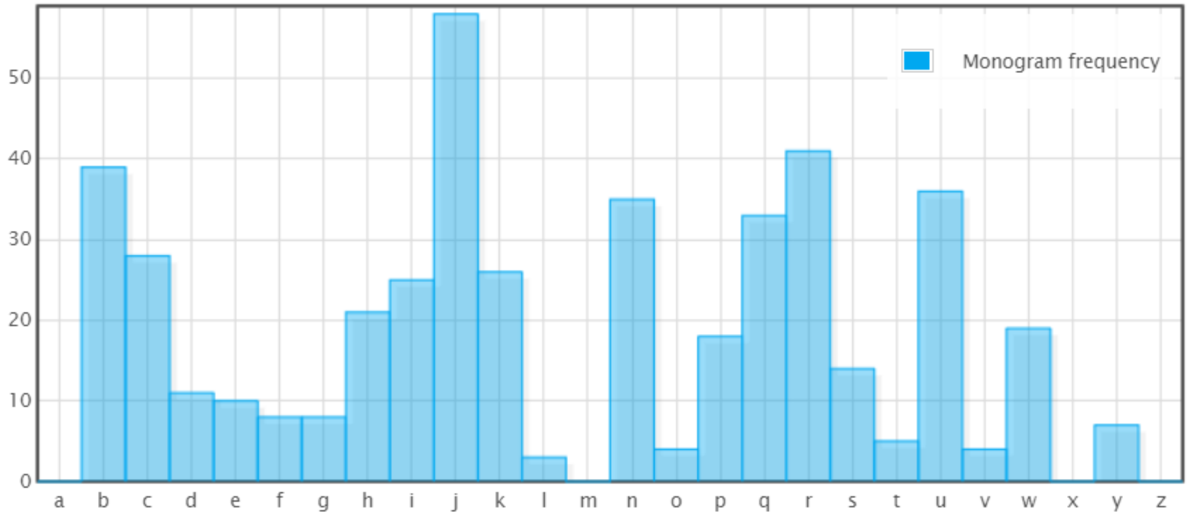
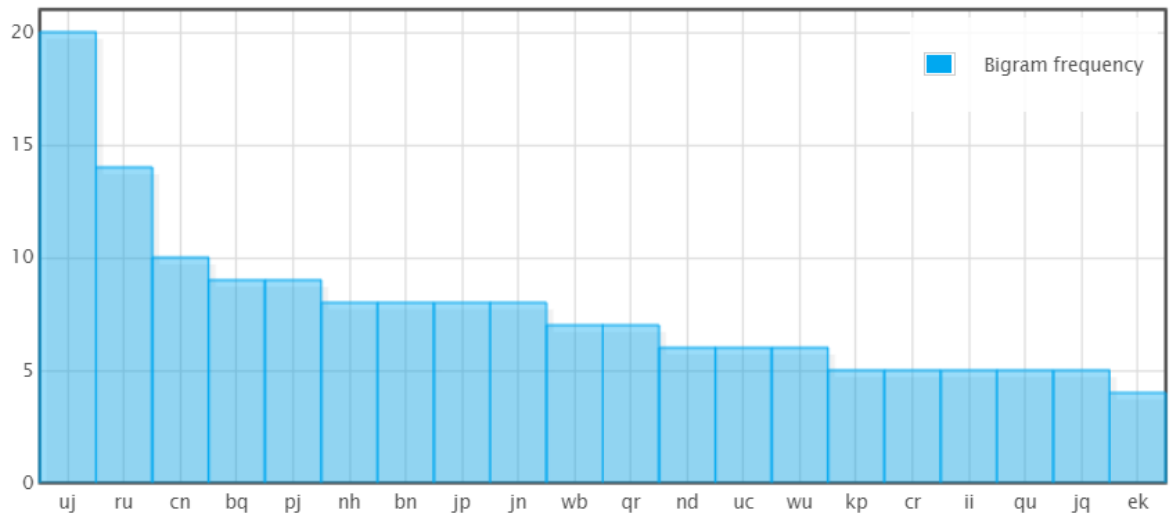
Ciphertext encrypt by Monoalphabetic cipher

