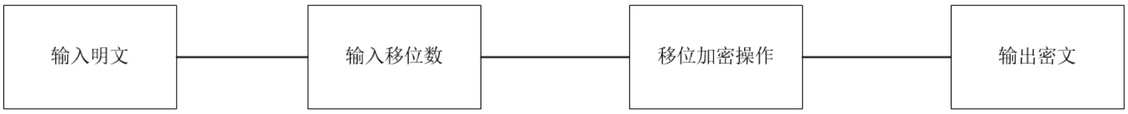
密码学第一次实验报告

1811430王瀚威

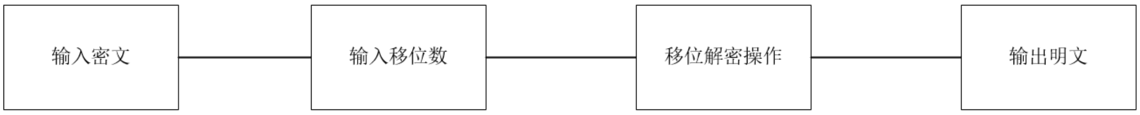
1.移位密码加密解密

加密流程图：



移位加密操作加密过程可简单地写成：
明文：m = m1m2...mi..., 则有
密文：c=c1c2...ci..., 其中 ci=(mi+key mod26)，i = 1，2，...。

