Lab1——古典密码算法及攻击方法

学号：2013921
姓名：周延霖
年级：2020级
专业：信息安全

一、实验内容说明

1、实验目的

通过C＋＋编程实现移位密码和单表置换密码算法，加深对经典密码体制的了解。并通过对这两种密码实施攻击，了解对古典密码体制的攻击方法。

2、实验要求

要求上述密码算法提供最后的算法流程图，并写出明文、加解密的结果。字母频率统计攻击方法要求写明置换表中确定每个字母的原因和攻击的步骤。

3、实验步骤

1. 根据实验原理部分对移位密码算法的介绍，自己创建明文信息，并选择一个密钥，编写移位密码算法实现程序，实现加密和解密操作。
2. 两个同学为一组，互相攻击对方用移位密码加密获得的密文，恢复出其明文和密钥。
3. 自己创建明文信息，并选择一个密钥，构建置换表。编写置换密码的加解密实现程序，实现加密和解密操作。
4. 用频率统计方法，试译下面用单表置换加密的一段密文：

```
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
```

二、实验环境

- 操作系统：macOS Montery 12.4
- 软件系统：Xcode
- 编译工具：Apple clang version 13.1.6 (clang-1316.0.21.2.5)
- 编程语言：C++

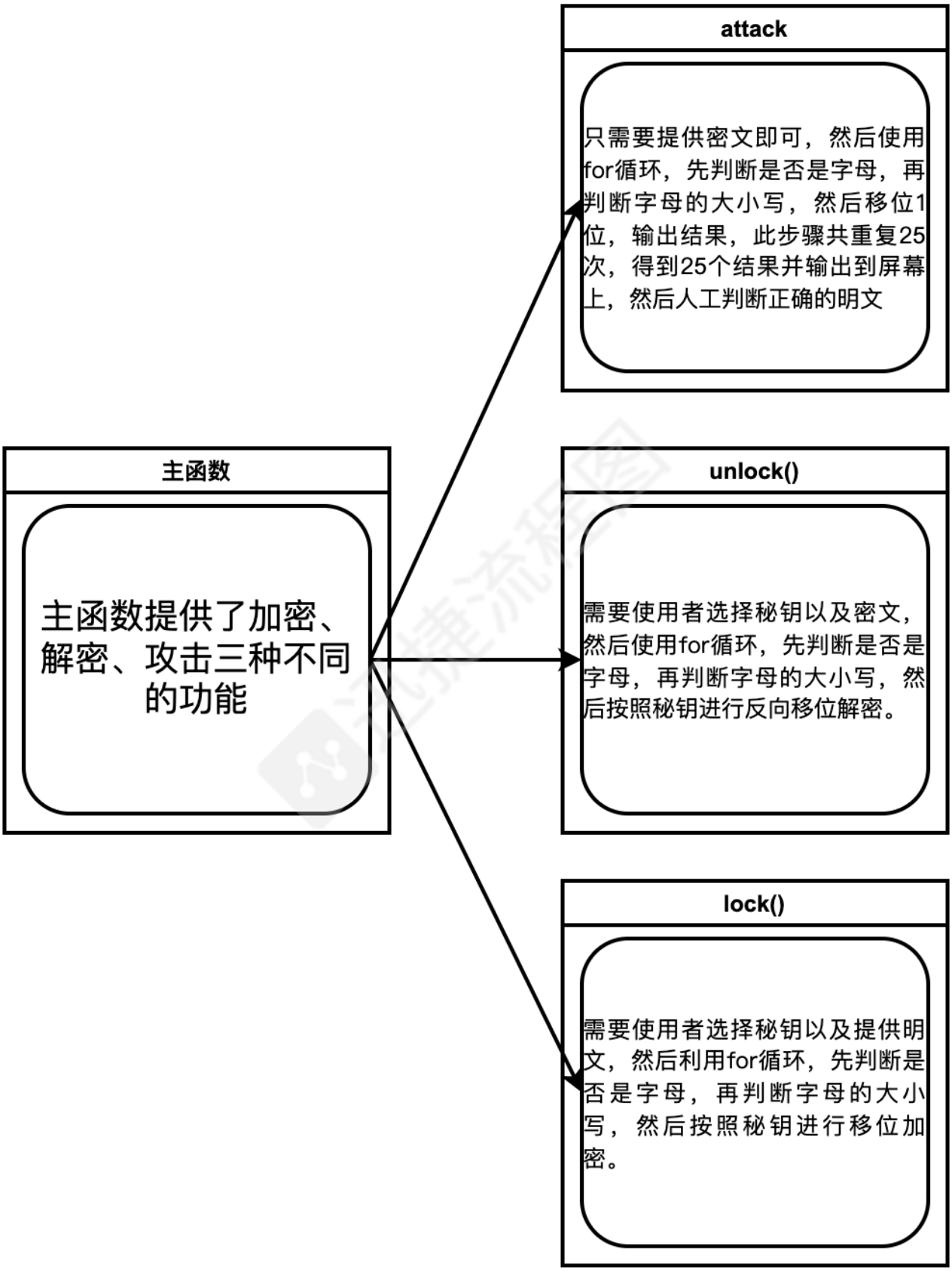
三、实验过程

本次实验首先编写移位密码的加密解密以及攻击程序，然后编写单表置换密码的加密解密程序，最后破译老师所给的密文，以下为具体过程：

1、移位密码

> 流程图

- 移位密码的加密流程图如下图所示：



加密解密过程

- 加密函数首先需要用户选择密钥
- 然后输入自己需要加密的密文
- 通过for循环来对其加密，不是字母的字符不会变化，是字母的字符进行移位加密

- 最后输出加密结果，具体函数如下：

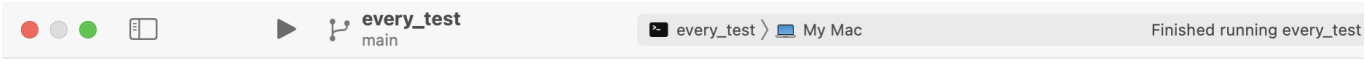
```
void lock()
{
    cout << "请选择你想使用的密钥: " << endl;
    int a;
    cin >> a;
    cin.ignore();
    a = a % 26;
    cout << "请输入你想要加密的字符串: " << endl;
    string str;
    getline(cin, str);
    for (int i = 0; i < str.length(); i++)
    {
        if (64 < str[i] && str[i] < 91)
        {
            str[i] = (str[i] + a) % 91;
            if (str[i] < 65)
                str[i] += 65;
        }
        else
        {
