

PRACTICAL-4

AIM: Write a C program to implement encryption and decryption of Polyalphabetic Cipher(Vigenere).

INTRODUCTION:

- The best known and one of the simplest such algorithms is referred to as the Vigenere cipher, where the set of related monoalphabetic substitution rules consists of the 26 Caesar ciphers, with shifts of 0 through 25. Each cipher is denoted by a key letter, which is the ciphertext letter that substitutes for the plaintext letter 'a' and which are used in turn, as shown next.
- It makes cryptanalysis harder with more alphabets to guess and flatter frequency distribution. But the Vigenere cipher still do not completely obscure the underlying language characteristics.

CODE:

```
#include<stdio.h>
#include<string.h>
int main(){
    char msg[100], key[20];
    fflush(stdin);
    printf("Enter the plain text : ");
    gets(msg);
    printf("Enter Key : ");
    gets(key);
    int msgLen=0, keyLen=0, i=0, j=0;
    msgLen = strlen(msg);
    keyLen = strlen(key);
    printf("%d \n", msgLen);
    char newKey[msgLen], encryptedMsg[msgLen], decryptedMsg[msgLen];
    for(i = 0, j = 0; i < msgLen; ++i, ++j){
        if(j == keyLen)
            j = 0;
```



```

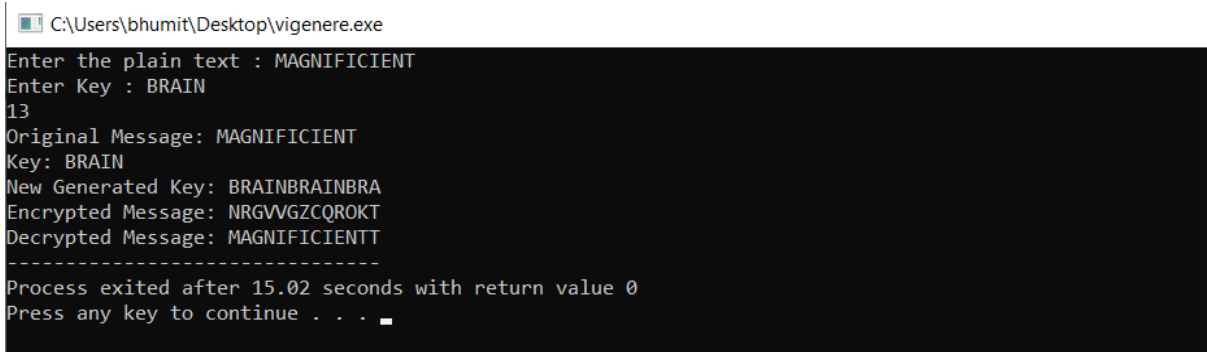
        newKey[i] = key[j];
    }
    newKey[i] = '\0';
//encryption
for(i = 0; i < msgLen; ++i){
    encryptedMsg[i] = ((msg[i] + newKey[i]) % 26) + 'A';
}
encryptedMsg[i] = '\0';

//decryption
for(i = 0; i < msgLen; ++i){
    decryptedMsg[i] = (((encryptedMsg[i] - newKey[i]) + 26) % 26) + 'A';
}
decryptedMsg[i] = '\0';
printf("Original Message: %s", msg);
printf("\nKey: %s", key);
printf("\nNew Generated Key: %s", newKey);
printf("\nEncrypted Message: %s", encryptedMsg);
printf("\nDecrypted Message: %s", decryptedMsg);

    return 0;
}

```

OUTPUT:



```

C:\Users\bhumit\Desktop\vigenere.exe
Enter the plain text : MAGNIFICIENT
Enter Key : BRAIN
13
Original Message: MAGNIFICIENT
Key: BRAIN
New Generated Key: BRAINBRAINBRA
Encrypted Message: NRGVVGZCQROKT
Decrypted Message: MAGNIFICIENTT
-----
Process exited after 15.02 seconds with return value 0
Press any key to continue . . .

```

