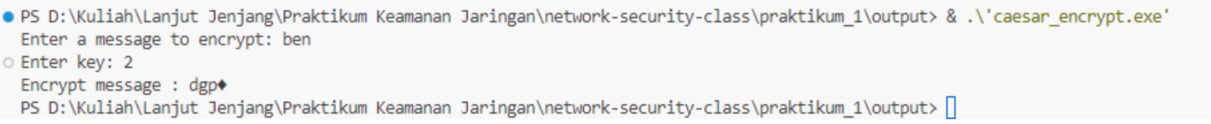
Listing program

- **Caesar Encryption**

```
#include <stdio.h>
int main()
{
    char message[3], ch;
    int i, key;
    printf("Enter a message to encrypt: ");
    gets(message);
    printf("Enter key: ");
    scanf("%d", &key);

    for (i = 0; message[i] != '\0'; ++i)
    {
        ch = message[i];
        if (ch >= 'a' && ch <= 'z')
        {
            ch = ch + key;
            if (ch > 'z')
            {
                ch = ch - 'z' + 'a' - 1;
            }
            message[i] = ch;
        }
        else if (ch >= 'A' && ch <= 'Z')
        {
            ch = ch + key;
            if (ch > 'Z')
            {
                ch = ch - 'Z' + 'A' - 1;
            }
            message[i] = ch;
        }
    }
    printf("Encrypt message : %s", message);
    return 0;
}
```

Output :



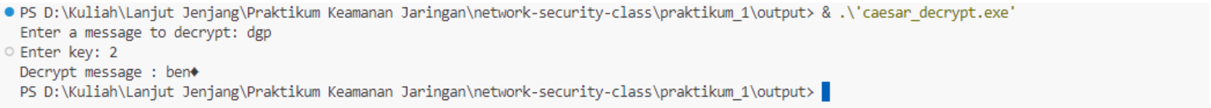
```
● PS D:\Kuliah\Lanjut Jenjang\Praktikum Keamanan Jaringan\network-security-class\praktikum_1\output> & .\'caesar_encrypt.exe'
  Enter a message to encrypt: ben
○ Enter key: 2
  Encrypt message : dgp♦
  PS D:\Kuliah\Lanjut Jenjang\Praktikum Keamanan Jaringan\network-security-class\praktikum_1\output> ▯
```

- **Caesar Decryption**

```
#include <stdio.h>

int main()
{
    char message[3], ch;
    int i, key;

    printf("Enter a message to decrypt: ");
    gets(message);
```

```c
    printf("Enter key: ");
    scanf("%d", &key);

    for (i = 0; message[i] != '\0'; ++i)
    {
        ch = message[i];
        if (ch >= 'a' && ch <= 'z')
        {
            ch = ch - key;
            if (ch > 'z')
            {
                ch = ch + 'z' - 'a' + 1;
            }
            message[i] = ch;
        }
        else if (ch >= 'A' && ch <= 'Z')
        {
            ch = ch - key;
            if (ch > 'Z')
            {
                ch = ch + 'Z' - 'A' + 1;
            }
            message[i] = ch;
        }
    }
    printf("Decrypt message : %s", message);
    return 0;
}
```

Output :

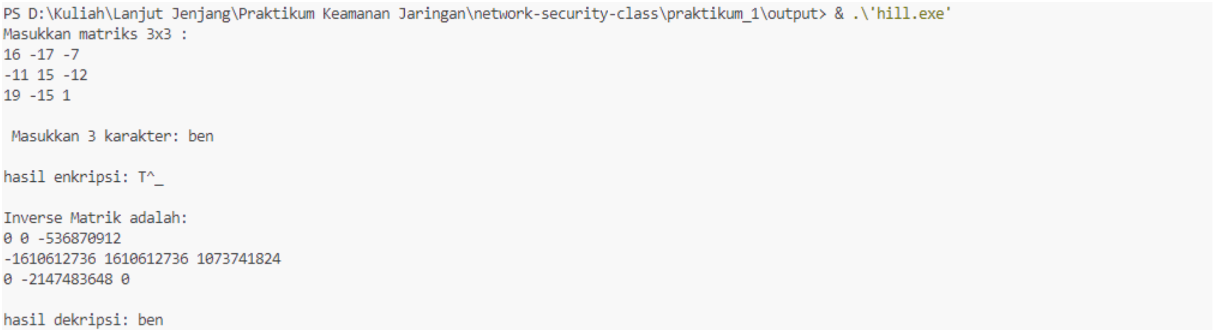


```
● PS D:\Kuliah\Lanjut Jenjang\Praktikum Keamanan Jaringan\network-security-class\praktikum_1\output> & .\'caesar_decrypt.exe'
  Enter a message to decrypt: dgp
○ Enter key: 2
  Decrypt message : ben♦
  PS D:\Kuliah\Lanjut Jenjang\Praktikum Keamanan Jaringan\network-security-class\praktikum_1\output> █
```

