INFORMATION SECURITY AND CRYPTOGRAPHY

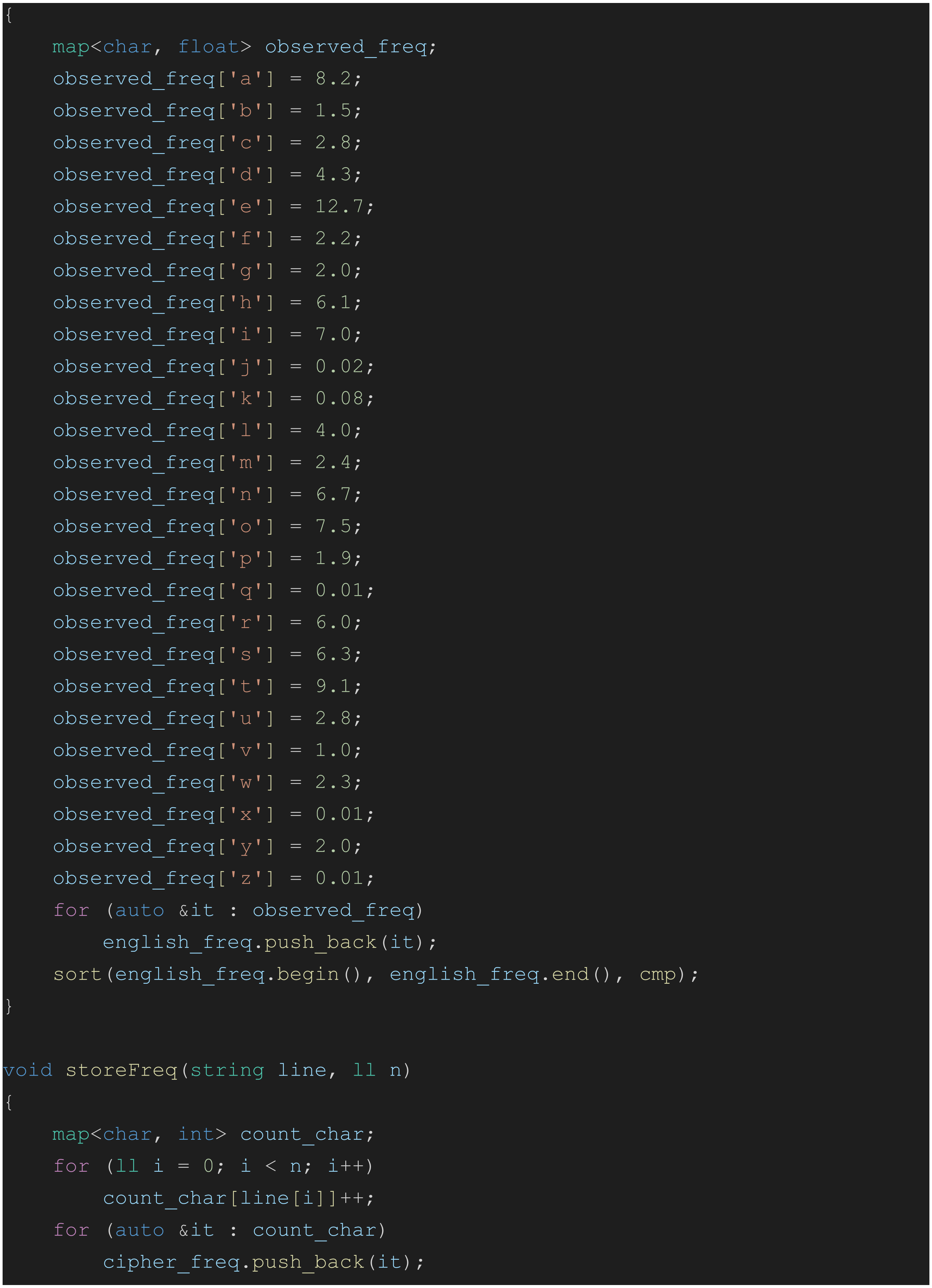
Assignment: 1

U20CS005
BANSI MARAKANA

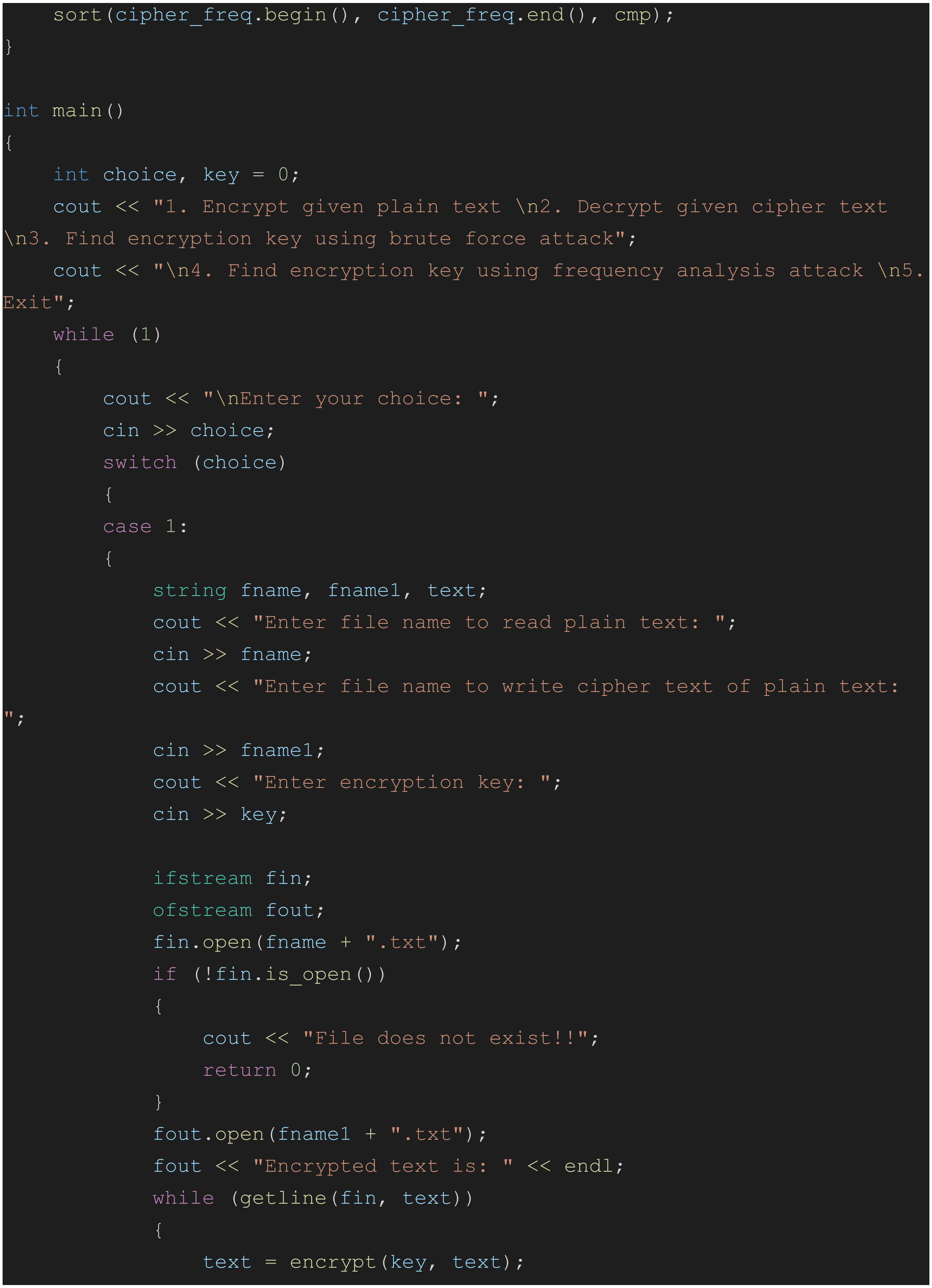
1. Implement a menu driven program for Caesar Cipher with following functions.
a. Encrypt given plain text.
b. Decrypt given ciphertext.
c. Find encryption key using brute force attack.
d.Find encryption key using frequency analysis attack.
Note: Consider file as an input in the program.

