1. Caesar

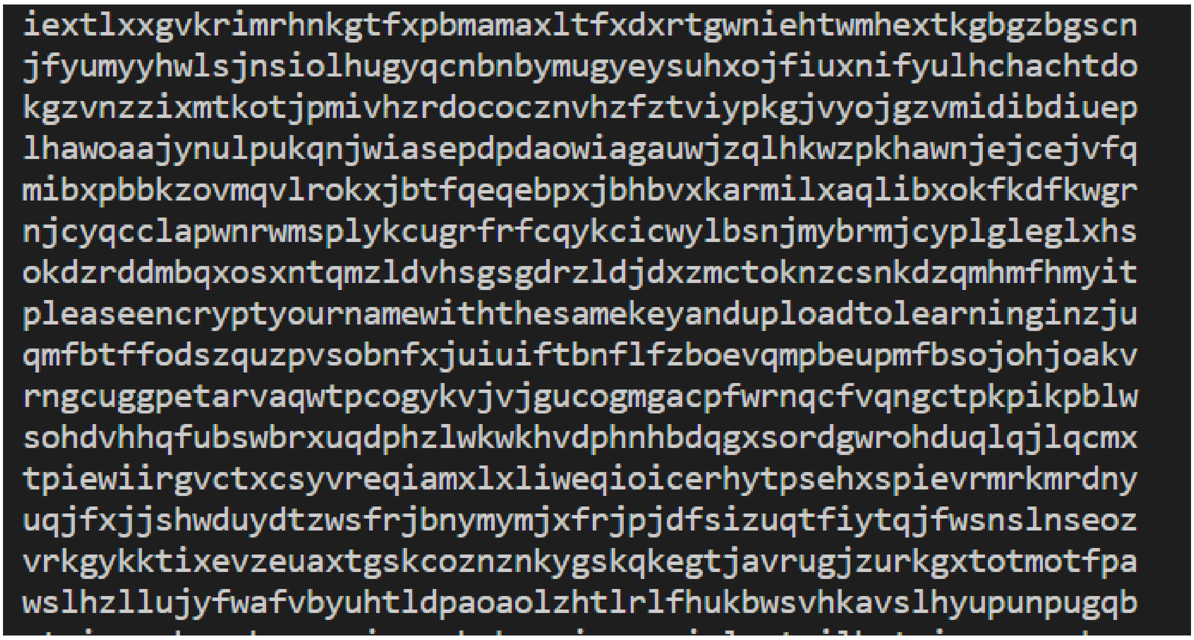
作业是破译这一段密码：`FBUQIUUDSHOFJOEKHDQCUMYJXJXUIQCUAUOQDTKFBEQTJEBUQHDYDWYDPZK`

因为是凯撒密码，所以我直接写了一个程序来对每一位移动一个字母，然后查看输出进行判断，程序如下：

因为大写不便于查看，我先全部转成了小写字母

```cpp
#include <bits/stdc++.h>
using namespace std;
int main()
{
    string str;
    cin >> str;
    for (int i = 0; i < str.size(); i++)
    {
        str[i] += 32;
    }
    for (int cnt = 0; cnt < 26; cnt++)
    {
        for (int i = 0; i < str.size(); i++)
        {
            str[i] += 1;
            if (str[i] == 'z' + 1)
                str[i] = 'a';
        }
        cout << str << endl;
    }
}
```

