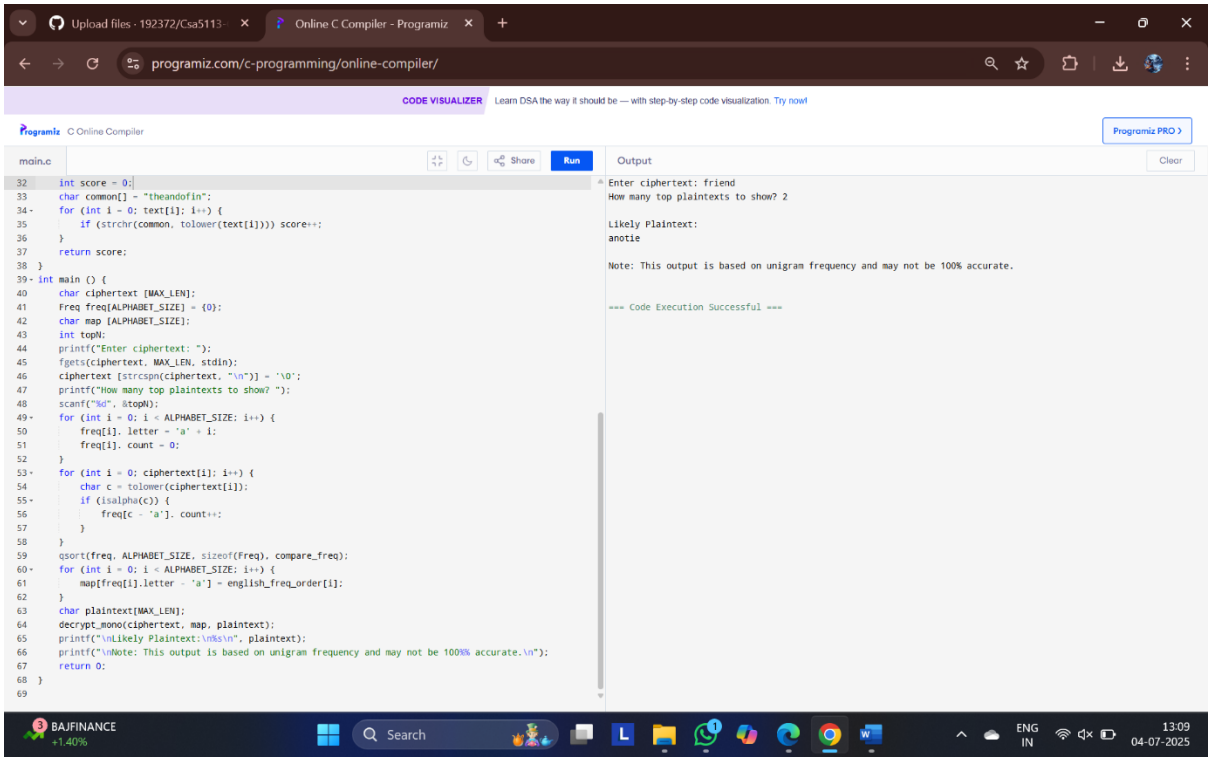
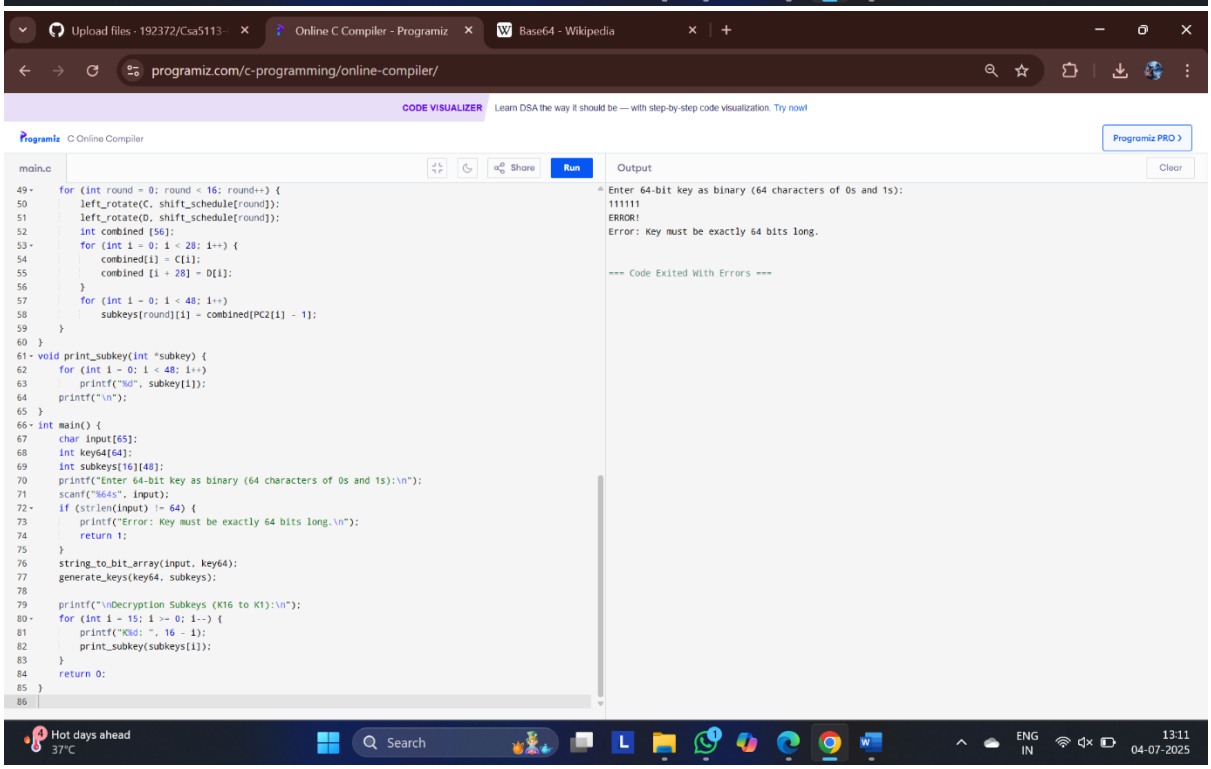
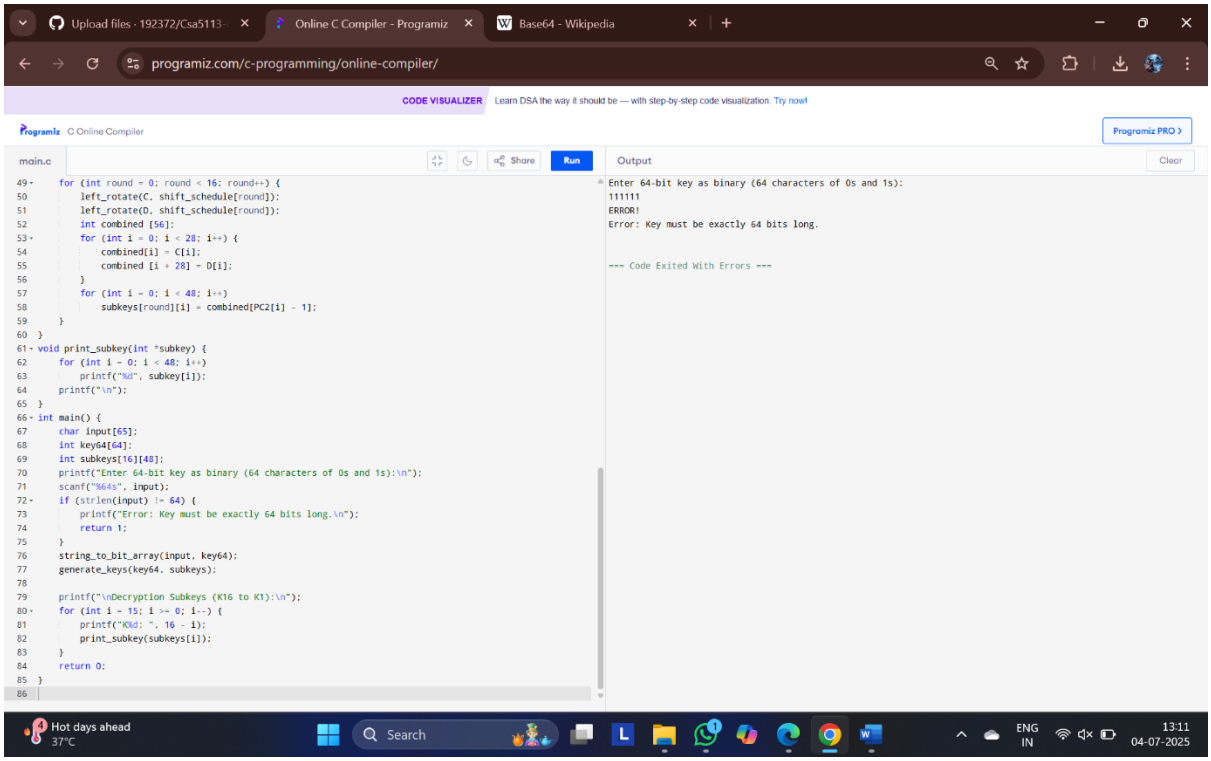