PROGRAM:

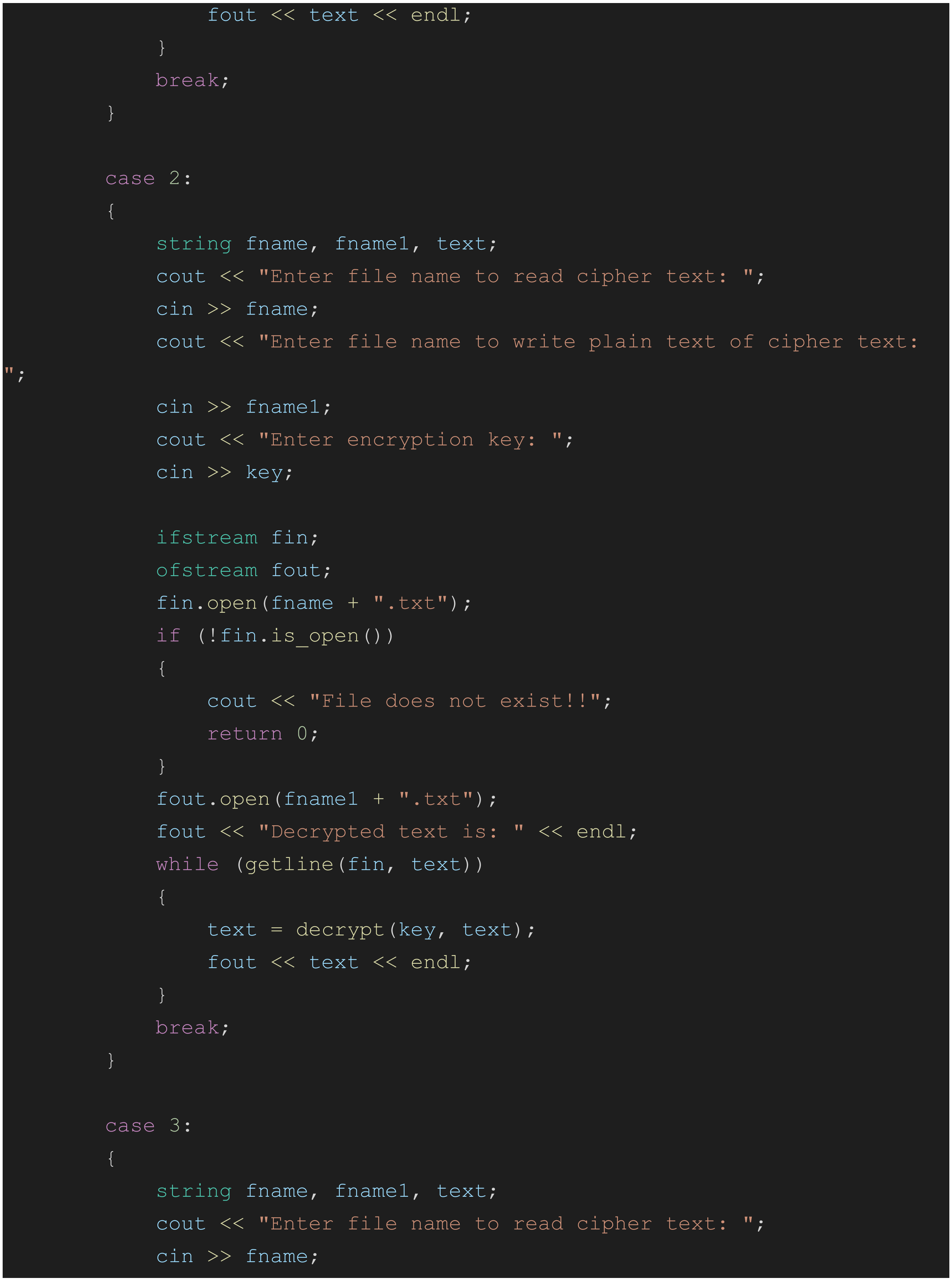
```
#include <bits/stdc++.h>
#include <iostream>
#include <fstream>
using namespace std;
typedef long long ll;

string encrypt(int key, string text)
{
    for (ll i = 0; i < text.length(); i++)
        text[i] = char(text[i] + key - 'a') % 26 + 'a';
    return text;
}

string decrypt(int key, string text)
{
    for (ll i = 0; i < text.length(); i++)
        text[i] = char(text[i] - key - 'a' + 26) % 26 + 'a';
    return text;
}

vector<pair<char, float>> english_freq;
vector<pair<char, float>> cipher_freq;
bool cmp(pair<char, float> &a, pair<char, float> &b)
{
    return a.second > b.second;
}

void initMostFreqCount()
```