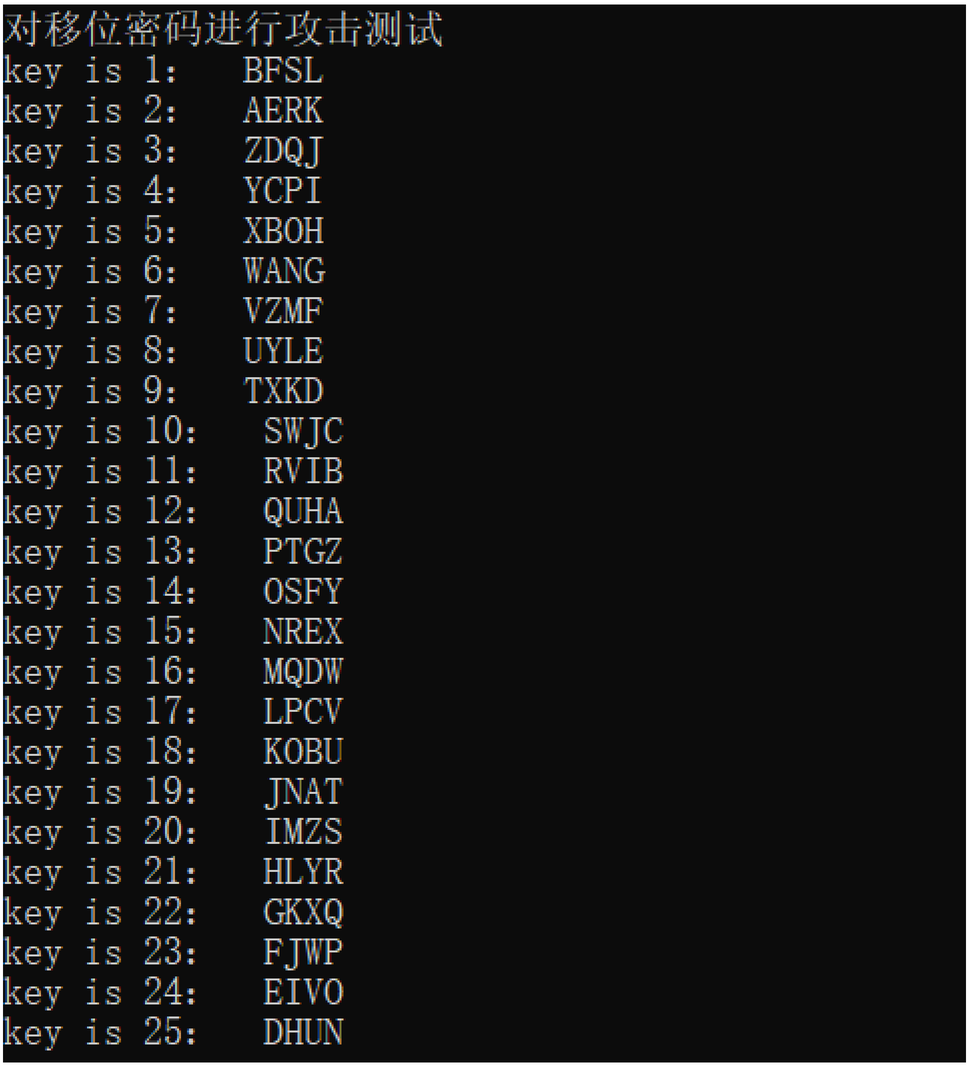
解密流程图：



移位解密操作解密过程可简单地写成：
明文：m = m1m2...mi..., 则有
密文：c=c1c2...ci..., 其中 ci=(mi-key mod26)，i = 1，2，...。

2.对移位密码的攻击

移位密码是一种最简单的密码，其有效密钥空间大小为25，因此可以用穷举的方法观察其解密后的输出，判断适合的明文及其对应的移位数，因此遍历移位数进行暴力破解即可。实例如下：



```
对移位密码进行攻击测试
key is 1:    BFSL
key is 2:    AERK
key is 3:    ZDQJ
key is 4:    YCPI
key is 5:    XBOH
key is 6:    WANG
key is 7:    VZMF
key is 8:    UYLE
key is 9:    TXKD
key is 10:    SWJC
key is 11:    RVIB
key is 12:    QUHA
key is 13:    PTGZ
key is 14:    OSFY
key is 15:    NREX
key is 16:    MQDW
key is 17:    LPCV
key is 18:    KOBU
key is 19:    JNAT
key is 20:    IMZS
key is 21:    HLYR
key is 22:    GKXQ
key is 23:    FJWP
key is 24:    EIVO
key is 25:    DHUN
```

由此可知，移位数为6，明文为WANG。

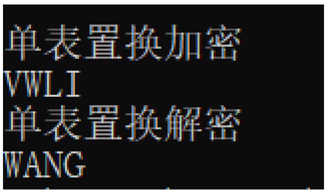
3.单表置换代码

单表置换密码就是根据字母表的置换对明文进行变换的方法，这里我们选定置换表为

ABCDEFGHIJKLMNOPQRSTUVWXYZ 对应

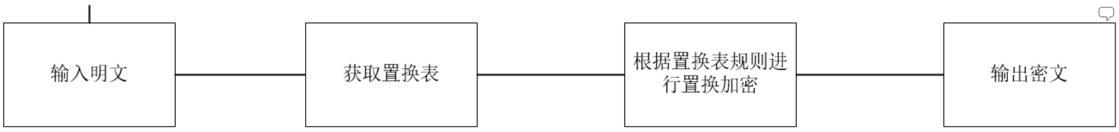
WANGHEIBCDFJKLMOPQRSTUVXYZ

在这个单表置换下，明文WANG对应的密文为VWLI。



```
单表置换加密
VWLI
单表置换解密
WANG
```

流程图如下:



4.对单表置换密码的攻击方法

在单表置换密码中，由于置换表字母组合方式有26！种，约为4.03×1026。所以采用穷举密钥的方法不是一种最有效的方法，因此我们利用自然语言的使用频率和英文的一些显著特征进行猜解。结果如下：

