Date: 22nd July 2024

22BCP116

Experiment 1 Information Security - Lab

AIM: Study and Implement a program for Caesar Cipher

Introduction:

In cryptography, Caesar cipher is one of the simplest and most widely known encryption techniques. The method is named after Julius Caesar, who used it in his private correspondence. In this technique, each letter in the plaintext is shifted a fixed number of places down or up the alphabet.

* Program - Normal Approach:

#include <stdio.h>

#include <string.h>

void encrypt(char text[], int shift) {

int i;

char ch;

for(i = 0; text[i] != '\0'; ++i) {

ch = text[i];

// Encrypt uppercase letters

if(ch >= 'A' && ch <= 'Z') {

ch = ch + shift;

if(ch > 'Z') {

ch = ch - 'Z' + 'A' - 1;

}

text[i] = ch;

}

// Encrypt lowercase letters

else if(ch >= 'a' && ch <= 'z') {

ch = ch + shift;

if(ch > 'z') {

ch = ch - 'z' + 'a' - 1;

}

text[i] = ch;

}

else if(ch >= '0' && ch <= '9') {

ch = (ch - '0' + shift) % 10 + '0';

text[i] = ch;

}

// Leave other characters (including whitespace) unchanged

}

}

void decrypt(char text[], int shift) {

int i;

char ch;

for(i = 0; text[i] != '\0'; ++i) {

ch = text[i];

// Decrypt uppercase letters

if(ch >= 'A' && ch <= 'Z') {

ch = ch - shift;

if(ch < 'A') {

ch = ch + 'Z' - 'A' + 1;

}

text[i] = ch;

}

// Decrypt lowercase letters

else if(ch >= 'a' && ch <= 'z') {

ch = ch - shift;

if(ch < 'a') {

ch = ch + 'z' - 'a' + 1;

}

text[i] = ch;

}

// Leave other characters (including whitespace) unchanged

}

}

int main() {

char text[100];

int shift;

int choice;

printf("Enter a message: ");

fgets(text, sizeof(text), stdin);

printf("Enter shift: ");

scanf("%d", &shift);

// Remove newline character from fgets input

int len = strlen(text);

if (len > 0 && text[len-1] == '\n') {

text[len-1] = '\0';

}

// Ask user for choice

printf("Choose an option:\n");

printf("1. Encrypt\n");

printf("2. Decrypt\n");

printf("Enter choice (1 or 2): ");

scanf("%d", &choice);

// Process based on user choice

if (choice == 1) {

encrypt(text, shift);

printf("Encrypted message: %s\n", text);

} else if (choice == 2) {

decrypt(text, shift);

printf("Decrypted message: %s\n", text);

} else {

printf("Invalid choice\n");

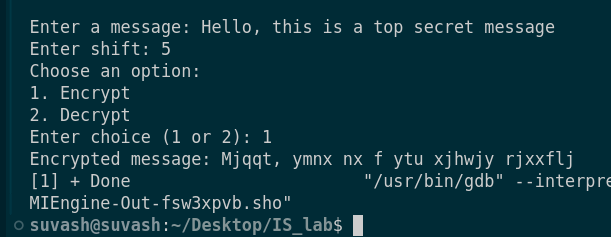
}

return 0;

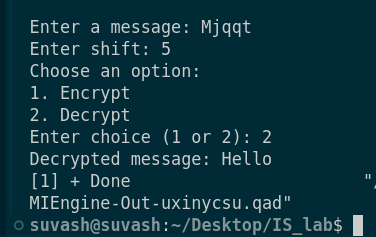
}

Output:

1. Encryption



2. Decryption



* Modified/Revised version

#include <stdio.h>

#include <string.h>

char shift\_char(char ch, int shift) {

// Normalize shift

int range;

char base;

if (ch >= 'A' && ch <= 'Z') {

base = 'A';

range = 26;

} else if (ch >= 'a' && ch <= 'z') {

base = 'a';

range = 26;

} else if (ch >= '0' && ch <= '9') {

base = '0';

range = 10;

} else {

return ch; // Do not encrypt/decrypt non-alphanumeric characters

}

// Normalize the shift to be within 0 to range-1

shift = ((shift % range) + range) % range;

// Perform the shift

return ((ch - base + shift) % range + range) % range + base;

}

// Function to process text with given shift

void process\_text(char text[], int shift) {

for (int i = 0; text[i] != '\0'; ++i) {

text[i] = shift\_char(text[i], shift);

}

}

int main() {

char text[100];

int shift;

int choice;

printf("Enter a message: ");

fgets(text, sizeof(text), stdin);

printf("Enter shift: ");

scanf("%d", &shift);

// Remove newline character from fgets input

int len = strlen(text);

if (len > 0 && text[len-1] == '\n') {

text[len-1] = '\0';

}

// Ask user for choice

printf("Choose an option:\n");

printf("1. Encrypt\n");

printf("2. Decrypt\n");

printf("Enter choice (1 or 2): ");

scanf("%d", &choice);

// Process based on user choice

if (choice == 1) {

process\_text(text, shift);

printf("Encrypted message: %s\n", text);

} else if (choice == 2) {

process\_text(text, -shift);

printf("Decrypted message: %s\n", text);

} else {

printf("Invalid choice\n");

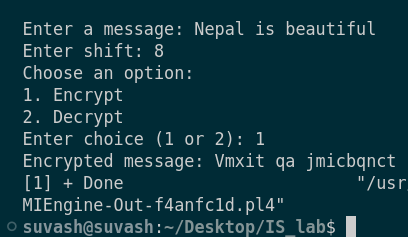
}

return 0;

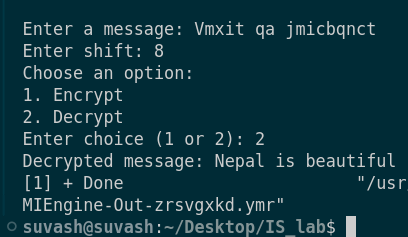
}

**Output:**

Encryption:



Decryption:



Cryptool Output:

1. Encryption

(Attach screenshots)

2. Decryption

(Attach screenshots)