部分结果如下：



```
iextlxxgvkrimrhnkgtfxpbmamaxltfxdxrtgwniehtwmhextkgbgzbgscn
jfyumyyhwlsjnsiolhugyqcnbnbymugyeysuhxojfiuxnifyulhchachtdo
kgzvnzzixmtkotjpmivhzrdococznvhzfztviypkgjvyojgzvmidibdiuep
lhawoaajynulpukqnjwiasepdpdaowiagauwjzqlhkwzpkhawnjejcejvfq
mibxpbbkzovmqvlrokxjbtfqeqebpxjbhbvxkarmilxaqlibxokfkdfkwgr
njcyqcclapwnrwmsplykcugrfrfcqykcicwylbsnjmybrmjcyplgleglxhs
okdzrddmbqxosxntqmzldvhsgsgdrzldjdxzmctoknzcsnkdzqmhmfhmyit
pleaseencryptyournamewiththesamekeyanduploadtolearninginzju
qmfbtffodszquzpvsobnfxjuiuiftbnflfzboevqmpbeupmfbsojohjoakv
rngcuggpetarvaqwtpcogykvjvjgucogmgacpfwrnqcfvqngctpkpikpblw
sohdvhhqfubswbrxuqdphzlwkwkhvdphnhbdqgxsordgwrohduqlqjlqcmx
tpiewiirgvctxcsyvreqiamxlxliweqioicerhytpsehxspievrmrkmrdny
uqjfxjjshwduydtzwsfrjbnymymjxfrjpjdfsizuqtfiytqjfwsnslnseoz
vrkgykktixevzeuaxtgskcoznznkygskqkegtjavrugjzurkgxtotmotfpa
wslhzllujyfwafvbyuhtldpaoaolzhtlrlfhukbwsvhkavslhyupunpugqb
```

经过仔细地去查看结果（一点都不明显!）——查看的方式主要看前几个字母和后几个字母像不像单词，最后这几个zju有点像，发现P开头的这句话`pleaseencryptyournamewiththesamekeyanduploadtolearninginzju`很可能有含义，大概率是答案。于是断句进行翻译。

please encrypt your name with the same key and upload to learning in zju

请用相同密钥加密你的名字并且提交在学在浙大。本句bias为16

我的名字是`CAIJIAWEI` 加密后结果为`SQYZYQMUY`

2.Vignere

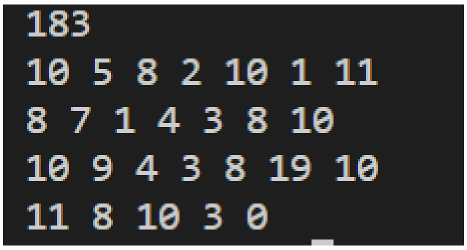
密码是

`ktbueluegvitnthuexmonveggmrcgxptlyhhjaogchoemqchpdnetxupbqntietiabpsmaoncnwvoutiugtagmmqsxtvxaoniiogtagmbpsmtuvvihpstpdvcrxhokvhxotawswquunewcgxptlcrxtevtubvewcnwwsxfsnptswtagakvoyyak`