密文：

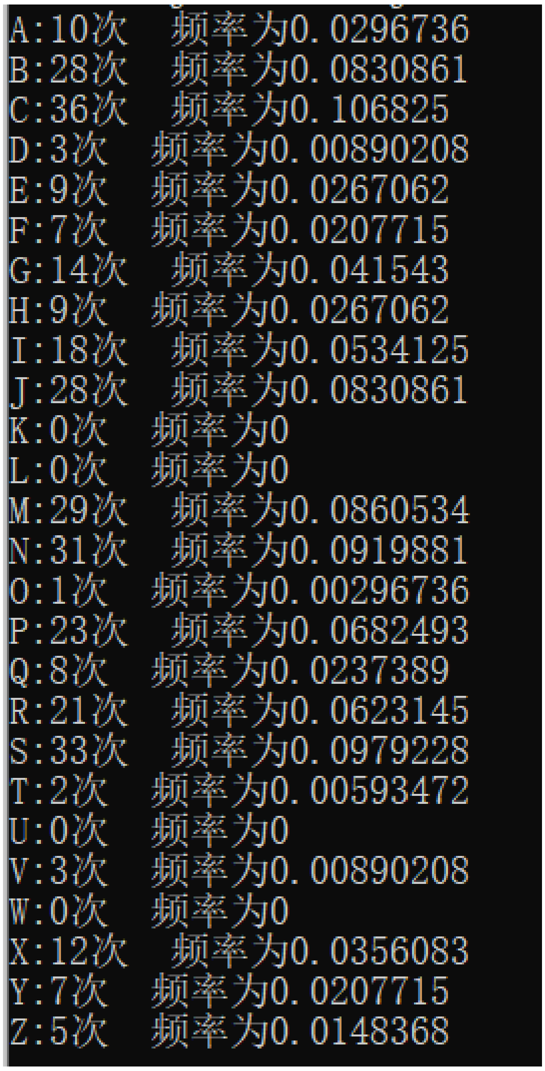
SIC GCBSPNA XPMHACQ JB GPYXSMEPNXIY JR SINS MF SPNBRQJSSJBE JBFMPQNSJMB FPMQ N XMJBS N SM N XMJBS H HY QCNBR MF N XMRRJHAY JBRCGZPC GINBBCA JB RZGI N VNY SINS SIC MPJEJBNA QCRRNEC GNB MBAY HC PCGMTCPCD HY SIC PJEISFZA PCGJXJCBSR SIC XNPSJGJXNBSR JB SIC SPNBRNGSJMB NPC NAJGC SIC MPJEJBNSMP MF SIC QCRRNEC HMH SIC PCGCJTCP NBD MRGNP N XMRRJHAC MXXMBCBS VIM VJRICR SM ENJB ZBNZSIMPJOCD GMBSPMA MF SIC QCRRNEC

明文：

THE CENTRAL PROBLEM IN CRYPTOGRAPHY IS THAT OF TRANSMITTING INFORMATION FROM A POINT A TO A POINT B BY MEANS OF A POSSIBLY INSECURE CHANNEL IN SUCH A WAY THAT THE ORIGINAL MESSAGE CAN ONLY BE RECOVERED BY THE RIGHTFUL RECIPIENTS THE PARTICIPANTS IN THE TRANSACTION ARE ALICE THE ORIGINATOR OF THE MESSAGE BOB THE RECEIVER AND OSCAR A POSSIBLE OPPONENT WHO WISHES TO GAIN UNAUTHORIZED CONTROL OF THE MESSAGE

猜解过程如下

首先，对出现字母进行频率分析，得到结果如下：



```
A:10次  频率为0.0296736
B:28次  频率为0.0830861
C:36次  频率为0.106825
D:3次  频率为0.00890208
E:9次  频率为0.0267062
F:7次  频率为0.0207715
G:14次  频率为0.041543
H:9次  频率为0.0267062
I:18次  频率为0.0534125
J:28次  频率为0.0830861
K:0次  频率为0
L:0次  频率为0
M:29次  频率为0.0860534
N:31次  频率为0.0919881
O:1次  频率为0.00296736
P:23次  频率为0.0682493
Q:8次  频率为0.0237389
R:21次  频率为0.0623145
S:33次  频率为0.0979228
T:2次  频率为0.00593472
U:0次  频率为0
V:3次  频率为0.00890208
W:0次  频率为0
X:12次  频率为0.0356083
Y:7次  频率为0.0207715
Z:5次  频率为0.0148368
```

具体为

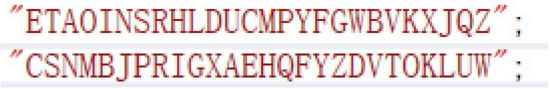
c ------- 0.106825 s ------- 0.0979228 n ------- 0.0919881 m ------- 0.0860534 b ------- 0.0830861 j ------- 0.0830861 p ------- 0.0682493 r ------- 0.0623145 i ------- 0.0534125   g ------- 0.041543

x ------- 0.0356083 a ------- 0.0296736 e ------- 0.0267062 h ------- 0.0267062 q ------- 0.0237389

f ------- 0.0207715 y ------- 0.0207715 z ------- 0.0148368 d ------- 0.00890208 v ------- 0.00890208