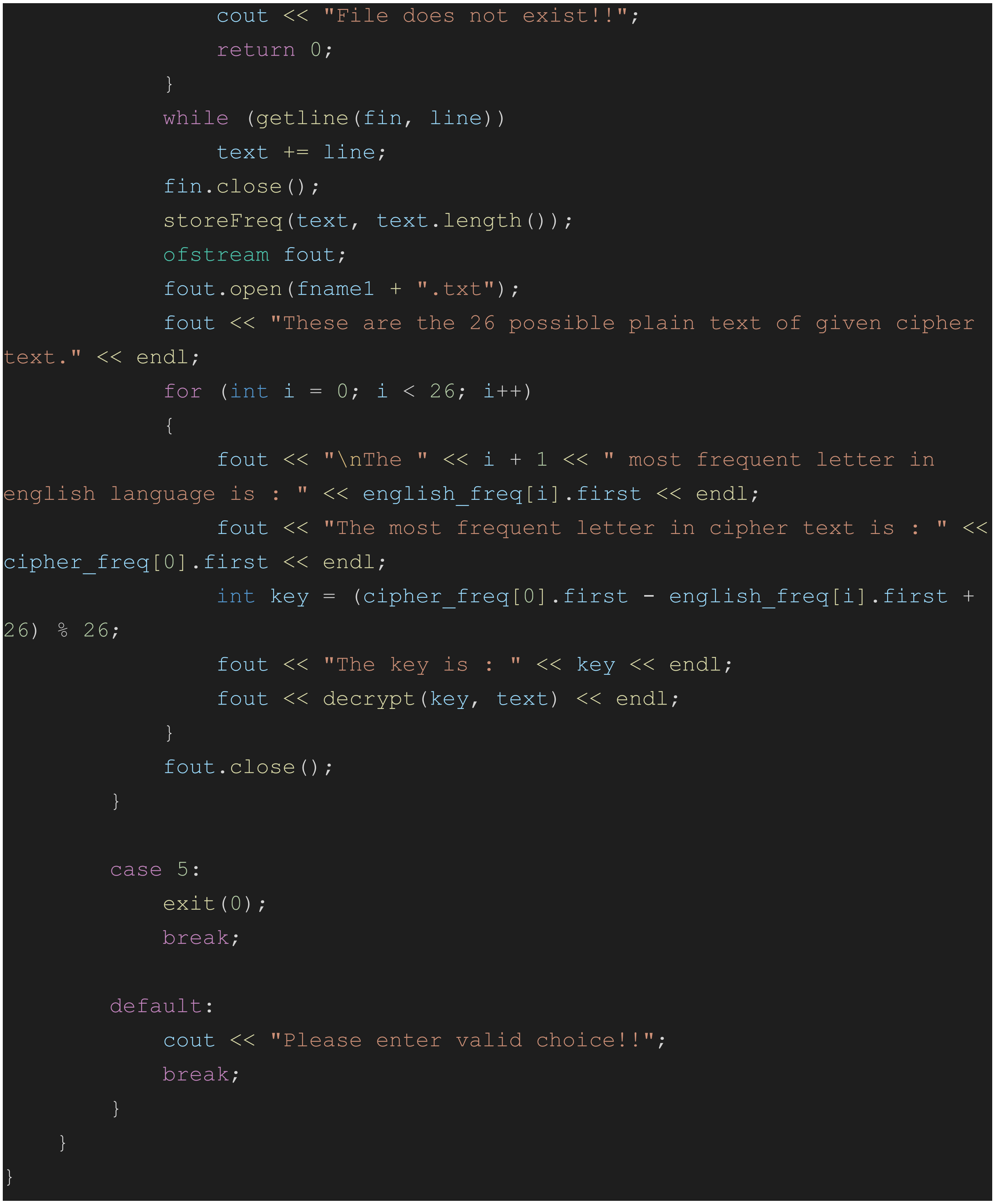
```
{
    map<char, float> observed_freq;
    observed_freq['a'] = 8.2;
    observed_freq['b'] = 1.5;
    observed_freq['c'] = 2.8;
    observed_freq['d'] = 4.3;
    observed_freq['e'] = 12.7;
    observed_freq['f'] = 2.2;
    observed_freq['g'] = 2.0;
    observed_freq['h'] = 6.1;
    observed_freq['i'] = 7.0;
    observed_freq['j'] = 0.02;
    observed_freq['k'] = 0.08;
    observed_freq['l'] = 4.0;
    observed_freq['m'] = 2.4;
    observed_freq['n'] = 6.7;
    observed_freq['o'] = 7.5;
    observed_freq['p'] = 1.9;
    observed_freq['q'] = 0.01;
    observed_freq['r'] = 6.0;
    observed_freq['s'] = 6.3;
    observed_freq['t'] = 9.1;
    observed_freq['u'] = 2.8;
    observed_freq['v'] = 1.0;
    observed_freq['w'] = 2.3;
    observed_freq['x'] = 0.01;
    observed_freq['y'] = 2.0;
    observed_freq['z'] = 0.01;
    for (auto &it : observed_freq)
        english_freq.push_back(it);
    sort(english_freq.begin(), english_freq.end(), cmp);
}

void storeFreq(string line, ll n)
{
    map<char, int> count_char;
    for (ll i = 0; i < n; i++)
        count_char[line[i]]++;
    for (auto &it : count_char)
        cipher_freq.push_back(it);
```



```
    sort(cipher_freq.begin(), cipher_freq.end(), cmp);
}

int main()
{
    int choice, key = 0;
    cout << "1. Encrypt given plain text \n2. Decrypt given cipher text \n3. Find encryption key using brute force attack";
    cout << "\n4. Find encryption key using frequency analysis attack \n5. Exit";
    while (1)
    {
        cout << "\nEnter your choice: ";
        cin >> choice;
        switch (choice)
        {
        case 1:
        {
            string fname, fname1, text;
            cout << "Enter file name to read plain text: ";
            cin >> fname;
            cout << "Enter file name to write cipher text of plain text: ";
            cin >> fname1;
            cout << "Enter encryption key: ";
            cin >> key;

            ifstream fin;
            ofstream fout;
            fin.open(fname + ".txt");
            if (!fin.is_open())
            {
                cout << "File does not exist!!";
                return 0;
            }
            fout.open(fname1 + ".txt");
            fout << "Encrypted text is: " << endl;
            while (getline(fin, text))
            {
                text = encrypt(key, text);
```



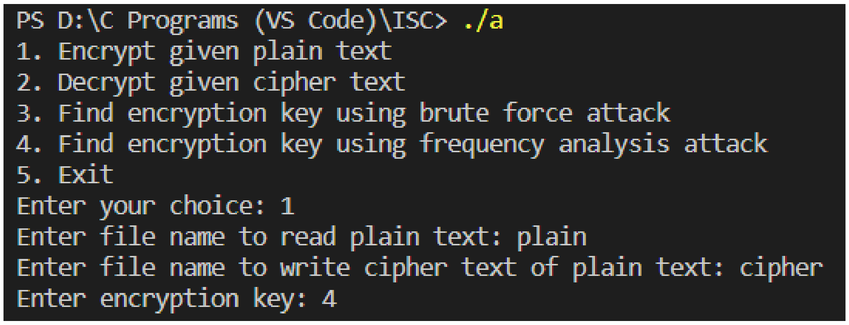
```
                fout << text << endl;
            }
            break;
        }

        case 2:
        {
            string fname, fname1, text;
            cout << "Enter file name to read cipher text: ";
            cin >> fname;
            cout << "Enter file name to write plain text of cipher text:
";

            cin >> fname1;
            cout << "Enter encryption key: ";
            cin >> key;

            ifstream fin;
            ofstream fout;
            fin.open(fname + ".txt");
            if (!fin.is_open())
            {
                cout << "File does not exist!!";
                return 0;
            }
            fout.open(fname1 + ".txt");
            fout << "Decrypted text is: " << endl;
            while (getline(fin, text))
            {
                text = decrypt(key, text);
                fout << text << endl;
            }
            break;
        }

        case 3:
        {
            string fname, fname1, text;
            cout << "Enter file name to read cipher text: ";
            cin >> fname;
            cout << "Enter file name to write all possible plain text of
cipher text: ";
            cin >> fname1;

            ifstream fin;
            ofstream fout;
            fout.open(fname1 + ".txt");
            fout << "These are the 26 possible plain text of given cipher
text." << endl;
            fout.close();
            for (int i = 0; i < 26; i++)
            {
                fin.open(fname + ".txt");
                fout.open(fname1 + ".txt", ios::app);
                fout << endl << "For key = " << i << endl;
                while (getline(fin, text))
                {
                    text = decrypt(i, text);
                    fout << text << endl;
                }
                fin.close();
                fout.close();
            }
            break;
        }

        case 4:
        {
            initMostFreqCount();
            string fname, fname1, text, line;
            cout << "Enter file name to read cipher text: ";
            cin >> fname;
            cout << "Enter file name to write all possible plain text of
cipher text: ";
            cin >> fname1;

            ifstream fin;
            fin.open(fname + ".txt");
            if (!fin.is_open())
            {
```



```
                cout << "File does not exist!!";
                return 0;
            }
            while (getline(fin, line))
                text += line;
            fin.close();
            storeFreq(text, text.length());
            ofstream fout;
            fout.open(fname1 + ".txt");
            fout << "These are the 26 possible plain text of given cipher
text." << endl;
            for (int i = 0; i < 26; i++)
            {
                fout << "\nThe " << i + 1 << " most frequent letter in
english language is : " << english_freq[i].first << endl;
                fout << "The most frequent letter in cipher text is : " <<
cipher_freq[0].first << endl;
                int key = (cipher_freq[0].first - english_freq[i].first +
26) % 26;
                fout << "The key is : " << key << endl;
                fout << decrypt(key, text) << endl;
            }
            fout.close();
        }

        case 5:
            exit(0);
            break;

        default:
            cout << "Please enter valid choice!!";
            break;
        }
    }
}
```