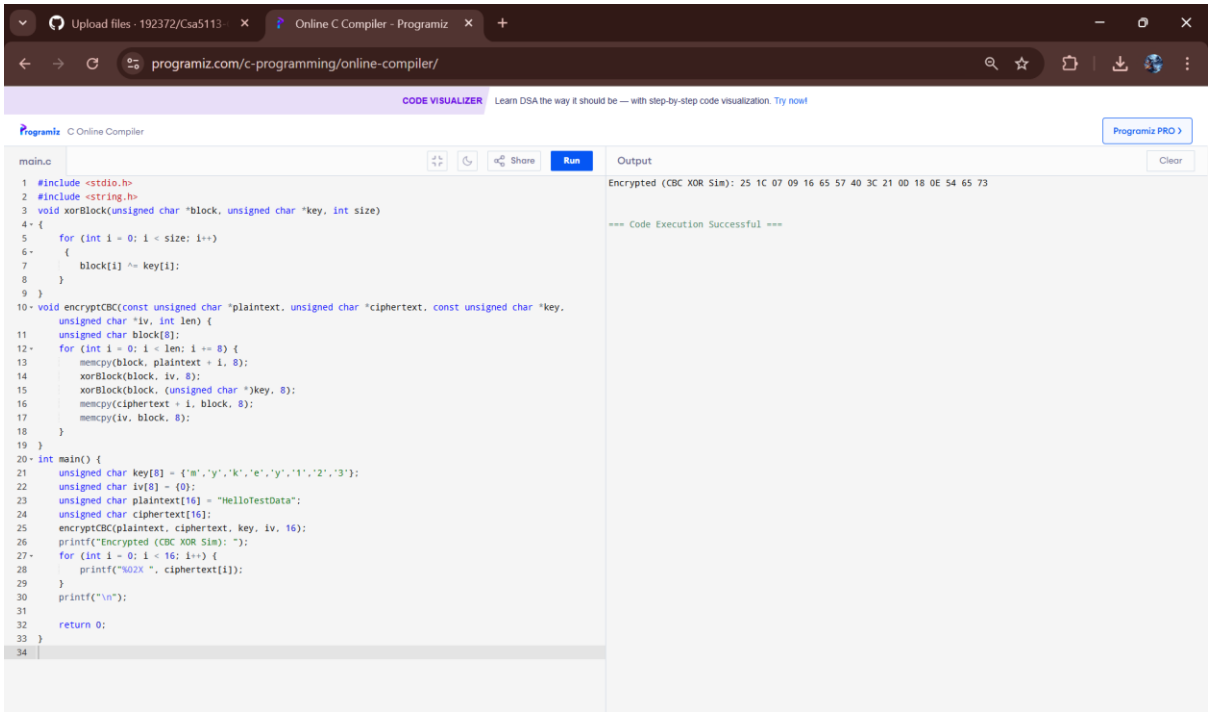

Upload files - 192372/Csa5113
Online C Compiler - Programiz
programiz.com/c-programming/online-compiler/
CODE VISUALIZER
Learn DSA the way it should be — with step-by-step code visualization. Try now!