t ------- 0.00593472 o ------- 0.00296736   频率为0的暂时不写入

将出现频率最高的c和s替换成英文中出现频率最高的e和t，由于多次单独出现字母n，因此可以猜测n对应的明文为a，其余字母按照频率进行依次对应，构建置换表如下：



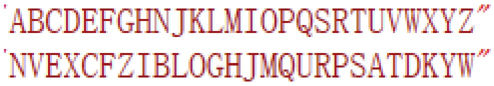
```
"ETAOINSRHLDUCMPYFGWBVKXJQZ";
"CSNMBJPRIGXAEHQFYZDVTOKLUW";
```

猜解结果为：

THE LEITSAU DSOMUEP NI LSFDTOCSADHF NR THAT OY TSAIRPNTTNIC NIYOSPATNOI YSOP A DONIT A TO A DONIT M MF PEAIR OY A DORRNMUF NIRELGSE LHAIIEU NI RGLH A BAF THAT THE OSNCNIAU PERRACE LAI OIUF ME SELOVESEW MF THE SNCHTYGU SELNDNEITR THE DASTNLNDAITR NI THE TSAIRALTNOI ASE AUNLE THE OSNCNIATOS OY THE PERRACE MOM THE SELENVES AIW ORLAS A DORRNMUE ODDOIEIT BHO BNRHER TO CANI GIAGTHOSNKEW LOITSOU OY THE PERRACE

发现THAT完整且合理的单词，因此H置换为I为正确的，此外可以进一步修改，将DOINT改为POINT，即X对应P，此外，根据英语语法可以将OY改为OF，NR改为IS，得到 IS THAT OF 的搭配

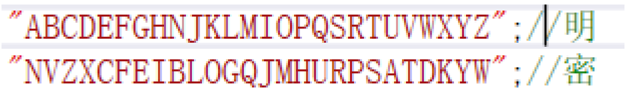
新的置换表和得到新的明文如下：



```
"ABCDEFGHNJKLMIOPQSRTUVWXYZ"
"NVEXCFZIBLOGHJMQURPSATDKYW"
```

THE LENTRAU DROMUEP IN LRYDTOCRADHY IS THAT OF TRANSPITTINC INFORPATION FROP A DOINT A TO A DOINT M MY PEANS OF A DOSSIMUY INSELGRE LHANNEU IN SGLH A BAY THAT THE ORICINAU PESSACE LAN ONUY ME RELOVEREW MY THE RICHTFGU RELIDIENTS THE DARTILIDANTS IN THE TRANSALTION ARE AUILE THE ORICINATOR OF THE PESSACE MOM THE RELEIVER ANW OSLAR A DOSSIMUE ODDONENT BHO BISHES TO CAIN GNAGTHORIKEW LONTROU OF THE PESSACE

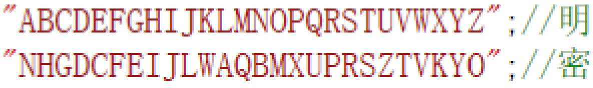
可以再次替换， TRANSPITTINC INFORPATION FROP为TRANSMITTING INFORMATION FROM得到新的置换表和明文如下：



```
"ABCDEFGHNJKLMIOPQSRTUVWXYZ";//明
"NVZXCFEIBLOGQJMHURPSATDKYW";//密
```