            if (96 < str[i] && str[i] < 123)
            {
                str[i] = (str[i] + a) % 123;
                if (str[i] < 97)
                    str[i] += 97;
            }
        }
    }
    cout << "=====================================================" <<
endl;
    cout << "加密结果为: " << endl;
    cout << str << endl;
    cout << "=====================================================" <<
endl;
}
```

- 解密函数首先需要用户选择密钥
- 然后输入自己需要解密的密文
- 通过for循环来对其解密，不是字母的字符不会变化，是字母的字符进行移位解密
- 最后输出解密结果，具体函数如下：

```
void unlock()
{
    cout << "请选择你想使用的密钥: " << endl;
    int a;
    cin >> a;
    cin.ignore();
    a = a % 26;
```

```
    cout << "请输入你想要解密的字符串: " << endl;
    string str;
    getline(cin, str);
    for (int i = 0; i < str.length(); i++)
    {
        if (64 < str[i] && str[i] < 91)
        {
            str[i] = str[i] - a;
            if (str[i] < 65)
                str[i] += 26;
        }
        else
        {
            if (96 < str[i] && str[i] < 123)
            {
                str[i] = str[i] - a;
                if (str[i] < 97)
                    str[i] += 26;
            }
        }
    }
    cout << "==================================================" <<
endl;
    cout << "解密结果为: " << endl;
    cout << str << endl;
    cout << "==================================================" <<
endl;
}
```