这串密文有点长，所以我想优先靠字母出现频率判断，于是写一个程序记录各个字母出现的次数。

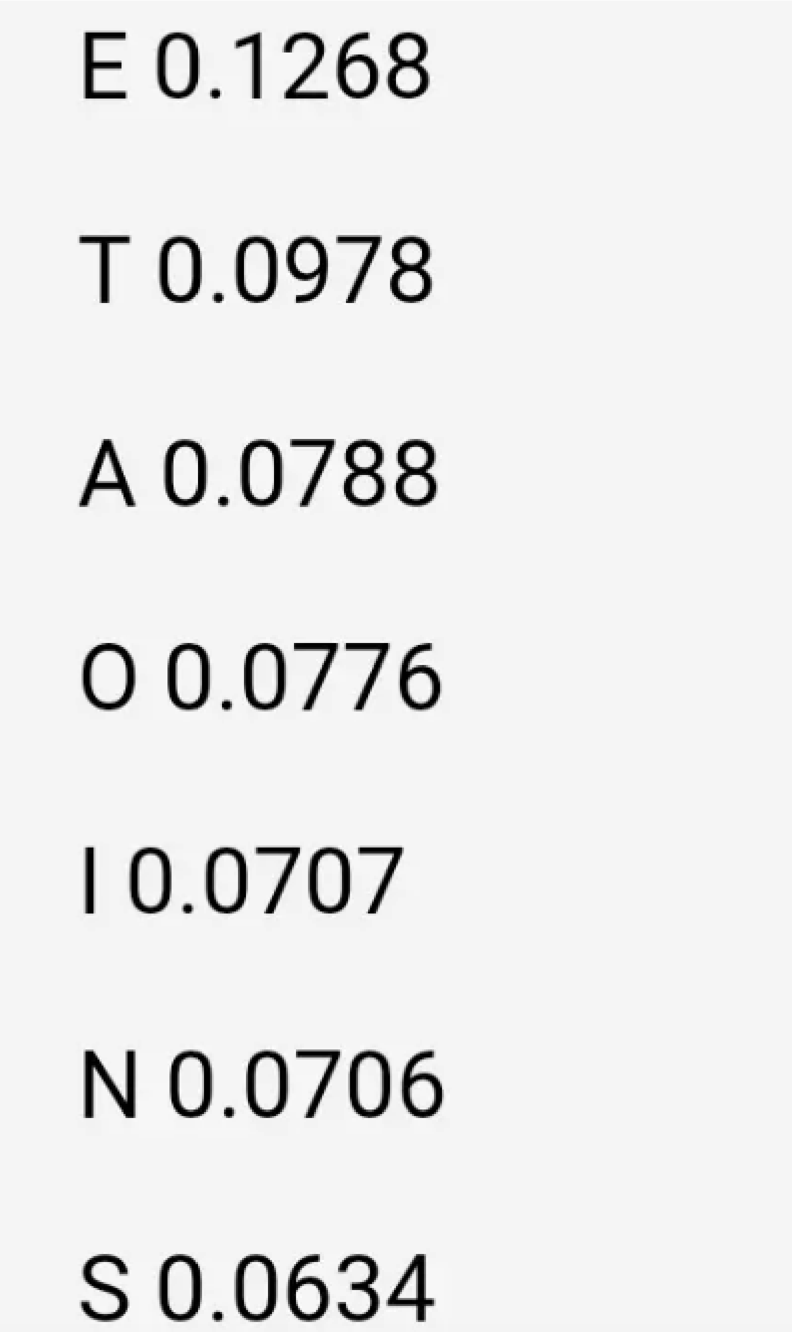
```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int keep[26] = {};
    string str;
    cin >> str;
    cout << str.size() << endl;
    for (int i = 0; i < str.size(); i++)
    {
        keep[str[i] - 'a']++;
    }
    for (int i = 0; i < 26; i++)
    {
        cout << keep[i] << " ";
        if ((i + 1) % 7 == 0)
            cout << "\n";
    }
}
```

结果如下：



```
183
10 5 8 2 10 1 11
8 7 1 4 3 8 10
10 9 4 3 8 19 10
11 8 10 3 0
```

其中最多的t字母出现了19次，比其他字母高几乎一倍，而我们查询字母频率表



E 0.1268

T 0.0978

A 0.0788

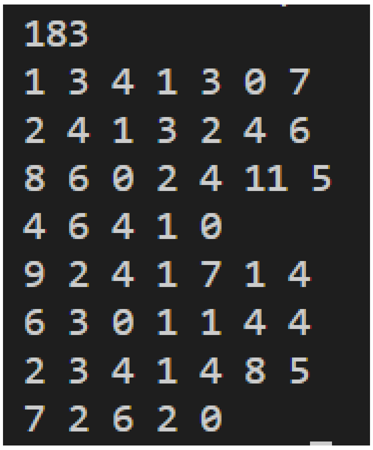
O 0.0776

I 0.0707

N 0.0706

S 0.0634

发现似乎频率过高了，再结合Vignere密码的机制，我转头尝试把单词分成两个一组，三个一组去进行频率的计次。



```
183
1 3 4 1 3 0 7
2 4 1 3 2 4 6
8 6 0 2 4 11 5
4 6 4 1 0
9 2 4 1 7 1 4
6 3 0 1 1 4 4
2 3 4 1 4 8 5
7 2 6 2 0
```