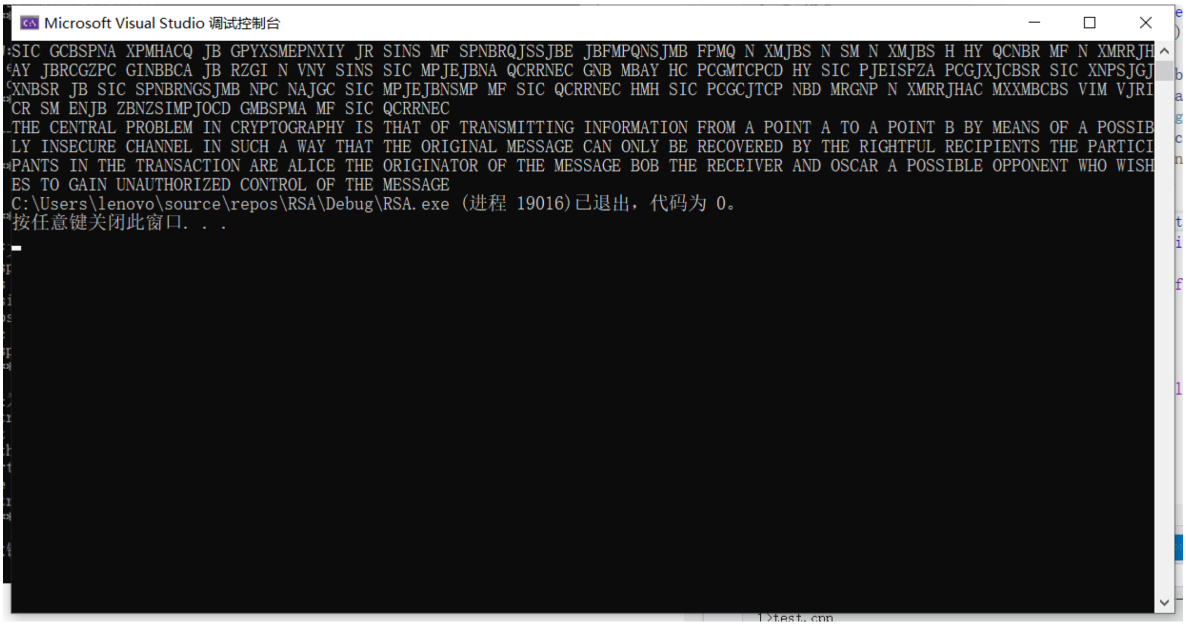
THE LENTRAU DROPUEM IN LRYDTOGRADHY IS THAT OF TRANSMITTING INFORMATION FROM A DOINT A TO A DOINT P PY MEANS OF A DOSSIPUY INSELCRE LHANNEU IN SCLH A BAY THAT THE ORIGINAU MESSAGE LAN ONUY PE RELOVEREW PY THE RIGHTFCU RELIDIENTS THE DARTILIDANTS IN THE TRANSALTION ARE AUILE THE ORIGINATOR OF THE MESSAGE POP THE RELEIVER ANW OSLAR A DOSSIPUE ODDONENT BHO BISHES TO GAIN CNACTHORIKEW LONTROU OF THE MESSAGE

此时显然可以看出各个字母的对应关系如下



```
"ABCDEFGHIJKLMNOPQRSTUVWXYZ";//明
"NHGDCFEIJLWAQBMXUPRSZTVKYO";//密
```