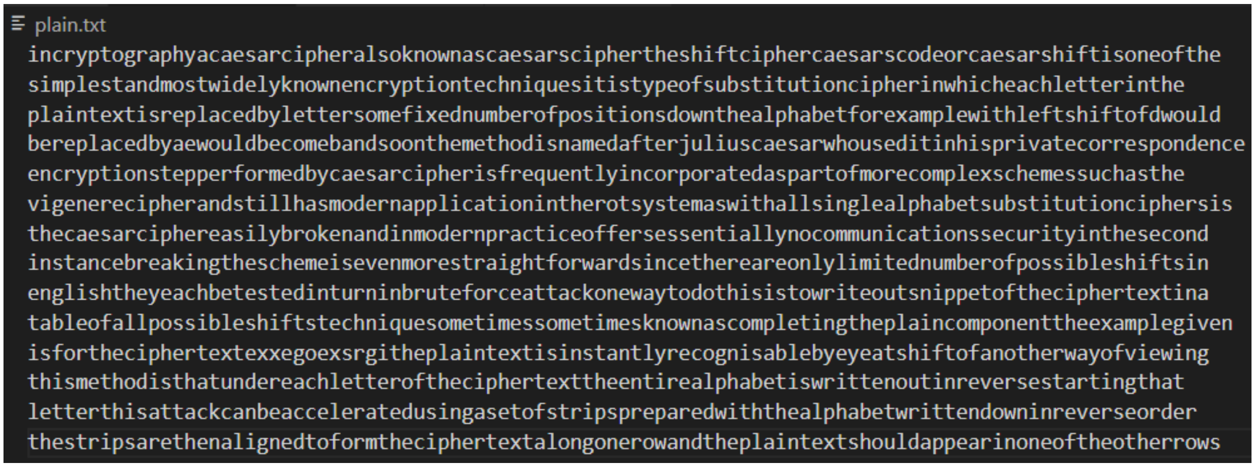
OUTPUT:

a. Encryption:



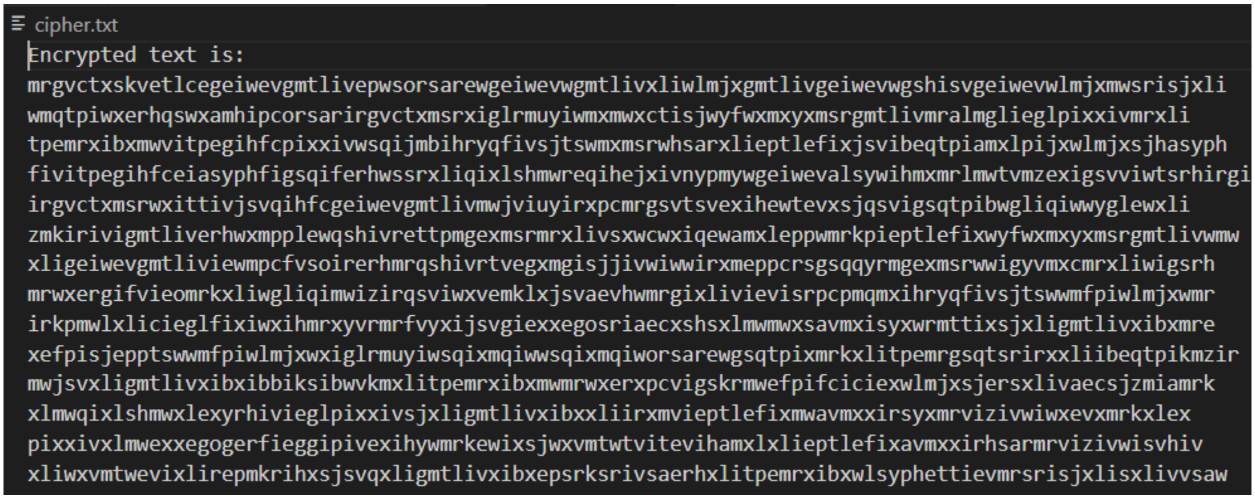
```
PS D:\C Programs (VS Code)\ISC> ./a
1. Encrypt given plain text
2. Decrypt given cipher text
3. Find encryption key using brute force attack
4. Find encryption key using frequency analysis attack
5. Exit
Enter your choice: 1
Enter file name to read plain text: plain
Enter file name to write cipher text of plain text: cipher
Enter encryption key: 4
```

Plain.txt:



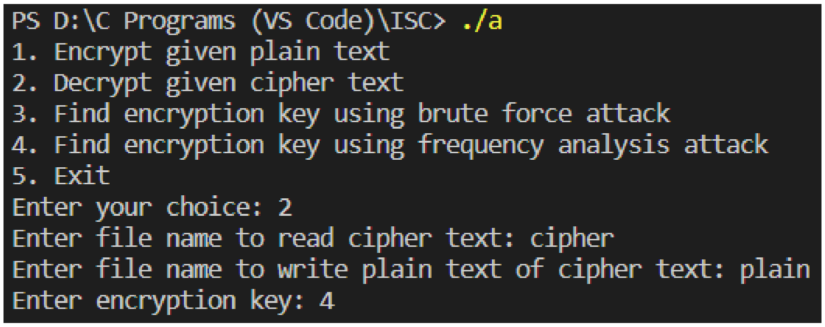
```
incryptographyacaesarcipheralsoknownascaesarsciphertheshiftciphercaesarscodeorcaesarshiftisoneofthe
simplestandmostwidelyknownencryptiontechniquesitistypeofsubstitutioncipherinwhicheachletterinthe
plaintextisreplacedbylettersomefixednumberofpositionsdownthealphabetforexamplewithleftshiftofdwould
bereplacedbyaewouldbecomebandsoonthemethodisnamedafterjuliuscaesarwhouseditinhisprivatecorrespondence
encryptionstepperformedbycaesarcipherisfrequentlyincorporatedaspartofmorecomplexschemessuchasthe
vigenerecipherandstillhasmodernapplicationintherotsystemaswithallsinglealphabetsubstitutionciphersis
thecaesarciphereasilybrokenandinmodernpracticeoffersessentiallynocommunicationssecurityinthesecond
instancebreakingtheschemeisevenmorestraightforwardsincethereareonlylimitednumberofpossibleshiftsin
englishtheyeachbetestedinturninbruteforceattackonewaytodothisistowriteoutsnippetoftheciphertextina
tableofallpossibleshiftstechniquesometimessometimesknownascompletingtheplaincomponenttheexamplegiven
isfortheciphertextexxegoexsrgitheplaintextisinstantlyrecognisablebyeyeatshiftofanotherwayofviewing
thismethodisthatundereachletteroftheciphertexttheentirealphabetiswrittenoutinreversestartingthat
letterthisattackcanbeacceleratedusingasetofstripspreparedwiththealphabetwrittendowninreverseorder
thestripsarethenalignedtoformtheciphertextalongonerowandtheplaintextshouldappearinoneoftheotherrows
```