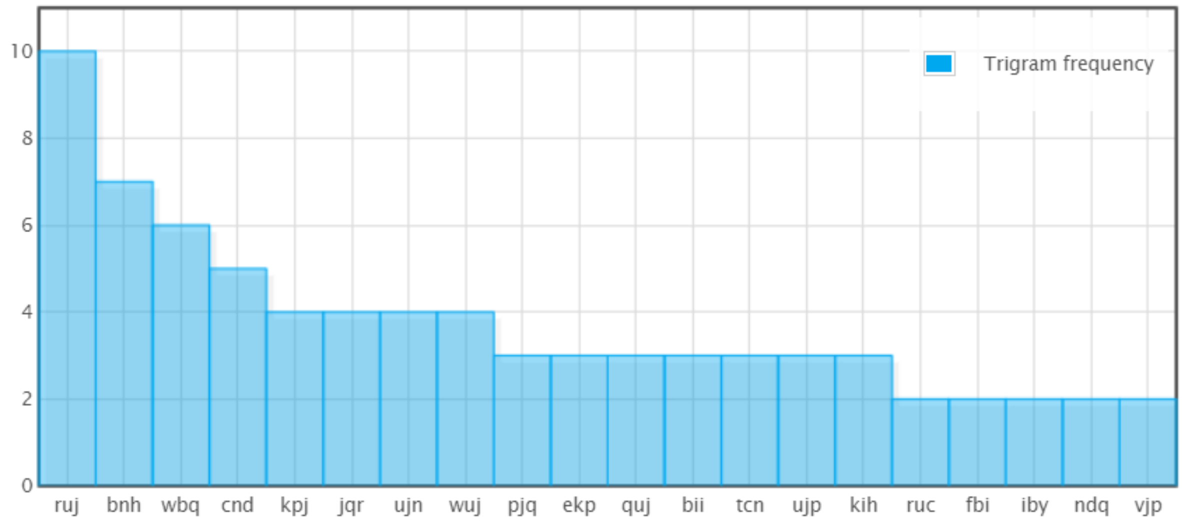
cn kihjn rcojq wujn wcqucnd qrcii ujiljh knj rujpj icvjh b tcnd wukqj hbsdurjpq wjpj bii fjbsrcesi fsr ruj yksndjqr wbq qk fjbsrcesi rubr ruj qsn crqjie wucgu ubq qjjn qk osgu wbq bqrkncqujh wujnjvjp cr quknj cn ujp ebgj gikqj fy ruj tcndq gbqrij iby b dpjbr hbpt ekpjqr bnh snhjp bn kih icojrpjj cn ruj ekpjqr wbq b wjii bnh wujn ruj hby wbq vjpy wbpo ruj tcndq gucih wjnr ksr cnrk ruj ekpjqr bnh qbr hkwn fy ruj qchj ke ruj gkki eksnrbcn bnh wujn quj wbq fkpjh quj rkkt b

dkihjn fbii bnh rupjw cr sl kn ucdu bnh gbsdur cr bnh rucq fbii wbq ujp ebvkpcrj libyrucnd



A	B	C	D	E	F	G	H	I	J
NULL	39	28	11	10	8	8	21	25	58
K	L	M	N	O	P	Q	R	S	T
26	3	NULL	35	4	18	33	41	14	5
U	V	W	X	Y	Z				
36	4	19	NULL	7	NULL				