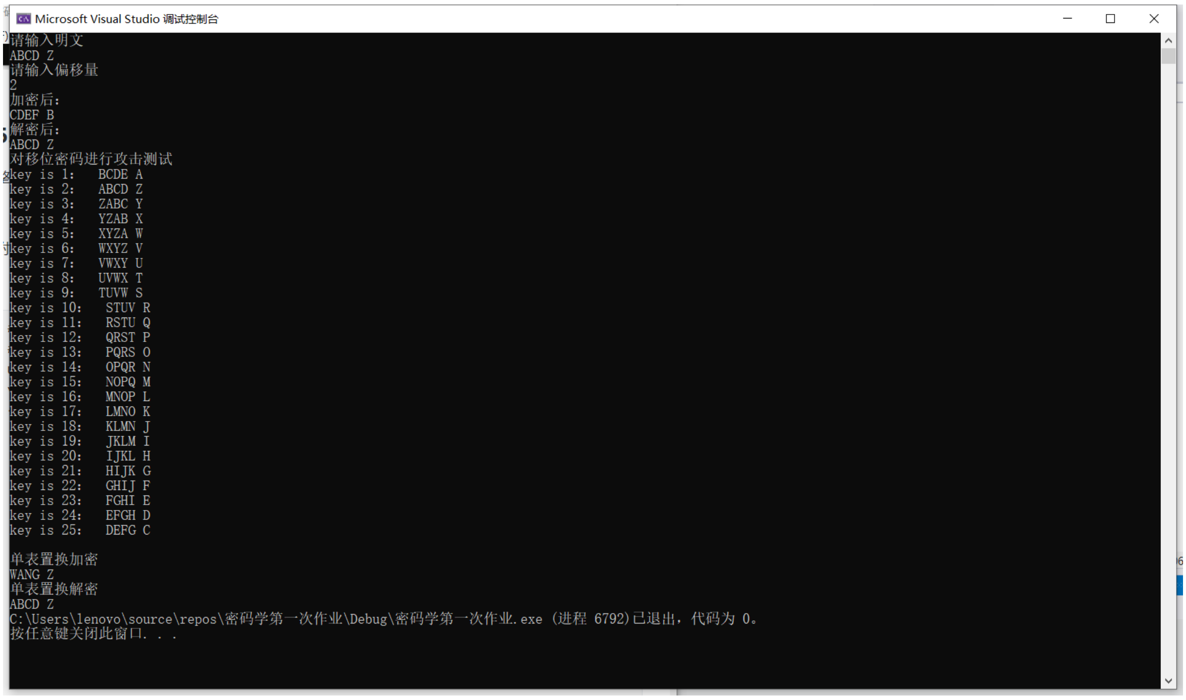
用此时得到的置换表再次进行解密，结果正确。



```
SIC GCBSPNA XPMHACQ JB GPYXSMEPNXIY JR SINS MF SPNBRQJSSJBE JBFMPQNSJMB FPMQ N XMJBS N SM N XMJBS H HY QCNBR MF N XMRRJH
AY JBRCGZPC GINBBCA JB RZGI N VNY SINS SIC MPJEJBNA QCRRNEC GNB MBAY HC PCGMTCPCD HY SIC PJEISFZA PCGJXJCBSR SIC XNPSJGJ
XNBSR JB SIC SPNBRNGSJMB NPC NAJGC SIC MPJEJBNSMP MF SIC QCRRNEC HMH SIC PCGCJTCP NBD MRGNP N XMRRJHAC MXXMBCBS VIM VJRI
CR SM ENJB ZBNZSIMPJOCD GMBSPMA MF SIC QCRRNEC
THE CENTRAL PROBLEM IN CRYPTOGRAPHY IS THAT OF TRANSMITTING INFORMATION FROM A POINT A TO A POINT B BY MEANS OF A POSSIB
LY INSECURE CHANNEL IN SUCH A WAY THAT THE ORIGINAL MESSAGE CAN ONLY BE RECOVERED BY THE RIGHTFUL RECIPIENTS THE PARTICI
PANTS IN THE TRANSACTION ARE ALICE THE ORIGINATOR OF THE MESSAGE BOB THE RECEIVER AND OSCAR A POSSIBLE OPPONENT WHO WISH
ES TO GAIN UNAUTHORIZED CONTROL OF THE MESSAGE
C:\Users\lenovo\source\repos\RSA\Debug\RSA.exe (进程 19016)已退出，代码为 0。
按任意键关闭此窗口. . .
```

5.编程部分效果及代码

整体实验效果截图如下：



```
请输入明文
ABCD Z
请输入偏移量
2
加密后：
CDEF B
解密后：
ABCD Z
对移位密码进行攻击测试
key is 1:   BCDE A
key is 2:   ABCD Z
key is 3:   ZABC Y
key is 4:   YZAB X
key is 5:   XYZA W
key is 6:   WXYZ V
key is 7:   VWXY U
key is 8:   UVWX T
key is 9:   TUVW S
key is 10:   STUV R
key is 11:   RSTU Q
key is 12:   QRST P
key is 13:   PQRS O
key is 14:   OPQR N
key is 15:   NOPQ M
key is 16:   MNOP L
key is 17:   LMNO K
key is 18:   KLMN J
key is 19:   JKLM I
key is 20:   IJKL H
key is 21:   HIJK G
key is 22:   GHIJ F
key is 23:   FGHI E
key is 24:   EFGH D
key is 25:   DEFG C

单表置换加密
WANG Z
单表置换解密
ABCD Z
C:\Users\lenovo\source\repos\密码学第一次作业\Debug\密码学第一次作业.exe (进程 6792)已退出，代码为 0。
按任意键关闭此窗口. . .
```