Cipher.txt



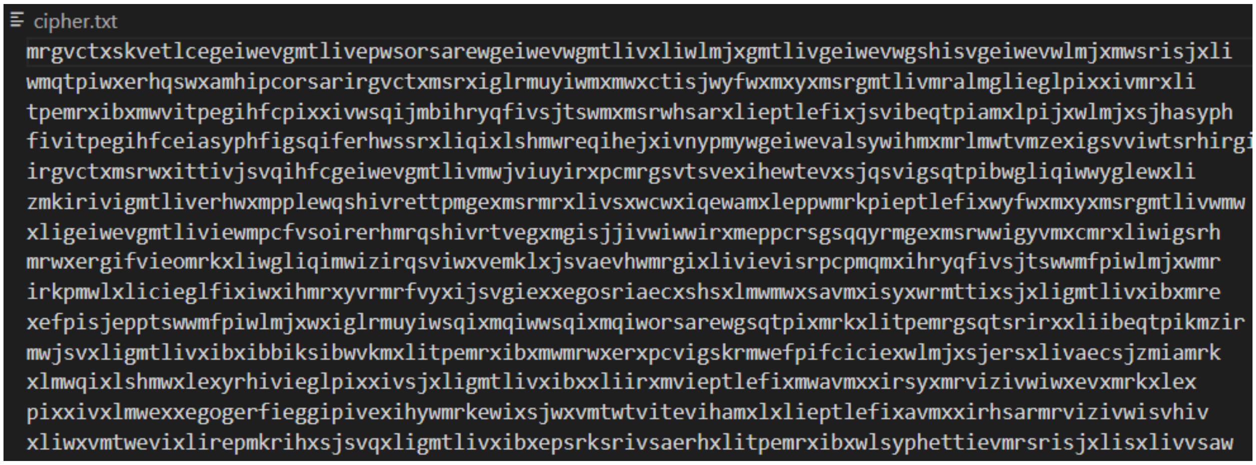
```
Encrypted text is:
mrgvctxskvetlcegeiwevgmtlivepwsorsarewgeiwevwgmtlivxliwlmjxgmtlivgeiwevwgshisvgeiwevwlmjxmwsrisjxli
wmqtpiwxerhqswxamhipcorsarirgvctxmsrxiglrmuyiwmxmwxctisjwyfwxmxyxmsrgmtlivmralmglieglpixxivmrxli
tpemrxibxmwvitpegihfcpixxivwsqijmbihryqfivsjtswmxmsrwhsarxlieptlefixjsvibeqtpiamxlpijxwlmjxsjhasyph
fivitpegihfceiasyphfigsqiferhwssrxliqixlshmwreqihejxivnypmywgeiwevalsywihmxmrlmwtvmzexigsvviwtsrhirgi
irgvctxmsrwxittivjsvqihfcgeiwevgmtlivmwjviuyirxpcmrgsvtsvexihewtevxsjqsvigsqtpibwgliqiwwyglewxli
zmkirivigmtliverhwxmpplewqshivrettpmgexmsrmrxlivsxwcwxiqewamxleppwmrkpieptlefixwyfwxmxyxmsrgmtlivwmw
xligeiwevgmtliviewmpcfvsoirerhmrqshivrtvegxmgisjjivwiwwirxmeppcrsgsqqyrmgexmsrwwigyvmxcmrxliwigsrh
mrwxergifvieomrkxliwgliqimwizirqsviwxvemklxjsvaevhwmrgixlivievisrpcpmqmxihryqfivsjtswwmfpiwlmjxwmr
irkpmwlxlicieglfixiwxihmrxyvrmrfvyxijsvgiexxegosriaecxshsxlmwmwxsavmxisyxwrmttixsjxligmtlivxibxmre
xefpisjepptswwmfpiwlmjxwxiglrmuyiwsqixmqiwwsqixmqiworsarewgsqtpixmrkxlitpemrgsqtsrirxxliibeqtpikmzir
mwjsvxligmtlivxibxibbiksibwvkmxlitpemrxibxmwmrwxerxpcvigskrmwefpifciciexwlmjxsjersxlivaecsjzmiamrk
xlmwqixlshmwxlexyrhivieglpixxivsjxligmtlivxibxxliirxmvieptlefixmwavmxxirsyxmrvizivwiwxevxmrkxlex
pixxivxlmwexxegogerfieggipivexihywmrkewixsjwxvmtwtvitevihamxlxlieptlefixavmxxirhsarmrvizivwisvhiv
xliwxvmtwevixlirepmkrihxsjsvqxligmtlivxibxepsrksrivsaerhxlitpemrxibxwlsyphettievmrsrisjxlisxlivvsaw
```

b. Decryption:



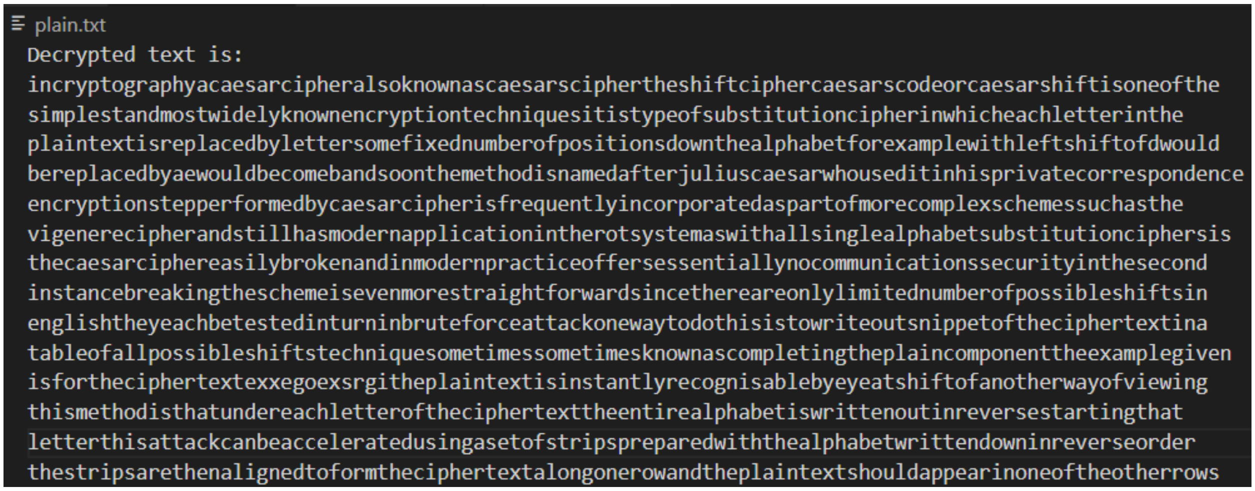
```
PS D:\C Programs (VS Code)\ISC> ./a
1. Encrypt given plain text
2. Decrypt given cipher text
3. Find encryption key using brute force attack
4. Find encryption key using frequency analysis attack
5. Exit
Enter your choice: 2
Enter file name to read cipher text: cipher
Enter file name to write plain text of cipher text: plain
Enter encryption key: 4
```