- 最后的加解密输出结果如下:




```
==================================================
粥小霖的移位加密器，以下为你可以进行的操作：
1.加密
2.解密
3.攻击
4.退出
==================================================
请输入相应数字进行你想要的操作：
1
请选择你想使用的密钥：
7
请输入你想要加密的字符串：
I love you!
==================================================
加密结果为：
P svcl fvb!
==================================================
==================================================
粥小霖的移位加密器，以下为你可以进行的操作：
1.加密
2.解密
3.攻击
4.退出
==================================================
请输入相应数字进行你想要的操作：
2
请选择你想使用的密钥：
7
请输入你想要解密的字符串：
P svcl fvb!
==================================================
解密结果为：
I love you!
==================================================
```

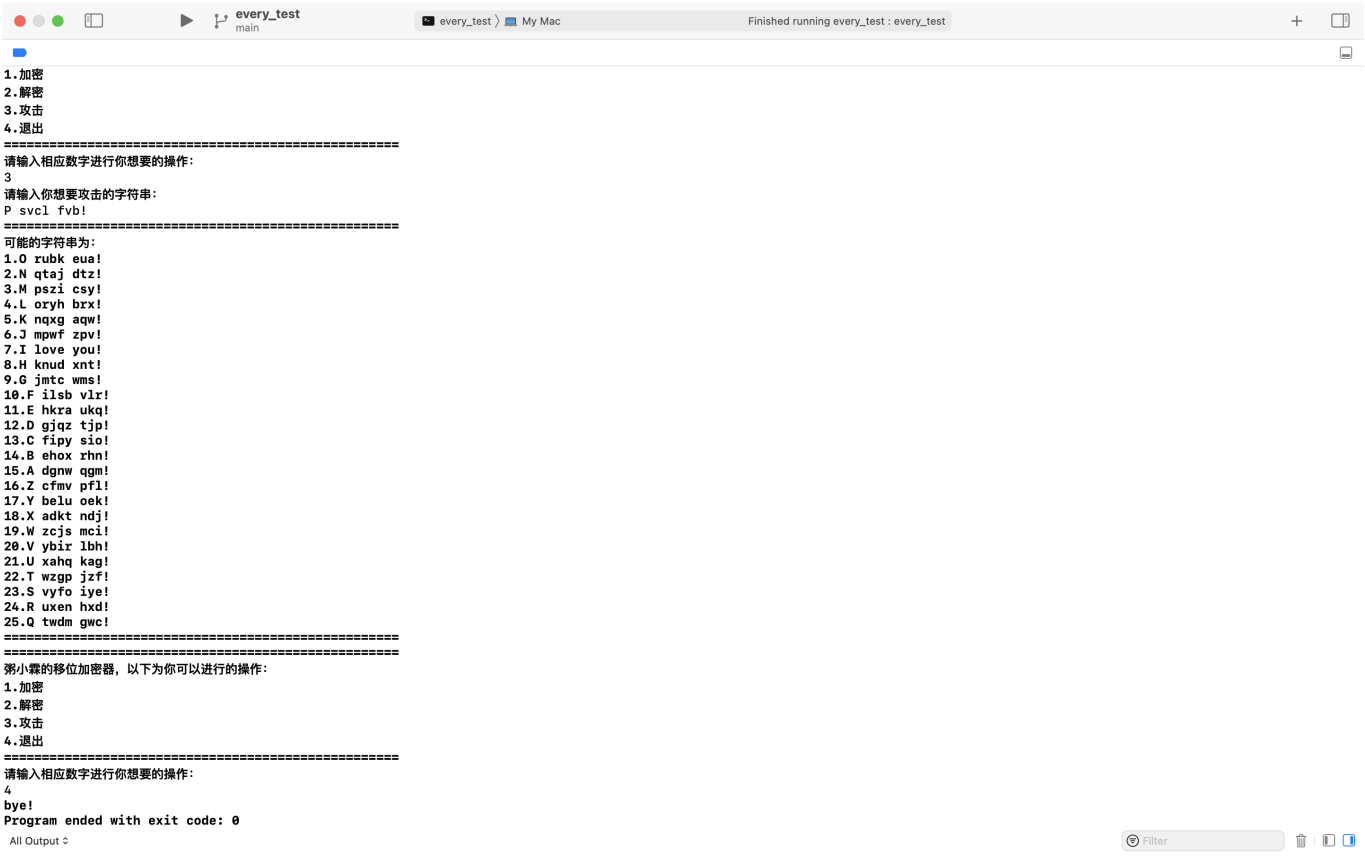
攻击过程

- 攻击函数首先需要用户输入需要攻击的字符串
- 通过for循环来对其攻击，一共需要进行25次
- 最后输出25个结果，并由人工判断正确的结果，具体函数如下：

```
void attack()
{
    cin.ignore();
    cout << "请输入你想要攻击的字符串：" << endl;