附源代码：

```
#include <iostream>
#include <string.h>
using namespace std;
void enc(char* before, char* after, int num)
{
    for (int i = 0; i < strlen(before); i++)
    {
        if (int(before[i]) <= int('z') && int(before[i]) >= int('a'))
        {
            after[i] = int('a') + (int(before[i]) + num - int('a')) % 26;
        }
        else if (int(before[i]) <= int('Z') && int(before[i]) >= int('A'))
        {
            after[i] = int('A') + ((int(before[i]) + num - int('A')) % 26);
        }
        else
        {
            after[i] = before[i];
        }
    }
}
void dec(char* before, char* after, int num)
{
    for (int i = 0; i < strlen(before); i++)
    {
        if (int(before[i]) <= int('z') && int(before[i]) >= int('a'))
        {
            after[i] = int('a') + (int(before[i]) + 26 - num - int('a')) % 26;
        }
        else if (int(before[i]) <= int('Z') && int(before[i]) >= int('A'))
        {
            after[i] = int('A') + ((int(before[i]) + 26 - num - int('A')) % 26);
        }
        else
        {
            after[i] = before[i];
        }
    }
}
int main() {
    int num;//偏移量

    //cout << (-1) % 26;
    char to_enc[256];
    char after_enc[256];
    char after_dec[256];
    cout << "请输入明文" << endl;
    cin.getline(to_enc, 255);
    int n = strlen(to_enc);
    cout << "请输入偏移量 " << "    " << endl;
    cin >> num;
    after_enc[n] = 0;
    enc(to_enc, after_enc, num);
    cout << "加密后：" << endl;
    for (int i = 0; i < n; i++)
        cout << after_enc[i];

    dec(after_enc, after_dec, num);
    cout << endl << "解密后：" << endl;
    for (int i = 0; i < n; i++)
        cout << after_dec[i];

    cout << endl << "对移位密码进行攻击测试" << endl;
    for (int i = 1; i <= 25; i++)
    {
        dec(after_enc, after_dec, i);
        cout << "key is " << i << "：  ";
        for (int i = 0; i < n; i++)
            cout << after_dec[i];
        cout << endl;
    }
    /*
    字母表加解密密
        cout << endl << "用字母表" << endl;
        char a[27] = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
        for (int i = 0; i < strlen(to_enc); i++)
        {
            if (int(to_enc[i]) <= int('Z') && int(to_enc[i]) >= int('A'))
            {
                after_enc[i] = a[(int(to_enc[i]) - int('A') + num) % 26];
            }
            else
            {
                after_enc[i] = to_enc[i];
            }
        }
        for (int i = 0; i < strlen(to_enc); i++)
            cout << after_enc[i];

            */

            //置换
    char a[27] = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
    char ZHB[27] = "WANGHEIBCDFJKLMOPQRSTUVXYZ";
    //NVEXCYZIBLOGHJMQURPSATDKFW
    cout << endl << "单表置换加密" << endl;
    for (int i = 0; i < strlen(to_enc); i++)
    {
        if (to_enc[i] == ' ')
        {
            after_enc[i] = ' ';
            continue;
        }
        else
            for (int j = 0; j <= 25; j++)
                if (to_enc[i] == a[j])
                {
                    after_enc[i] = ZHB[j];
                    break;
                }
    }
    for (int i = 0; i < strlen(to_enc); i++)
        cout << after_enc[i];

    cout << endl << "单表置换解密" << endl;
    for (int i = 0; i < strlen(after_enc); i++)
    {
        if (after_enc[i] == ' ')
        {
            after_dec[i] = ' ';
            continue;
        }
        else
            for (int j = 0; j <= 25; j++)
                if (after_enc[i] == ZHB[j])
                {
                    after_dec[i] = a[j];
                    break;
                }
    }
    for (int i = 0; i < strlen(after_enc); i++)
    cout << after_dec[i];
    /*cout << endl << "单表置换解密" << endl;
    for (int i = 0; i < strlen(after_enc); i++)
    {
        if (int(after_enc[i]) <= int('Z') && int(after_enc[i]) >= int('A'))
        {
            after_dec[i] = ZHB[(int(after_enc[i]) - int('A')) % 26];
        }
        else
        {
            after_dec[i] = after_enc[i];
        }
    }
    for (int i = 0; i < strlen(after_enc); i++)
        cout << after_dec[i];*/
}
```