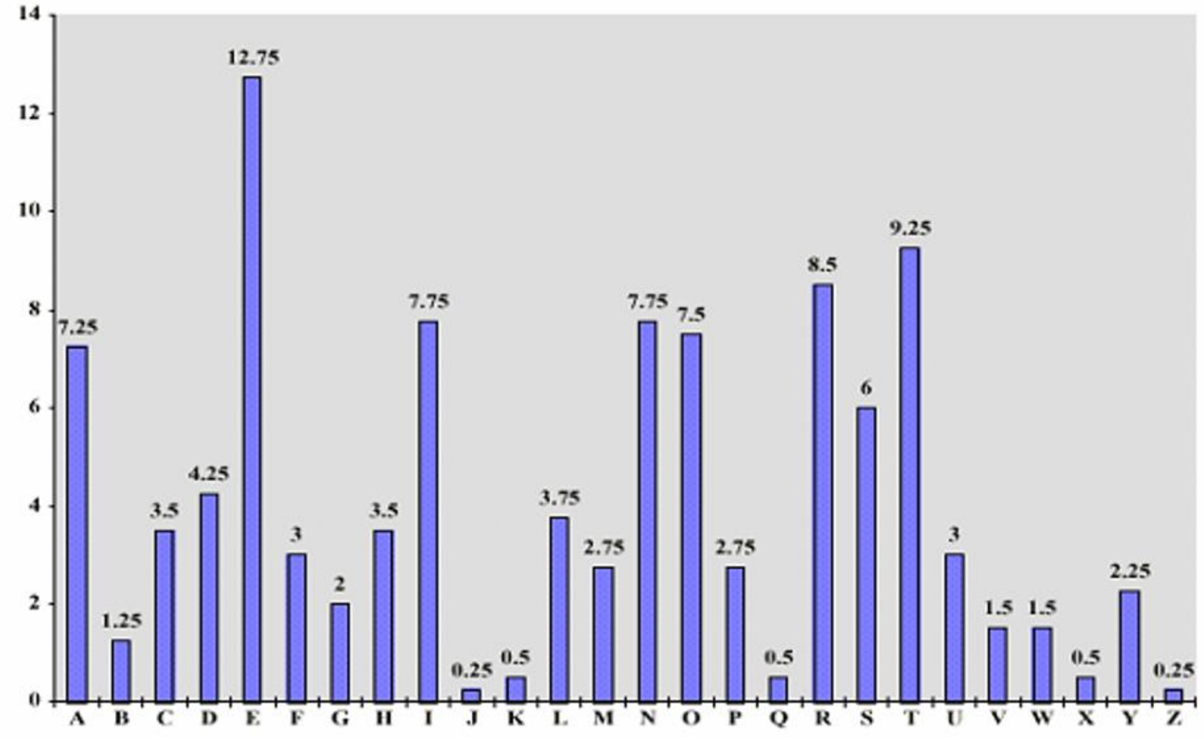


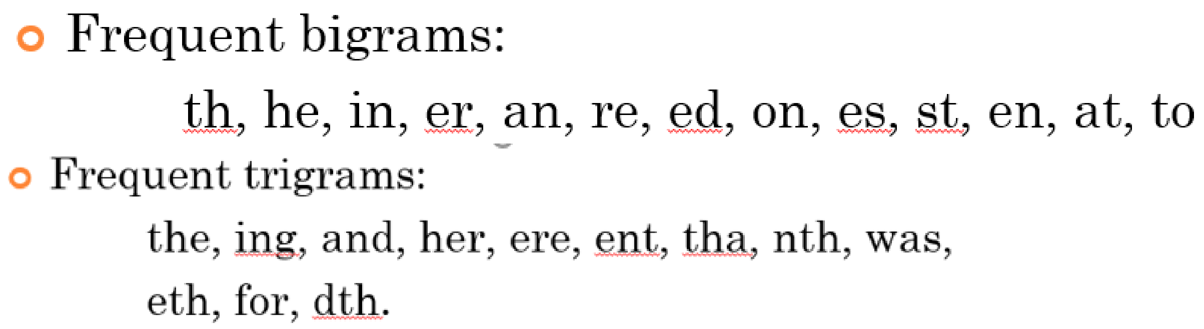


J is the highest of occurrence, and in English word letter of 'e' is the highest probabilities. I make an assumption that the 'J' is 'e' and from the single 'b' character we can assume the it's an 'A' character.

Y	A	B	C	D	E	F	G	H	I
X		**A**							
Y	J	K	L	M	N	O	P	Q	R
X	**E**								
Y	S	T	U	V	W	X	Y	Z	
X									

From the lecture slide, we can roughly know that





- Frequent bigrams:

  th, he, in, er, an, re, ed, on, es, st, en, at, to

- Frequent trigrams:

  the, ing, and, her, ere, ent, tha, nth, was,

  eth, for, dth.

According to the ciphertext 'ruj' 'ru' = 'th' and 'uj' can be 'he'

and since we found 'j' is e *not confirm* which can group into

Y	A	B	C	D	E	F	G	H	I
X		A							
Y	J	K	L	M	N	O	P	Q	R
X	E								**T**
Y	S	T	U	V	W	X	Y	Z	
X			**H**						

Back to the ciphertext, we notice that we can try to guess ‘ujp’, since we know the ‘uj’ is ‘he’ which only have 2 words ‘hex’ or ‘her’. But since it’s English, and frequency of ‘R’ show it’s 8.5% compare to ‘X’ which only have 0.5%. So I decide to put ‘p’ as ‘r’

Y	A	B	C	D	E	F	G	H	I
X		A							
Y	J	K	L	M	N	O	P	Q	R
X	E						**R**		T
Y	S	T	U	V	W	X	Y	Z	
X			H						

Now, we do another guessing on cipher ‘cr’, since we know the cipher text of ‘r’ is ‘t’, there’re a few probability which are ‘at’, ‘it’ but since ‘a’ is occupied in cipher ‘B’. Therefore, ciphertext of ‘c’ is ‘i’

Y	A	B	C	D	E	F	G	H	I
X		A	**I**						
Y	J	K	L	M	N	O	P	Q	R
X	E						R		T
Y	S	T	U	V	W	X	Y	Z	
X			H						

Yikes, we got another plaintext from ciphertext ‘c’. We can try to break ciphertext of ‘rucq’ and plaintext is ‘thi?’ we know that ‘q’ is ‘s’