Programiz C Online Compiler
Programiz PRO
main.c
Share
Run
Output
Clear
double score = 0;
for (int i = 0; i < 26; i++) {
double freq = (total > 0) ? (count[i] * 100.0 / total) : 0;
score += english_freq[i] * freq;
}
return score;
}
typedef struct {
int key;
char text [MAX_LEN];
double score;
} Result;
int compare_results(const void *a, const void *b) {
return ((Result *) b)->score - ((Result *)a)->score;
}
int main () {
char ciphertext [MAX_LEN];
int topN;
printf("Enter ciphertext: ");
fgets(ciphertext, MAX_LEN, stdin);
ciphertext [strcspn(ciphertext, "\n")] = '\0';
printf("How many top plaintexts to show? ");
scanf("%d", &topN);
Result results [26];
for (int k = 0; k < 26; k++) {
decrypt_caesar(ciphertext, k, results[k].text);
results[k].key = k;
results[k].score = score_text(results[k].text);
}
qsort(results, 26, sizeof(Result), compare_results);
printf("\nTop %d likely plaintexts:\n", topN);
for (int i = 0; i < topN && i < 26; i++) {
printf("Key = %2d | Score = %.2f | Plaintext: %s\n", results[i].key, results[i].score,
results[i].text);
}
return 0;
}
Enter ciphertext: hii bad
How many top plaintexts to show? 2
Top 2 likely plaintexts:
Key = 15 | Score = 639.62 | Plaintext: stt mlo
Key = 0 | Score = 565.63 | Plaintext: hii bad
--- Code Execution Successful ---
BAJFINANCE
+1.40%
Search
ENG
IN
13:08
04-07-2025

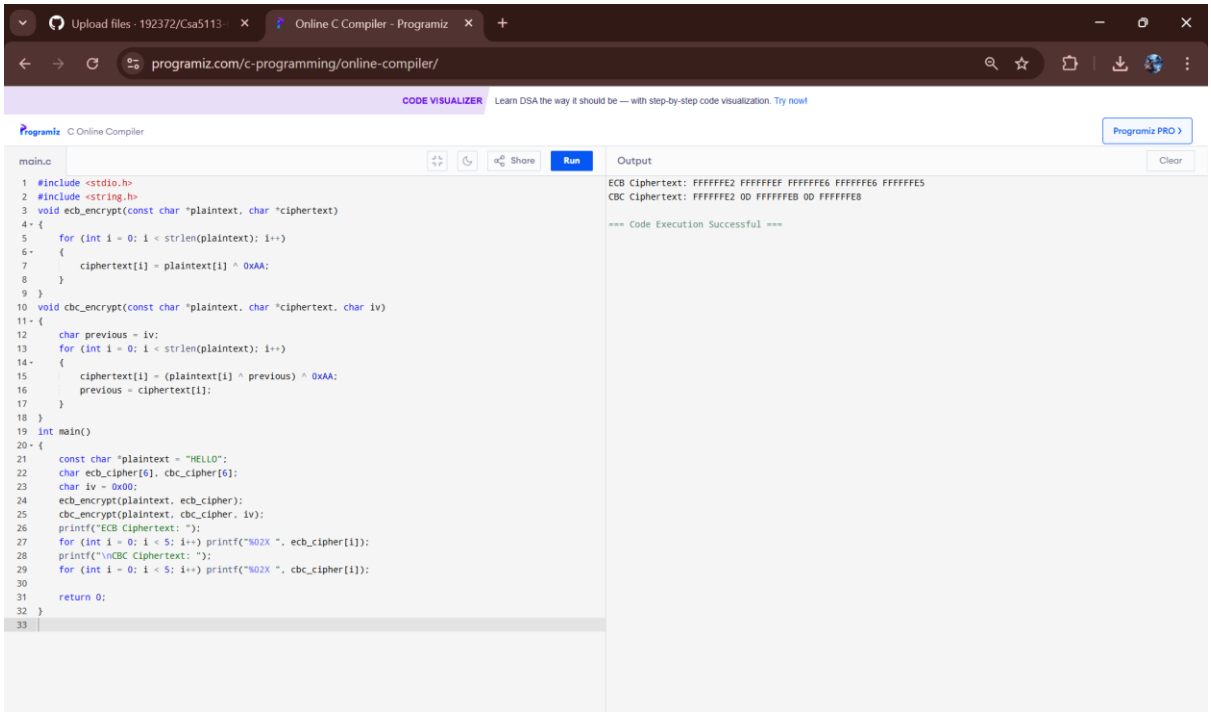

Upload files - 192372/Csa5113
Online C Compiler - Programiz
programiz.com/c-programming/online-compiler/
CODE VISUALIZER
Learn DSA the way it should be — with step-by-step code visualization. Try now!