- **Hill Encryption & Decryption**

```c
#include <stdio.h>
#include <math.h>

float encrypt[3][1], decrypt[3][1], a[3][3], b[3][3], mes[3][1], c[3][3];

void encryption();
void decryption();
void getKeyMessage();
void inverse();

void main()
{
    getKeyMessage();
    encryption();
    decryption();
}

void getKeyMessage()
{
    int i, j;
    char msg[4]; // Menambah satu elemen untuk karakter null

    printf("Masukkan matriks 3x3 :\n");

    for (i = 0; i < 3; i++)
        for (j = 0; j < 3; j++)
        {
            scanf("%f", &a[i][j]);
            c[i][j] = a[i][j];
        }

    printf("\n Masukkan 3 karakter: ");
    scanf("%s", msg);

    for (i = 0; i < 3; i++)
        mes[i][0] = msg[i] - 97;
}

void encryption()
{
    int i, j, k;

    for (i = 0; i < 3; i++)
        for (j = 0; j < 1; j++)
            for (k = 0; k < 3; k++)
                encrypt[i][j] = encrypt[i][j] + a[i][k] * mes[k][j];

    printf("\nhasil enkripsi: ");
    for (i = 0; i < 3; i++)
        printf("%c", (char)(fmod(encrypt[i][0], 26) + 97));
}

void decryption()
{
    int i, j, k;

    inverse();

    for (i = 0; i < 3; i++)
        for (j = 0; j < 1; j++)
            for (k = 0; k < 3; k++)
                decrypt[i][j] = decrypt[i][j] + b[i][k] * encrypt[k][j];

    printf("\nhasil dekripsi: ");
    for (i = 0; i < 3; i++)
        printf("%c", (char)(fmod(decrypt[i][0], 26) + 97));

    printf("\n");
}

void inverse()
{
    int i, j, k;
    float p, q;
    for (i = 0; i < 3; i++)
        for (j = 0; j < 3; j++)
        {
            if (i == j)
                b[i][j] = 1;
            else
                b[i][j] = 0;
        }

    for (k = 0; k < 3; k++)
    {
        for (i = 0; i < 3; i++)
        {
            p = c[i][k];
            q = c[k][k];

            for (j = 0; j < 3; j++)
            {
                if (i != k)
                {
                    c[i][j] = c[i][j] * q - p * c[k][j];
                    b[i][j] = b[i][j] * q - p * b[k][j];
                }
            }
        }
    }

    for (i = 0; i < 3; i++)
        for (j = 0; j < 3; j++)
            b[i][j] = b[i][j] / c[i][i];

    printf("\n\nInverse Matrik adalah:\n");
    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 3; j++)
            printf("%d ", b[i][j]);

        printf("\n");
    }
}
```

Output



```
PS D:\Kuliah\Lanjut Jenjang\Praktikum Keamanan Jaringan\network-security-class\praktikum_1\output> & .\'hill.exe'
Masukkan matriks 3x3 :
16 -17 -7
-11 15 -12
19 -15 1

 Masukkan 3 karakter: ben

hasil enkripsi: T^_

Inverse Matrik adalah:
0 0 -536870912
-1610612736 1610612736 1073741824
0 -2147483648 0

hasil dekripsi: ben
```