**Theory:**

The Caesar Cipher is one of the oldest methods to convert data into a format such that it cannot be recognized by unauthorized users. Encryption is a process of converting data into secret code and decryption is the exact opposite process i.e. converting code to original data. It shifts each letter a few positions right to encrypt it. Further, this can be shifted from the same positions to the left to get the message.

Julius Caesar developed a method to send secret messages to his generals in 100BC. On the way, even if the message gets into the hands of the enemy, the message couldn't be interpreted by the enemy.

Caeser gave a very simple method to encrypt and decrypt the information. It is also known as shift Caeser as the method shifts the character key positions ahead. If the current character is d and the key = 3, then Cipher text will be storing g i.e. 4 positions ahead of d. To decrypt the same, we go 4 positions behind for g which gives us d back.

Formula:

Encryption:

Cipher Text = (ch - 'a' + key) % 26 + 'a'

Decryption:

Plain Text = (ch - 'a' - key + 26) % 26 + 'a'

T(n) = O(n)

**Programming Language: C**

**IDE: VS-Code**

**Code :**

#include <stdio.h>

#include <ctype.h>

#define MAX\_SIZE 500

void encrypt() {

char text[MAX\_SIZE], ch;

int key, i;

// taking user input

printf("Enter a message to encrypt: ");

scanf("%s", text);

printf("Enter the key: ");

scanf("%d", &key);

// visiting character by character

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

ch = text[i];

// check for valid character

if (isalnum(ch)) {

// lower case characters

if (islower(ch)) {

ch = (ch - 'a' + key) % 26 + 'a';

}

// uppercase characters

else if (isupper(ch)) {

ch = (ch - 'A' + key) % 26 + 'A';

}

// numbers

else if (isdigit(ch)) {

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

}

}

// invalid character

else {

printf("Invalid Message");

return;

}

// adding encoded answer

text[i] = ch;

}

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

}

void decrypt() {

char text[MAX\_SIZE], ch;

int key, i;

// taking user input

printf("\n\n\nEnter a message to decrypt: ");

scanf("%s", text);

printf("Enter the key: ");

scanf("%d", &key);

//visiting each character

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

ch = text[i];

// check for valid characters

if (isalnum(ch)) {

// lower case characters

if (islower(ch)) {

ch = (ch - 'a' - key + 26) % 26 + 'a';

}

// uppercase characters

else if (isupper(ch)) {

ch = (ch - 'A' - key + 26) % 26 + 'A';

}

// numbers

else if (isdigit(ch)) {

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

}

}

// invalid characters

else {

printf("Invalid Message");

return;

}

// adding decoded character back

text[i] = ch;

}

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

}

int main() {

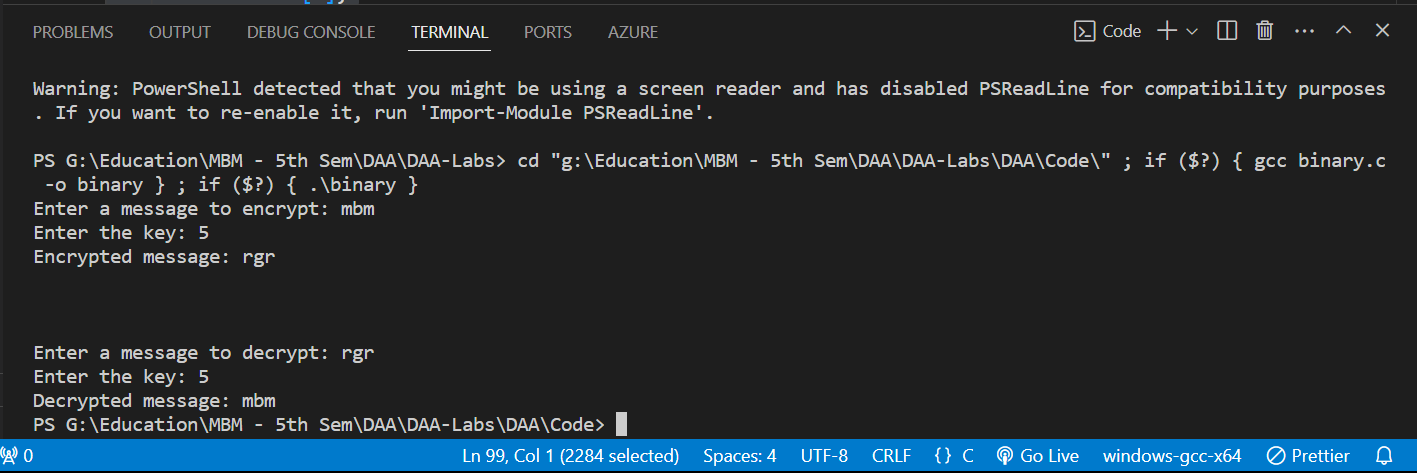
encrypt();

decrypt();

return 0; // Indicates successful execution

}

**Output**

****