    string str;
    getline(cin, str);
    cout << "==================================================" <<
endl;
    cout << "可能的字符串为：" << endl;
    for (int j = 1; j < 26; j++)
    {
        for (int i = 0; i < str.length(); i++)
        {
            if (64 < str[i] && str[i] < 91)
            {
                str[i] = str[i] - 1;
                if (str[i] < 65)
                    str[i] += 26;
            }
            else
```

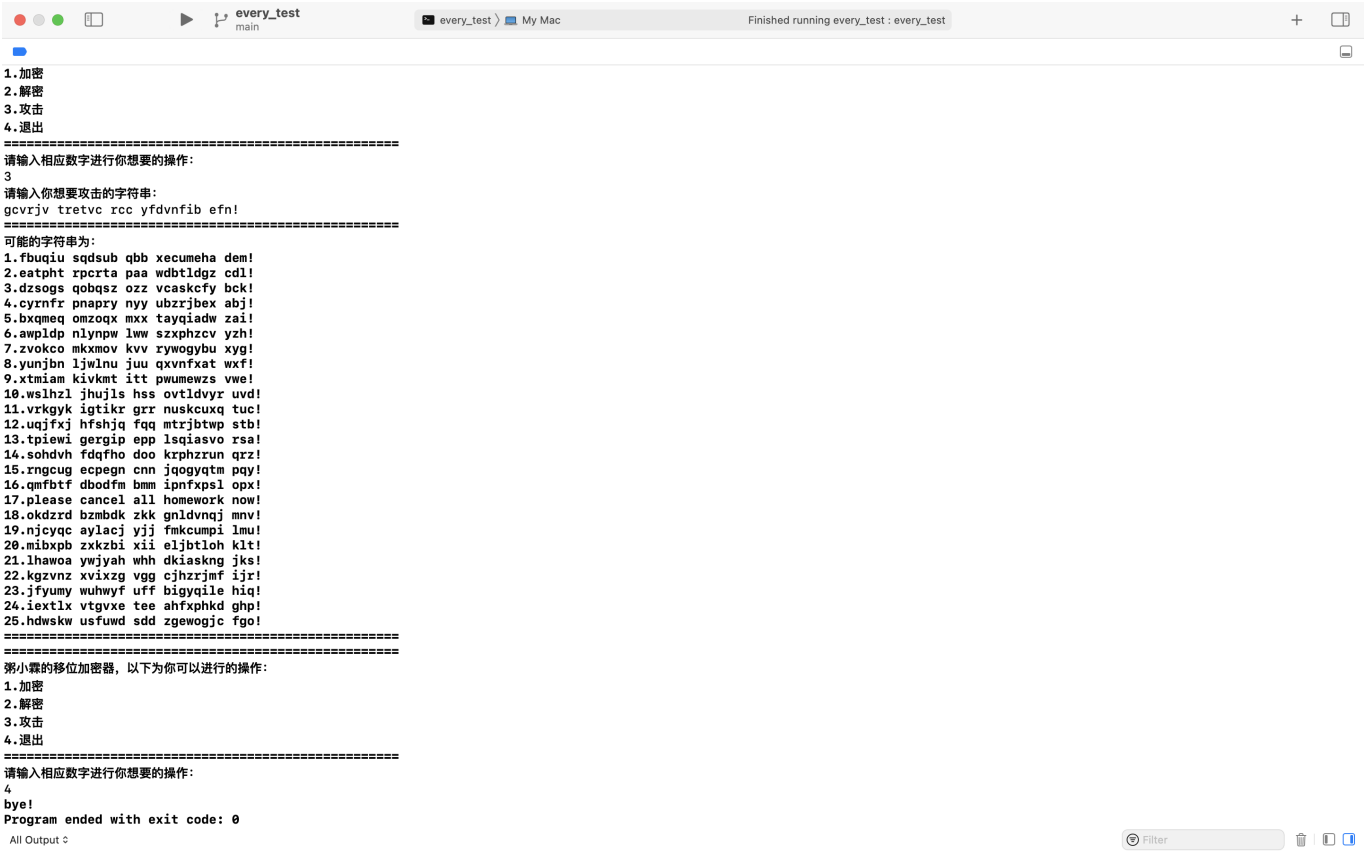
```
            {
                if (96 < str[i] && str[i] < 123)
                {
                    str[i] = str[i] - 1;
                    if (str[i] < 97)
                        str[i] += 26;
                }
            }
        }
        cout << j << "." << str << endl;
    }
    cout << "==================================================" <<
endl;
}
```