这是两个一组的统计频率结果，因为e的频率是高于其它字母不少的，所以我们假设e在上下两个情况下都有出现，即t/o/g代表e和a/t/e代表e，那我们进行这些假设：

上面t代表e，即往后推11位；上面o代表e，即往后推16位；上面g代表e，即往后推24位

下面a代表e，即往后推4位；下面t代表e，即往后推11位；不改显然不太可能

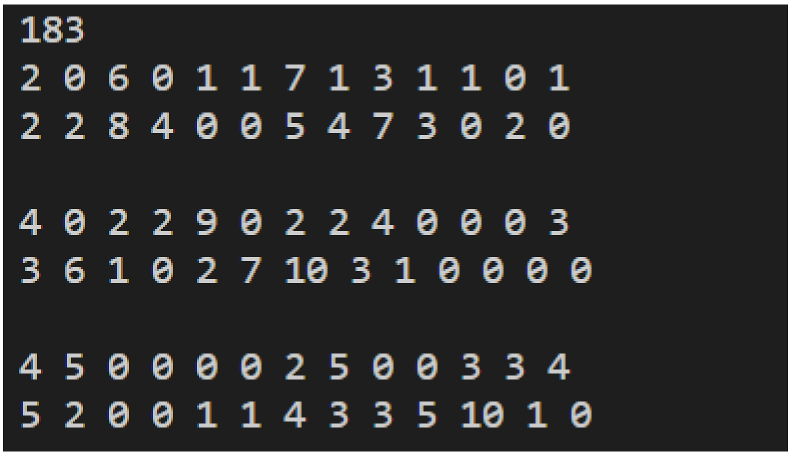
于是我们得到6（2*3）串结果

```
vxmyppfirztxyxsypbxsyzpkrqcgrbaxwcsluezknlzixunlahyiebftmuyxtiemlfawxezrnrhzzyem
fkeerqxudbeziezrtmzkeerqmtdqeygztlawetoznvilzogliseehwhufyyihgrbaxwgcbeigxffgihg
yahwijdraxdaeerevzzcjev

vemfpwfprgteyesfpixzygprrxcnriaewjssulzrnszpxbnsaoypeifambyetpetlmadxlzynyhgzfet
frelrxxbdiegilzyttzrelrxmadxefggtsadeaognciszvgsizelhdhbffyphnriaewnciepgefmgphn
yhhdiqdyaedhelrlvgzjjlv

axryupkiwzyxdxxyubcsdzukwqhgwbfxbcxlzeeksleicuslfhdijbktrudxyijmqffwceersrmzeyjm
kkjewqcuibjzneerymekjewqrtiqjylzylfwjttzsvnleollnsjemwmukydimgwbfxbghbjilxkflimg
damwnjirfxiajeweazecoea

aerfuwkpwgyedexfuiczdgurwxhnwifebjxszlerssepcbssfodpjikarbdeypjtqmfdcleysymgefjt
krjlwxcbiijgnleyyterjlwxraixjflgysfdjatgscnsevlsnzjlmdmbkfdpmnwifebnhijplekmlpmn
dhmdnqiyfeihjlwlagejola

ixzycpsiezgxlxfycbkslzckeqpgebnxjcflhemkalmikualnhlirbstzulxgirmyfnwkemraruzmyrm
skreeqkuqbrzvemrgmmkreeqztqqrytzglnwrtbzavvlmotlvsreuwuusyliugebnxjgpbritxsftiug
lauwvjqrnxqareeeizmcwei

iezfcwspeggeleffcikzlgcrexpneinejjfshlmrasmpkbasnolprisazblegprtymndklmyayugmfrt
srrlexkbqirgvlmygtmrrlexzaqxrftggsndrabgacvsmvtsvzrludubsflpuneinejnpirptesmtpun
lhudvqqyneqhrleligmjwli
```

这很显然都是天书，所以2个一组不对。我们改成三个一组，然后e在频率最高的两个字母之一。代码如下：

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int keep1[78] = {};
    string str;
    cin >> str;
    cout << str.size() << endl;
    for (int i = 0; i < str.size(); i++)
    {
        keep1[str[i] - 'a' + (i % 3) * 26]++;
    }
    for (int i = 0; i < 78; i++)
    {
        cout << keep1[i] << " ";
        if ((i + 1) % 13 == 0)
            cout << "\n";
        if ((i + 1) % 26 == 0)
            cout << "\n";
    }
}
```

结果如下：