Cipher.txt



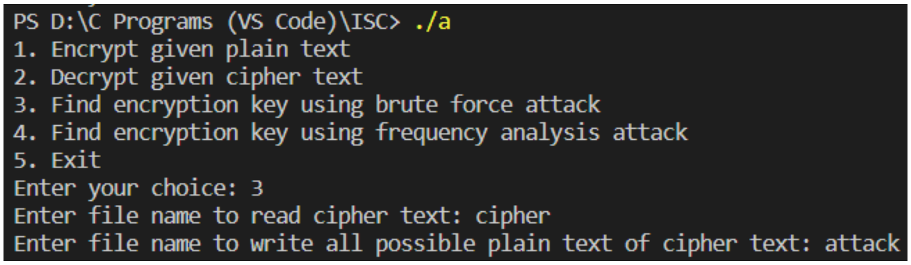
```
mrgvctxskvetlcegeiwevgmtlivepwsorsarewgeiwevwgmtlivxliwlmjxgmtlivgeiwevwgshisvgeiwevwlmjxmwsrisjxli
wmqtpiwxerhqswxamhipcorsarirgvctxmsrxiglrmuyiwmxmwxctisjwyfwxmxyxmsrgmtlivmralmglieglpixxivmrxli
tpemrxibxmwvitpegihfcpixxivwsqijmbihryqfivsjtswmxmsrwhsarxlieptlefixjsvibeqtpiamxlpijxwlmjxsjhasyph
fivitpegihfceiasyphfigsqiferhwssrxliqixlshmwreqihejxivnypmywgeiwevalsywihmxmrlmwtvmzexigsvviwtsrhirgi
irgvctxmsrwxittivjsvqihfcgeiwevgmtlivmwjviuyirxpcmrgsvtsvexihewtevxsjqsvigsqtpibwgliqiwwyglewxli
zmkirivigmtliverhwxmpplewqshivrettpmgexmsrmrxlivsxwcwxiqewamxleppwmrkpieptlefixwyfwxmxyxmsrgmtlivwmw
xligeiwevgmtliviewmpcfvsoirerhmrqshivrtvegxmgisjjivwiwwirxmeppcrsgsqqyrmgexmsrwwigyvmxcmrxliwigsrh
mrwxergifvieomrkxliwgliqimwizirqsviwxvemklxjsvaevhwmrgixlivievisrpcpmqmxihryqfivsjtswwmfpiwlmjxwmr
irkpmwlxlicieglfixiwxihmrxyvrmrfvyxijsvgiexxegosriaecxshsxlmwmwxsavmxisyxwrmttixsjxligmtlivxibxmre
xefpisjepptswwmfpiwlmjxwxiglrmuyiwsqixmqiwwsqixmqiworsarewgsqtpixmrkxlitpemrgsqtsrirxxliibeqtpikmzir
mwjsvxligmtlivxibxibbiksibwvkmxlitpemrxibxmwmrwxerxpcvigskrmwefpifciciexwlmjxsjersxlivaecsjzmiamrk
xlmwqixlshmwxlexyrhivieglpixxivsjxligmtlivxibxxliirxmvieptlefixmwavmxxirsyxmrvizivwiwxevxmrkxlex
pixxivxlmwexxegogerfieggipivexihywmrkewixsjwxvmtwtvitevihamxlxlieptlefixavmxxirhsarmrvizivwisvhiv
xliwxvmtwevixlirepmkrihxsjsvqxligmtlivxibxepsrksrivsaerhxlitpemrxibxwlsyphettievmrsrisjxlisxlivvsaw
```

Plain.txt:



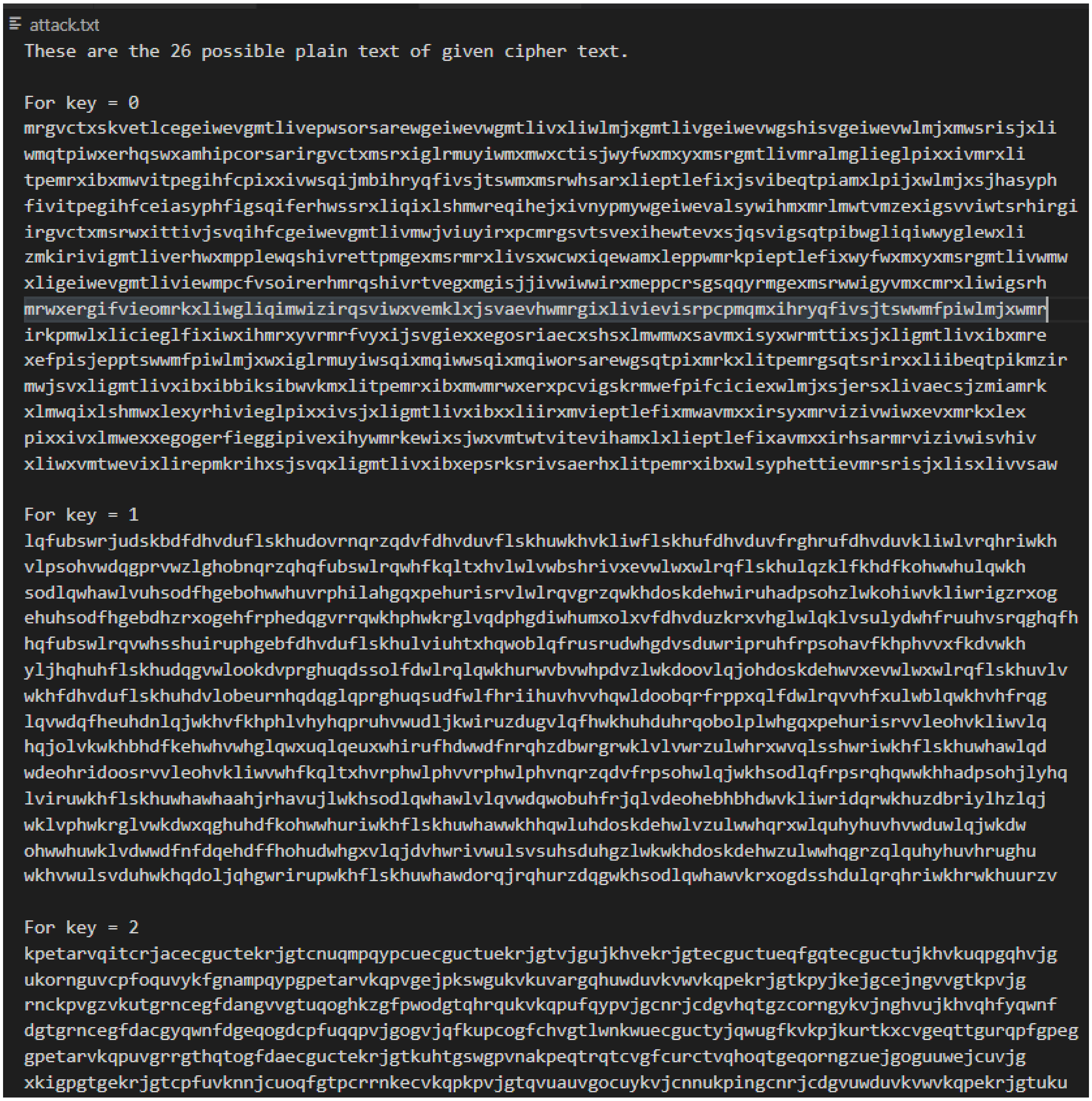
```
Decrypted text is:
incryptographyacaesarcipheralsoknownascaesarsciphertheshiftciphercaesarscodeorcaesarshiftisoneofthe
simplestandmostwidelyknownencryptiontechniquesitistypeofsubstitutioncipherinwhicheachletterinthe
plaintextisreplacedbylettersomefixednumberofpositionsdownthealphabetforexamplewithleftshiftofdwould
bereplacedbyaewouldbecomebandsoonthemethodisnamedafterjuliuscaesarwhouseditinhisprivatecorrespondence
encryptionstepperformedbycaesarcipherisfrequentlyincorporatedaspartofmorecomplexschemessuchasthe
vigenerecipherandstillhasmodernapplicationintherotsystemaswithallsinglealphabetsubstitutionciphersis
thecaesarciphereasilybrokenandinmodernpracticeoffersessentiallynocommunicationssecurityinthesecond
instancebreakingtheschemeisevenmorestraightforwardsincethereareonlylimitednumberofpossibleshiftsin
englishtheyeachbetestedinturninbruteforceattackonewaytodothisistowriteoutsnippetoftheciphertextina
tableofallpossibleshiftstechniquesometimessometimesknownascompletingtheplaincomponenttheexamplegiven
isfortheciphertextexxegoexsrgitheplaintextisinstantlyrecognisablebyeyeatshiftofanotherwayofviewing
thismethodisthatundereachletteroftheciphertexttheentirealphabetiswrittenoutinreversestartingthat
letterthisattackcanbeacceleratedusingasetofstripspreparedwiththealphabetwrittendowninreverseorder
thestripsarethenalignedtoformtheciphertextalongonerowandtheplaintextshouldappearinoneoftheotherrows
```

c. Brute force attack:



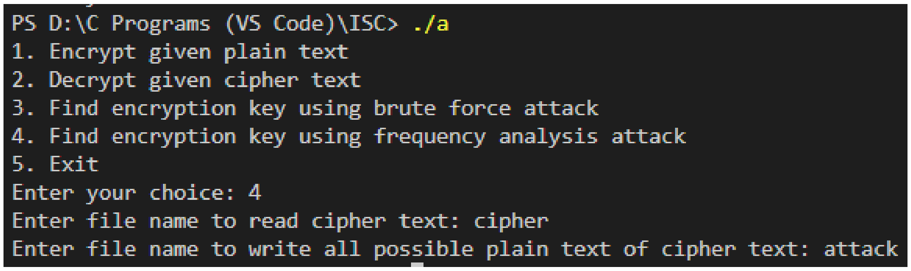
```
PS D:\C Programs (VS Code)\ISC> ./a
1. Encrypt given plain text
2. Decrypt given cipher text
3. Find encryption key using brute force attack
4. Find encryption key using frequency analysis attack
5. Exit
Enter your choice: 3
Enter file name to read cipher text: cipher
Enter file name to write all possible plain text of cipher text: attack
```

attack.txt



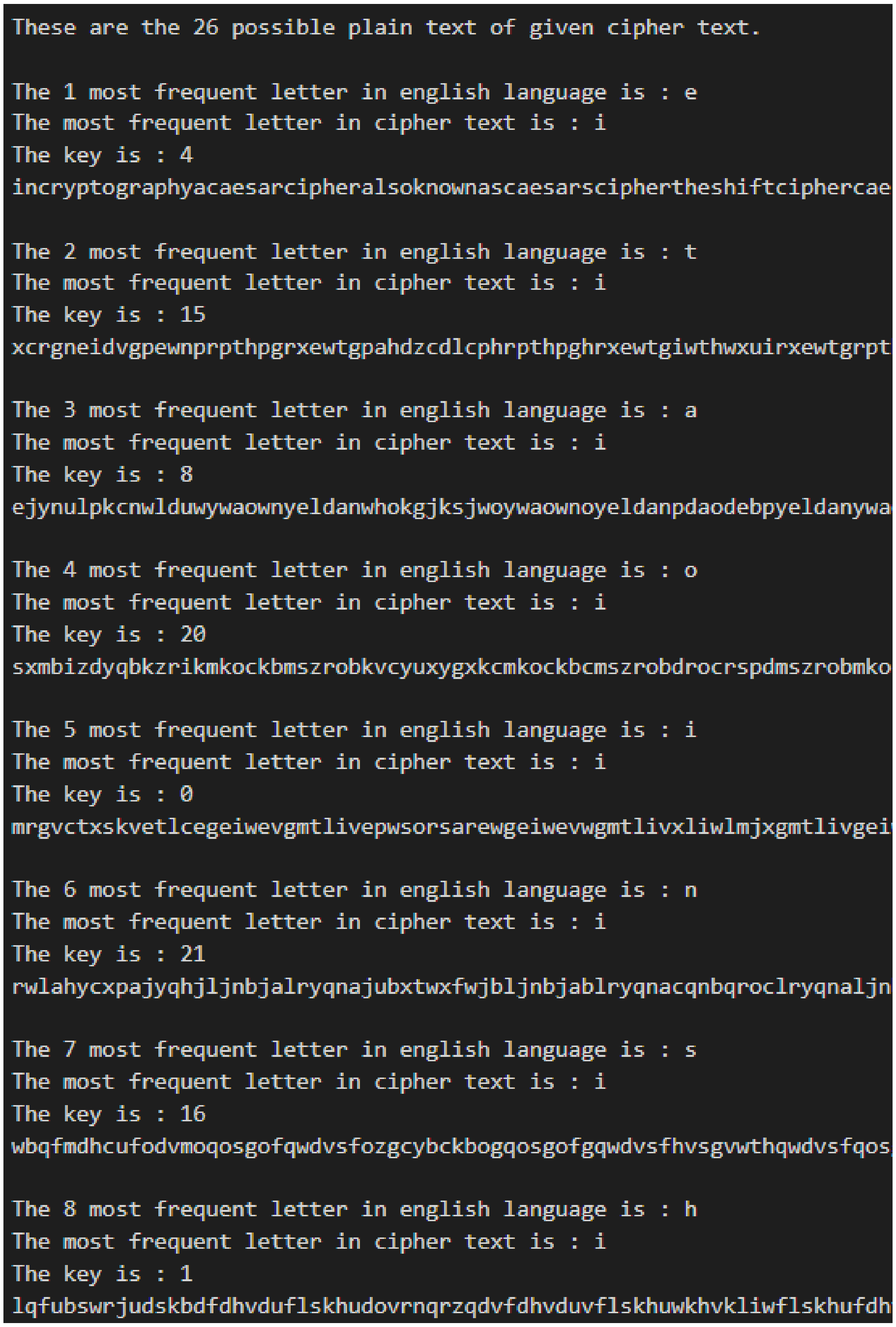
```
These are the 26 possible plain text of given cipher text.

For key = 0
mrgvctxskvetlcegeiwevgmtlivepwsorsarewgeiwevwgmtlivxliwlmjxgmtlivgeiwevwgshisvgeiwevwlmjxmwsrisjxli
wmqtpiwxerhqswxamhipcorsarirgvctxmsrxiglrmuyiwmxmwxctisjwyfwxmxyxmsrgmtlivmralmglieglpixxivmrxli
tpemrxibxmwvitpegihfcpixxivwsqijmbihryqfivsjtswmxmsrwhsarxlieptlefixjsvibeqtpiamxlpijxwlmjxsjhasyph
fivitpegihfceiasyphfigsqiferhwssrxliqixlshmwreqihejxivnypmywgeiwevalsywihmxmrlmwtvmzexigsvviwtsrhirgi
irgvctxmsrwxittivjsvqihfcgeiwevgmtlivmwjviuyirxpcmrgsvtsvexihewtevxsjqsvigsqtpibwgliqiwwyglewxli
zmkirivigmtliverhwxmpplewqshivrettpmgexmsrmrxlivsxwcwxiqewamxleppwmrkpieptlefixwyfwxmxyxmsrgmtlivwmw
xligeiwevgmtliviewmpcfvsoirerhmrqshivrtvegxmgisjjivwiwwirxmeppcrsgsqqyrmgexmsrwwigyvmxcmrxliwigsrh
mrwxergifvieomrkxliwgliqimwizirqsviwxvemklxjsvaevhwmrgixlivievisrpcpmqmxihryqfivsjtswwmfpiwlmjxwmr
irkpmwlxlicieglfixiwxihmrxyvrmrfvyxijsvgiexxegosriaecxshsxlmwmwxsavmxisyxwrmttixsjxligmtlivxibxmre
xefpisjepptswwmfpiwlmjxwxiglrmuyiwsqixmqiwwsqixmqiworsarewgsqtpixmrkxlitpemrgsqtsrirxxliibeqtpikmzir
mwjsvxligmtlivxibxibbiksibwvkmxlitpemrxibxmwmrwxerxpcvigskrmwefpifciciexwlmjxsjersxlivaecsjzmiamrk
xlmwqixlshmwxlexyrhivieglpixxivsjxligmtlivxibxxliirxmvieptlefixmwavmxxirsyxmrvizivwiwxevxmrkxlex
pixxivxlmwexxegogerfieggipivexihywmrkewixsjwxvmtwtvitevihamxlxlieptlefixavmxxirhsarmrvizivwisvhiv
xliwxvmtwevixlirepmkrihxsjsvqxligmtlivxibxepsrksrivsaerhxlitpemrxibxwlsyphettievmrsrisjxlisxlivvsaw

For key = 1
lqfubswrjudskbdfdhvduflskhudovrnqrzqdvfdhvduvflskhuwkhvkliwflskhufdhvduvfrghrufdhvduvkliwlvrqhriwkh
vlpsohvwdqgprvwzlghobnqrzqhqfubswlrqwhfkqltxhvlwlvwbshrivxevwlwxwlrqflskhulqzklfkhdfkohwwhulqwkh
sodlqwhawlvuhsodfhgebohwwhuvrphilahgqxpehurisrvlwlrqvgrzqwkhdoskdehwiruhadpsohzlwkohiwvkliwrigzrxog
ehuhsodfhgebdhzrxogehfrphedqgvrrqwkhphwkrglvqdphgdiwhumxolxvfdhvduzkrxvhglwlqklvsulydwhfruuhvsrqghqfh
hqfubswlrqvwhsshuiruphgebfdhvduflskhulviuhtxhqwoblqfrusrudwhgdvsduwripruhfrpsohavfkhphvvxfkdvwkh
yljhqhuhflskhudqgvwlookdvprghuqdssolfdwlrqlqwkhurwvbvwhpdvzlwkdoovlqjohdoskdehwvxevwlwxwlrqflskhuvlv
wkhfdhvduflskhuhdvlobeurnhqdqglqprghuqsudfwlfhriihuvhvvhqwldoobqrfrppxqlfdwlrqvvhfxulwblqwkhvhfrqg
lqvwdqfheuhdnlqjwkhvfkhphlvhyhqpruhvwudljkwiruzdugvlqfhwkhuhduhrqobolplwhgqxpehurisrvvleohvkliwvlq
hqjolvkwkhbhdfkehwhvwhglqwxuqlqeuxwhirufhdwwdfnrqhzdbwrgrwklvlvwrzulwhrxwvqlsshwriwkhflskhuwhawlqd
wdeohridoosrvvleohvkliwvwhfkqltxhvrphwlphvvrphwlphvnqrzqdvfrpsohwlqjwkhsodlqfrpsrqhqwwkhhadpsohjlyhq
lviruwkhflskhuwhawhaahjrhavujlwkhsodlqwhawlvlqvwdqwobuhfrjqlvdeohebhbhdwvkliwridqrwkhuzdbriylhzlqj
wklvphwkrglvwkdwxqghuhdfkohwwhuriwkhflskhuwhawwkhhqwluhdoskdehwlvzulwwhqrxwlquhyhuvhvwduwlqjwkdw
ohwwhuwklvdwwdfnfdqehdffhohudwhgxvlqjdvhwrivwulsvsuhsduhgzlwkwkhdoskdehwzulwwhqgrzqlquhyhuvhrughu
wkhvwulsvduhwkhqdoljqhgwrirupwkhflskhuwhawdorqjrqhurzdqgwkhsodlqwhawvkrxogdsshdulqrqhriwkhrwkhuurzv

For key = 2
kpetarvqitcrjacecguctekrjgtcnuqmpqypcuecguctuekrjgtvjgujkhvekrjgtecguctueqfgqtecguctujkhvkuqpgqhvjg
ukornguvcpfoquvykfgnampqypgpetarvkqpvgejpkswgukvkuvargqhuwduvkvwvkqpekrjgtkpyjkejgcejngvvgtkpvjg
rnckpvgzvkutgrncegfdangvvgtuqoghkzgfpwodgtqhrqukvkqpufqypvjgcnrjcdgvhqtgzcorngykvjnghvujkhvqhfyqwnf
dgtgrncegfdacgyqwnfdgeqogdcpfuqqpvjgogvjqfkupcogfchvgtlwnkwuecguctyjqwugfkvkpjkurtkxcvgeqttgurqpfgpeg
gpetarvkqpuvgrrgthqtogfdaecguctekrjgtkuhtgswgpvnakpeqtrqtcvgfcurctvqhoqtgeqorngzuejgoguuwejcuvjg
xkigpgtgekrjgtcpfuvknnjcuoqfgtpcrrnkecvkqpkpvjgtqvuauvgocuykvjcnnukpingcnrjcdgvuwduvkvwvkqpekrjgtuku
```