- 最后的攻击输出结果如下：
  - 可以判断字符串为`7.I love you!`



与队友互相攻击

- 队友魏伯繁给我的密文为`gcvrjv tretvc rcc yfdvnfib efn!`
  - 运行程序后的结果如下图所示：

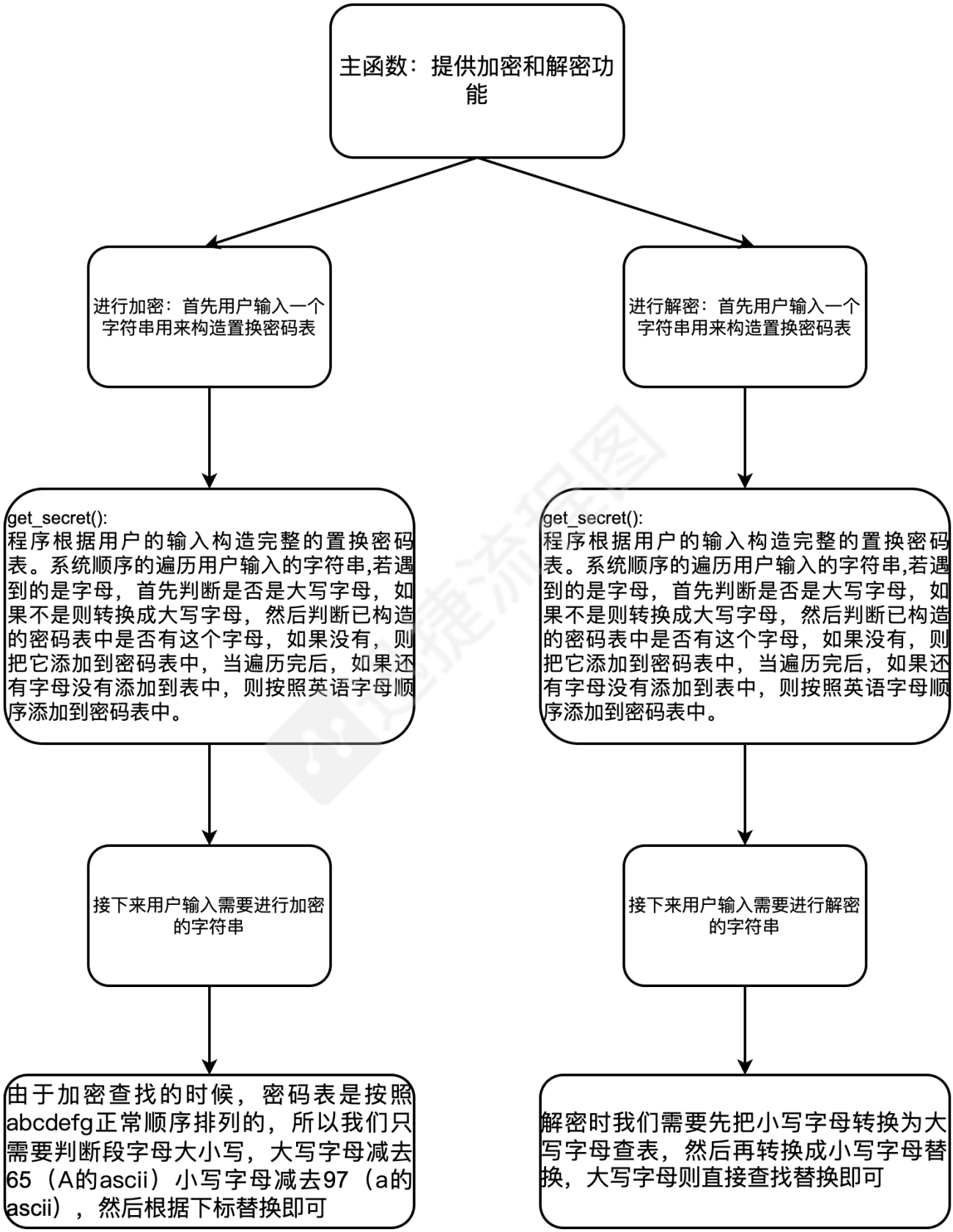


- 通过输出结果可以看到应该正确的明文为`please cancel all homework now!`

2、单表置换密码

流程图

- 单表置换密码的加密流程图如下图所示：



置换密码表原理

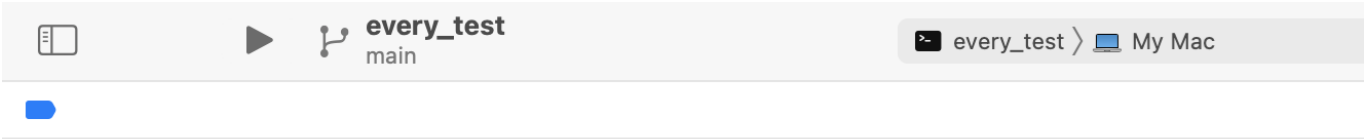
单表置换实现的一个关键问题是关于置换表的构造。置换表的构造可以有各种不同的途径，主要考虑的是记忆的方便。如使用一个短语或句子，删去其中的重复部分，作为置换表的前面的部分，然后把没有用到的字母按字母表的顺序依次放入置换表中，本次实验就是用这种方式来构造的。

加密解密过程

- 加密函数首先需要用户输入一个字符串用来构造密码表
- 然后调用get_secret()函数来构造密码表并在屏幕上显示
- 然后输入自己需要加密的字符串
- 通过for循环并按照密码表来对其加密，不是字母的字符不会变化，是字母的字符进行加密
- 最后输出加密结果，具体函数如下：

```
void lock()
{
    cin.ignore();
    cout << "请输入一个你喜欢的短语以构造置换密码表: " << endl;
    getline(cin, str2);
    str2 = get_secret(str2);
    cout << "请输入你想要加密的字符串: " << endl;
    getline(cin, a);
    for (int i = 0; i < a.length(); i++)
    {
        if (64 < a[i] && a[i] < 91)
        {
            a[i] = str2[a[i] - 65];
        }
        else
        {
            if (96 < a[i] && a[i] < 123)
                a[i] = str2[a[i] - 97] + 32;
        }
    }
    cout << "==========================================================" <<
endl;
    cout << "加密结果为: " << endl;
    cout << a << endl;
    cout << "==========================================================" <<
endl;
}
```