```
183
2 0 6 0 1 1 7 1 3 1 1 0 1
2 2 8 4 0 0 5 4 7 3 0 2 0

4 0 2 2 9 0 2 2 4 0 0 0 3
3 6 1 0 2 7 10 3 1 0 0 0 0

4 5 0 0 0 0 2 5 0 0 3 3 4
5 2 0 0 1 1 4 3 3 5 10 1 0
```

所以我们假设：

第一排：p/g代表e，密钥为推15个或推24个

第二排：t/e代表e，密钥为推11个或一个也不推

第三排：x/b代表e，密钥为推7个或推3个

即有8（$2*2*2$）种情况：

答案分别是：

```
zeijpsjpnktaceojpebzukpnvxyrreeesnsoylvvnodptfnoeouteejaifyaxpaxliedtpzurydkzbit
bvehvxtfdeigepzuxtvvehvxiedtifcktoedaeocrcewzrksedehlddffbcpdrreeesrceipcifikpdr
ydldeudueezlehvlrkzfnlr

zeejpojpjktwcekjpabzqkpjvxurraeeonskylrvnkdppfnkeoqteajaefywxpwxleedppzqryzkzxit
xvedvxpfdaigapzqxtrvedvxeedpifyktkedweoyrcawznksadedldzffxcpzrraeeorcaipyifekpzr
yzldaudqeevledvlnkzbnln

ztijesjenkiactojeeboukenvmyrgeetsnhoyavvcodetfcoeduttejpifnaxeaxaiestpourndkobii
bvthvmtfseivepouxivvthvmiestiuckioesaedcrreworkhedthlsdfubcedrgeetsrreieciuikedr
ndlseusuetzlthvarkofnar

ztejeojejkiwctkjeaboqkejvmurgaetonhkyarvckdepfckedqttajpefnwxewxaeesppoqrnzkoxii
xvtdvmpfsaivapoqxirvtdvmeespiuykikeswedyrrawonkhadtdlszfuxcezrgaetorraieyiuekezr
nzlsausqetvltdvankobnan

ieispsspnttaleospekzutpnexyareneswsohlvenomptononouceesaioyagpaglindtyzuaydtzbrt
beehextodergeyzugtveehexindtrfcttondanocacefzrtsemehuddofblpdarenesacerpcrfitpda
ydudeddunezuehelrtzfwlr

ieesposppjttwlekspakzqtpjexuaraneowskhlrenkmpponknoqceasaeoywgpwglendpyzqayztzxrt
xeedexpodargayzqgtreedexendprfyttkndwnoyacafzntsamedudzofxlpzaraneoacarpyrfetpza
yzudaddqnevuedelntzbwln

itisessentialtoseekoutenemyagentswhohavecometoconductespionageagainstyouandtobri
bethemtoserveyougivetheminstructionsandcareforthemthusdoubledagentsarerecruiteda
ndusedsuntzutheartofwar