Programiz C Online Compiler
Programiz PRO
main.c
Share
Run
Output
Clear
int score = 0;
char common[] = "theandofin";
for (int i = 0; text[i]; i++) {
if (strchr(common, tolower(text[i]))) score++;
}
return score;
}
int main () {
char ciphertext [MAX_LEN];
Freq freq[ALPHABET_SIZE] = {0};
char map [ALPHABET_SIZE];
int topN;
printf("Enter ciphertext: ");
fgets(ciphertext, MAX_LEN, stdin);
ciphertext [strcspn(ciphertext, "\n")] = '\0';
printf("How many top plaintexts to show? ");
scanf("%d", &topN);
for (int i = 0; i < ALPHABET_SIZE; i++) {
freq[i].letter = 'a' + i;
freq[i].count = 0;
}
for (int i = 0; ciphertext[i]; i++) {
char c = tolower(ciphertext[i]);
if (isalpha(c)) {
freq[c - 'a'].count++;
}
}
qsort(freq, ALPHABET_SIZE, sizeof(Freq), compare_freq);
for (int i = 0; i < ALPHABET_SIZE; i++) {
map[freq[i].letter - 'a'] = english_freq_order[i];
}
char plaintext[MAX_LEN];
decrypt_mono(ciphertext, map, plaintext);
printf("\nLikely Plaintext:\n%s\n", plaintext);
printf("\nNote: This output is based on unigram frequency and may not be 100%% accurate.\n");
return 0;
}
Enter ciphertext: friend
How many top plaintexts to show? 2
Likely Plaintext:
anotie
Note: This output is based on unigram frequency and may not be 100% accurate.
--- Code Execution Successful ---
BAJFINANCE
+1.40%
Search
ENG
IN
13:09
04-07-2025


Upload files - 192372/Csa5113
Online C Compiler - Programiz
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 score = 0;
char common[] = "theandofin";
for (int i = 0; text[i]; i++) {
if (strchr(common, tolower(text[i]))) score++;
}
return score;
}
int main () {
char ciphertext [MAX_LEN];
Freq freq[ALPHABET_SIZE] = {0};
char map [ALPHABET_SIZE];
int topN;
printf("Enter ciphertext: ");
fgets(ciphertext, MAX_LEN, stdin);
ciphertext [strcspn(ciphertext, "\n")] = '\0';
printf("How many top plaintexts to show? ");
scanf("%d", &topN);
for (int i = 0; i < ALPHABET_SIZE; i++) {
freq[i].letter = 'a' + i;
freq[i].count = 0;
}
for (int i = 0; ciphertext[i]; i++) {
char c = tolower(ciphertext[i]);
if (isalpha(c)) {
freq[c - 'a'].count++;
}
}
qsort(freq, ALPHABET_SIZE, sizeof(Freq), compare_freq);
for (int i = 0; i < ALPHABET_SIZE; i++) {
map[freq[i].letter - 'a'] = english_freq_order[i];
}
char plaintext[MAX_LEN];
decrypt_mono(ciphertext, map, plaintext);
printf("\nLikely Plaintext:\n%s\n", plaintext);
printf("\nNote: This output is based on unigram frequency and may not be 100%% accurate.\n");
return 0;
}
Enter ciphertext: friend
How many top plaintexts to show? 2
Likely Plaintext:
anotie
Note: This output is based on unigram frequency and may not be 100% accurate.
=== Code Execution Successful ===
BAJFINANCE
+1.40%
Search
ENG
IN
13:09
04-07-2025


Upload files - 192372/Csa5113
Online C Compiler - Programiz
Base64 - Wikipedia
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
for (int round = 0; round < 16; round++) {
left_rotate(C, shift_schedule[round]);
left_rotate(D, shift_schedule[round]);
int combined [56];
for (int i = 0; i < 28; i++) {
combined[i] = C[i];
combined [i + 28] = D[i];
}
for (int i = 0; i < 48; i++)
subkeys[round][i] = combined[PC2[i] - 1];
}
}
void print_subkey(int *subkey) {
for (int i = 0; i < 48; i++)
printf("%d", subkey[i]);
printf("\n");
}
int main() {
char input[65];
int key64[64];
int subkeys[16][48];
printf("Enter 64-bit key as binary (64 characters of 0s and 1s):\n");
scanf("%64s", input);
if (strlen(input) != 64) {
printf("Error: Key must be exactly 64 bits long.\n");
return 1;
}
string_to_bit_array(input, key64);
generate_keys(key64, subkeys);
printf("\nDecryption Subkeys (K16 to K1):\n");
for (int i = 15; i >= 0; i--) {
printf("K%d: ", 16 - i);
print_subkey(subkeys[i]);
}
return 0;
}
Enter 64-bit key as binary (64 characters of 0s and 1s):
111111
ERROR!