- 加密结果如下图所示：




```
==================================================
粥小霖的单表置换加密器，以下为你可以进行的操作：
1.加密
2.解密
3.退出
==================================================
请输入相应数字进行你想要的操作：
1
请输入一个你喜欢的短语以构造置换密码表：
I love you!
==================================================
根据输入所构造的密码表为：
ABCDEFGHIJKLMNOPQRSTUVWXYZ
ILOVEYUABCDFGHJKMNPQRSTWXZ
==================================================
请输入你想要加密的字符串：
Do you love me?
==================================================
加密结果为：
Vj xjr fjse ge?
==================================================
```

- 解密函数首先需要用户输入一个字符串用来构造密码表
- 然后调用get_secret()函数来构造密码表并在屏幕上显示
- 然后输入自己需要解密的字符串
- 通过for循环并按照密码表来对其解密，不是字母的字符不会变化，是字母的字符进行解密
- 最后输出加密解密，具体函数如下：

```
void unlock()
{
    cin.ignore();
    cout << "请输入一个你喜欢的短语以构造置换密码表：" << endl;
    getline(cin, str2);
    str2 = get_secret(str2);
    cout << "请输入你想要解密的字符串：" << endl;
    getline(cin, a);
    for (int i = 0; i < a.length(); i++)
    {
        if (64 < a[i] && a[i] < 91)
        {
            for (int j = 0; j < 26; j++)
            {
                if (a[i] == str2[j])
                {
                    a[i] = str1[j];
                    break;
                }
            }
        }
        else
```

```
        {
            if (96 < a[i] && a[i] < 123)
            {
                for (int j = 0; j < 26; j++)
                {
                    if (a[i] - 32 == str2[j])
                    {
                        a[i] = str1[j] + 32;
                        break;
                    }
                }
            }
        }
    }
    cout << "==================================================" <<
endl;
    cout << "解密结果为: " << endl;
    cout << a << endl;
    cout << "==================================================" <<
endl;
}
```

- 解密结果如下图所示：

```
==================================================
粥小霖的单表置换加密器，以下为你可以进行的操作：
1.加密
2.解密
3.退出
==================================================
请输入相应数字进行你想要的操作：
2
请输入一个你喜欢的短语以构造置换密码表：
I love you!
==================================================
根据输入所构造的密码表为：
ABCDEFGHIJKLMNOPQRSTUVWXYZ
ILOVEYUABCDFGHJKMNPQRSTWXZ
==================================================
请输入你想要解密的字符串：
Vj xjr fjse ge?
==================================================
解密结果为：
Do you love me?
==================================================
==================================================
粥小霖的单表置换加密器，以下为你可以进行的操作：
1.加密
2.解密
3.退出
==================================================
请输入相应数字进行你想要的操作：
3
bye!
Program ended with exit code: 0
```

All Output

- 最后调用的get_secret()函数代码如下：

```
string get_secret(string str)
{
    string secret(26,' ');
    int num = 0;
    for (int i = 0; i < str.length(); i++)
    {
        if ((96 < str[i] && str[i] < 123) || (64 < str[i] && str[i] < 91))
        {
            if (96 < str[i] && str[i] < 123)
                str[i] -= 32;
            for (int j = 0; j <= num; j++)
            {
                if (str[i] == secret[j])
                    break;
                else
                {
                    if (j == num)
                    {
                        secret[j] = str[i];
                        num++;
                        break;
                    }
                }
            }
        }
    }
    for (int i = 0; i < 26; i++)
    {
        for (int j = 0; j < num; j++)
        {
            if (str1[i] == secret[j])
                break;
            else
            {
                if (j == num - 1)
                {
                    secret[j + 1] = str1[i];
                    num++;
                    break;
                }
            }
        }
    }
    cout << "==================================================" <<
endl;
    cout << "根据输入所构造的密码表为: " << endl << str1 << endl << secret <<
endl;
    cout << "==================================================" <<
endl;
    return secret;
}
```