d. Frequency Analysis attack:



```
PS D:\C Programs (VS Code)\ISC> ./a
1. Encrypt given plain text
2. Decrypt given cipher text
3. Find encryption key using brute force attack
4. Find encryption key using frequency analysis attack
5. Exit
Enter your choice: 4
Enter file name to read cipher text: cipher
Enter file name to write all possible plain text of cipher text: attack
```

Attack.txt:



```
These are the 26 possible plain text of given cipher text.

The 1 most frequent letter in english language is : e
The most frequent letter in cipher text is : i
The key is : 4
incryptographyacaesarcipheralsoknownascaesarsciphertheshiftciphercae

The 2 most frequent letter in english language is : t
The most frequent letter in cipher text is : i
The key is : 15
xcrgneidvgpewnprpthpgrxewtgpahdzcdlcphrpthpghrxewtgiwthwxuirxewtgrpt

The 3 most frequent letter in english language is : a
The most frequent letter in cipher text is : i
The key is : 8
ejynulpkcnwlduwywaownyeldanwhokgjksjwoywaownoyeldanpdaodebpyeldanywa

The 4 most frequent letter in english language is : o
The most frequent letter in cipher text is : i
The key is : 20
sxmbizdyqbkzrikmkockbmszrobkvcyuxygxkcmkockbcmszrobdrocrspdmszrobmko

The 5 most frequent letter in english language is : i
The most frequent letter in cipher text is : i
The key is : 0
mrgvctxskvetlcegeiwevgmtlivepwsorsarewgeiwevwgmtlivxliwlmjxgmtlivgei

The 6 most frequent letter in english language is : n
The most frequent letter in cipher text is : i
The key is : 21
rwlahycxpajyqhjljnbjalryqnajubxtwxfwjbljnbjablryqnacqnbqroclryqnaljn

The 7 most frequent letter in english language is : s
The most frequent letter in cipher text is : i
The key is : 16
wbqfmdhcufodvmoqosgofqwdvsfozgcybckbogqosgofgqwdvsfhvsgvwthqwdvsfqos

The 8 most frequent letter in english language is : h
The most frequent letter in cipher text is : i
The key is : 1
lqfubswrjudskbdfdhvduflskhudovrnqrzqdvfdhvduvflskhuwkhvkliwflskhufdh
```