iteseosejtiwltkseakoqtejemuagantowhkhareckmepockndqctaspeonwgewgaenspyoqanztoxri
xetdemposarvayoqgiretdemenspruytiknswndyarafonthamtdusozouxlezagantoarareyrueteza
nzusadsqntvutdeantobwan
```

这下倒数第二排明显是一句话

其中字母的下标%3==0的话，字母的bias为24；

%3==1的话，字母的bias为0，即没变；

%3==2的话，字母的bias为19。

这里可以发现不能臆测不变，说明上面"不改显然不太可能"的论断是错的，下次需要注意不能主观猜测，否则可能解不出密文。

这句话是：

It is essential to seek out enemy agents who have come to conduct espionage against you and to bribe them to serve you, give them instructions and care for them, thus doubled agents are recruited and used, Sun Tzu, The Art of War.

孙子在《孙子兵法》中说过：必须找出前来对你进行间谍活动的敌特，贿赂他们为你服务，给他们指示和照顾他们，从而招募和使用双重间谍。这应该是反间计的一句话。因此成功解决问题

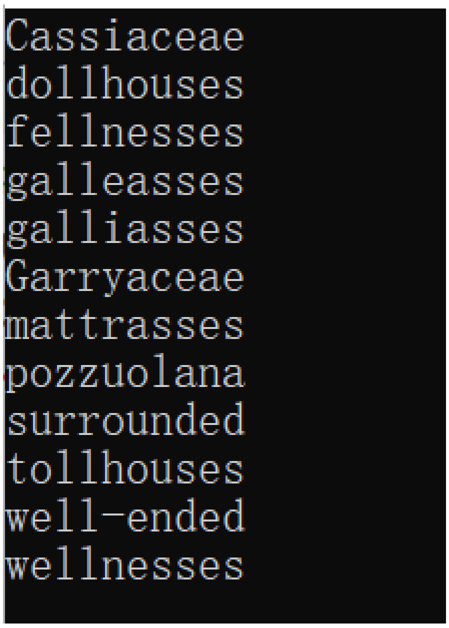
3.unknown

密码是`MAL TIRRUEZF CR MAL RKZYIOL EX MAL OIY UAE RICF “MAL ACWALRM DYEUPLFWL CR ME DYEU MAIM UL IZL RKZZEKYFLF GH OHRMLZH`

我注意到MAL出现了很多次，这很有可能是使用了子替换。我又接着注意到`RKZZEKYFLF`这个奇怪的单词，它的第三个和第四个字母都是Z，第二个和第六个字母都是K，第八个和第十个字母都是F。

然后我使用如下程序在所有单词里面寻找满足要求的单词。

```cpp
#include <bits/stdc++.h>
using namespace std;
int main()
{
    ifstream myfile;
    myfile.open("words.txt");
    if (!myfile.is_open())
    {
        cout << "no" << endl;
    }
    string str;
    // myfile >> str;
    // cout << str << endl;
    while (myfile >> str)
    {
        if (str.size() == 10 && (str[2] == str[3]) && (str[1] == str[5]) && 
(str[7] == str[9]))
        {
            cout << str << endl;
        }
    }
    myfile.close();
    return 0;
}
```



```
Cassiaceae
dollhouses
fellnesses
galleasses
galliasses
Garryaceae
mattrasses
pozzuolana
surrounded
tollhouses
well-ended
wellnesses
```

其中满足的单词如上所示，高度怀疑是surrounded，因为别的单词都不如它常见。而surrounded后面这个两个字母的单词，怀疑是by，因为surrounded by是固定搭配。

然后可以构建出如下的字母对应表格

原本的字母	**加密后结果**
S	R
U	K
R	Z
O	E
N	Y
D	F
E	L
B	G
Y	H

带着这张表去看密文，第三个单词的第二个字母是S，大概率是is，还有这个MAL，最后一个字母是E，前面是be动词的情况下不能是are，那怀疑是the，非常合理，倒数第五个单词，最后一个字母是E，后面跟are surrounded by，前面是主语，we应运而生。然后加长这个表

I	**C**
T	M
H	A
W	U

然后继续破解，DYEU这个单词是？NOW，那必然怀疑是know，DYEUPLFWL是know？ed？e，knowledge基本石锤，继续加长

K	**D**
L	P
G	W

照着这个规律继续破解，除了之前错判了of以为是on，但发现N的密文早就找到了，看错了之后，就逐渐解出全部密文，完整表格如下：

原本的字母	加密后结果
S	R
U	K
R	Z
O	E
N	Y
D	F
E	L
B	G
Y	H
I	C
T	M
H	A
W	U
K	D
L	P
G	W
M	O
P	T
F	X

自此找到了所有密文中密码的原文，翻译过来则是：

The password is the surname of the man who said "the highest knowledge is to know that we are surrounded by mystery."

密码是说“最高深的知识是知道我们被神秘所包围”的那个人的姓氏。