3、破译密文

密文在实验步骤处已经列出，这里就不再进行赘述，先对其进行分析并给出最后解密出的明文：

> 对密文进行分析

1、单独字母

我们可以看到图中单独出现的字母是N和H，各个字母出现的次数如下表所示：

字母	次数	字母	次数	字母	次数	字母	次数
A	10	B	28	C	36	D	3
E	9	F	7	G	14	H	9
I	18	J	28	K	0	L	0
M	29	N	31	O	1	P	23
Q	8	R	21	S	33	T	2
U	0	V	3	W	0	X	12
Y	7	Z	5				

但是正常字母出现的概率为：

字母	概率	字母	概率	字母	概率	字母	概率	字母	概率	字母	概率	字母	概率
e	11.67	t	9.53	o	8.22	i	7.81	a	7.73	n	6.71	s	6.55
r	5.97	h	4.52	l	4.3	d	3.24	u	3.21	c	3.06	m	2.8
p	2.34	y	2.22	f	2.14	g	2.00	w	1.69	b	1.58	v	1.03
k	0.79	x	0.30	j	0.23	q	0.12	z	0.09				

2、攻击分析

1. 首先我们对表中的单个单词做分析，在英语中，可以单个出现的只有I、A，对应题目中的单词H、N
2. 接着我们尝试找到密文中出现频率最高的两个字母，我们可以看到分别是C:36次，S:33次，他们可能对应的是e、t
3. 接着我们观察单词SIC，其在句子里出现的频率极高，推测其对应单词the
4. 我们观察的单词SINS，由于已知S和I，所以我们可以推断出这个单词为that，即N对应字母a
5. 我们观察单词NPC 则我们可以判断P对应字母r
6. 我们观察SM，很容易判断它对应单词to
7. 我们来查看VIM，他的对应的是who
8. 我们观察MF多次出现，判断其为of
9. 接着我们观察JB多次出现，同时也有JR出现，我们再观察频率，可以判断JB为in，JR为is
10. 我们观察FPMQ，很容易判断他为from
11. 我们观察XMJBS，很容易判断他为point

12. 我们将FPMQ N XMJBS N SM N XMJBS H翻译过来，from a point a to a point H,这样我们很容易判断H为b，同时后面的HY为by
13. 我们很容易判断XPMHACQ为problem
14. 观察ENJB，我们很容易判断段为gain
15. 观察GNB我们很容易判断其为can
16. 观察NBD，我们很容易判断为and
17. 观察PCGCJTCP，很容易判断为receiver
18. 观察RZGI，很容易判断为such
19. 最后我们观察ZBNZSIMPJOCD，很容易判断为unauthorized

3、置换密码表

最后得到的置换密码表如下表所示：

c	s	i	n	p	m	v	f	j	b	r	q	x	h	y	a	e	g	d	t	z	o
e	t	h	a	r	o	w	f	i	n	s	m	p	b	y	l	g	c	d	v	u	z

> 解密原文

于是我们便可根据以上关系进行解密，得到原文如下所示：

```
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.
```

四、总结与展望

1、总结

本次是密码学的第一次实验，在这次实验中对理论课上讲解的移位密码以及单表置换密码进行编程，使得对于这两种密码算法的原理和攻击方法更加的了解，也对密码学方面的编程更加的熟悉。

2、展望

在这次实验后，对密码学方面的知识更加的期待，也对这些原理和攻击方法更加的感兴趣，期待这学期自己更好的发展，万事胜意、心想事成。