Error: Key must be exactly 64 bits long.
=== Code Exited With Errors ===
Hot days ahead
37°C
Search
ENG
IN
13:11
04-07-2025


Upload files - 192372/Csa5113
Online C Compiler - Programiz
Base64 - Wikipedia
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
for (int round = 0; round < 16; round++) {
left_rotate(C, shift_schedule[round]);
left_rotate(D, shift_schedule[round]);
int combined [56];
for (int i = 0; i < 28; i++) {
combined[i] = C[i];
combined [i + 28] = D[i];
}
for (int i = 0; i < 48; i++)
subkeys[round][i] = combined[PC2[i] - 1];
}
}
void print_subkey(int *subkey) {
for (int i = 0; i < 48; i++)
printf("%d", subkey[i]);
printf("\n");
}
int main() {
char input[65];
int key64[64];
int subkeys[16][48];
printf("Enter 64-bit key as binary (64 characters of 0s and 1s):\n");
scanf("%64s", input);
if (strlen(input) != 64) {
printf("Error: Key must be exactly 64 bits long.\n");
return 1;
}
string_to_bit_array(input, key64);
generate_keys(key64, subkeys);
printf("\nDecryption Subkeys (K16 to K1):\n");
for (int i = 15; i >= 0; i--) {
printf("K%d: ", 16 - i);
print_subkey(subkeys[i]);
}
return 0;
}
Enter 64-bit key as binary (64 characters of 0s and 1s):
111111
ERROR!
Error: Key must be exactly 64 bits long.
=== Code Exited With Errors ===
Hot days ahead
37°C
Search
ENG
IN
13:11
04-07-2025


Upload files - 192372/Csa5113
Online C Compiler - Programiz
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 <string.h>
void xorBlock(unsigned char *block, unsigned char *key, int size)
{
for (int i = 0; i < size; i++)
{
block[i] ^= key[i];
}
}
void encryptCBC(const unsigned char *plaintext, unsigned char *ciphertext, const unsigned char *key,
unsigned char *iv, int len) {
unsigned char block[8];
for (int i = 0; i < len; i += 8) {
memcpy(block, plaintext + i, 8);
xorBlock(block, iv, 8);
xorBlock(block, (unsigned char *)key, 8);
memcpy(ciphertext + i, block, 8);
memcpy(iv, block, 8);
}
}
int main() {
unsigned char key[8] = {'m','y','k','e','y','1','2','3'};
unsigned char iv[8] = {0};
unsigned char plaintext[16] = "HelloTestData";
unsigned char ciphertext[16];
encryptCBC(plaintext, ciphertext, key, iv, 16);
printf("Encrypted (CBC XOR Sim): ");
for (int i = 0; i < 16; i++) {
printf("%02X ", ciphertext[i]);
}
printf("\n");
return 0;
}
Encrypted (CBC XOR Sim): 25 1C 07 09 16 65 57 40 3C 21 0D 18 0E 54 65 73
=== Code Execution Successful ===


Upload files - 192372/Csa5113
Online C Compiler - Programiz
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 <string.h>
void ecb_encrypt(const char *plaintext, char *ciphertext)
{
for (int i = 0; i < strlen(plaintext); i++)
{
ciphertext[i] = plaintext[i] ^ 0xAA;
}
}
void cbc_encrypt(const char *plaintext, char *ciphertext, char iv)
{
char previous = iv;
for (int i = 0; i < strlen(plaintext); i++)
{
ciphertext[i] = (plaintext[i] ^ previous) ^ 0xAA;
previous = ciphertext[i];
}
}
int main()
{
const char *plaintext = "HELLO";
char ecb_cipher[6], cbc_cipher[6];
char iv = 0x00;
ecb_encrypt(plaintext, ecb_cipher);
cbc_encrypt(plaintext, cbc_cipher, iv);
printf("ECB Ciphertext: ");
for (int i = 0; i < 5; i++) printf("%02X ", ecb_cipher[i]);
printf("\nCBC Ciphertext: ");
for (int i = 0; i < 5; i++) printf("%02X ", cbc_cipher[i]);
return 0;
}
ECB Ciphertext: FFFFFFE2 FFFFFFEF FFFFFFE6 FFFFFFE6 FFFFFFE5
CBC Ciphertext: FFFFFFE2 0D FFFFFFEB 0D FFFFFFE8
=== Code Execution Successful ===