Y	A	B	C	D	E	F	G	H	I
X		A	I						
Y	J	K	L	M	N	O	P	Q	R
X	E						R	**S**	T
Y	S	T	U	V	W	X	Y	Z	
X			H						

Now, try to break ciphertext of ‘cn’ = ‘I?’, which only have a few word ‘IP’,’IQ’,’IS’,’IF’,’IC’,’IN’. But since it’s a beginning of message so I think ‘IS’ and ‘IN’, ‘IF’ which have higher chances. But the frequency for single character is ‘N’ contain ‘7.75%’ so I assume that ‘N’ is ‘N’

Y	A	B	C	D	E	F	G	H	I
X		A	I						
Y	J	K	L	M	N	O	P	Q	R
X	E				**N**		R	S	T
Y	S	T	U	V	W	X	Y	Z	
X			H						

Since we got new plaintext from character ‘N’, we can now try to break ‘cnrk’, and we know that ‘INT?’ and now we might roughly know that the ciphertext of ‘k’ can be plaintext of ‘o’.

Y	A	B	C	D	E	F	G	H	I
X		A	I						
Y	J	K	L	M	N	O	P	Q	R
X	E	**O**			N		R	S	T
Y	S	T	U	V	W	X	Y	Z	
X			H						

Now we try to break ciphertext of ‘rcojq’ = ‘ti?es’, the character of ‘o’ can be ‘L’ or ‘M’ since ‘times’ or ‘tiles’, but since the message starting from ‘IN’ it might be something like times. Therefore, I assume that ciphertext of ‘O’ is ‘M’

Y	A	B	C	D	E	F	G	H	I
X		A	I						
Y	J	K	L	M	N	O	P	Q	R
X	E	**O**			N	M	R	S	T
Y	S	T	U	V	W	X	Y	Z	
X			H						

Now, let try to break ciphertext of ‘qrcii’ , we found that ‘STI??’, and ‘ii’ is the same which lead me think that the word is ‘still’.

Y	A	B	C	D	E	F	G	H	I
X		A	I						**L**
Y	J	K	L	M	N	O	P	Q	R
X	E	O			N	M	R	S	T
Y	S	T	U	V	W	X	Y	Z	
X			H		**W,G,H**				

After that, now break ciphertext of ‘wbq’ = ‘?AS’ which lead a few words ‘was’,’gas’,’has’, ‘bas’

We will come back later, since it’s too many character which might lead me to fill in the wrong answer.

Let move on to break ciphertext of ‘rkkt’ = ‘too?’ which have 2 words

‘Tool’ and ‘Took’ and since L have be found therefore I think ciphertext of ‘T’ is ‘K’

Y	A	B	C	D	E	F	G	H	I
X		A	I						L
Y	J	K	L	M	N	O	P	Q	R
X	E	O			N	M	R	S	T
Y	S	T	U	V	W	X	Y	Z	
X		**K**	H		W,G,H				

And now we going to break this ciphertext ‘tcbd’ and the plaintext we have is ‘kin?’, Which I think the answer is ‘KING’, ‘KIND’ *We will come back again*.

Y	A	B	C	D	E	F	G	H	I
X		A	I	**G/D**					L
Y	J	K	L	M	N	O	P	Q	R
X	E	O			N	M	R	S	T
Y	S	T	U	V	W	X	Y	Z	
X		K	H		W,G,H				

Now try to break ciphertext of ‘iby’ which gave us the plaintext of ‘LA?’ which I think is ‘LAY’. Therefore, Ciphertext of ‘Y’ is plaintext of ‘Y’.

Y	A	B	C	D	E	F	G	H	I
X		A	I	G/D		W,T,F,H			L
Y	J	K	L	M	N	O	P	Q	R
X	E	O			N	M	R	S	T
Y	S	T	U	V	W	X	Y	Z	
X		K	H		W,G,H		**Y**		

Again, break another ciphertext ‘fbii’ which gave us plaintext of ‘?ALL’

‘WALL’,’TALL’,’FALL’,’HALL’

Decrypt the ciphertext of ‘wjpj’ which gave us ‘?ERE’ and since we have some probability of ‘W’

‘WERE’,’GERE’,’HERE’ which means W can only be ‘W’ or ‘H’

Y	A	B	C	D	E	F	G	H	I
X		A	I	G/D		W,T,F,H			L
Y	J	K	L	M	N	O	P	Q	R
X	E	O			N	M	R	S	T
Y	S	T	U	V	W	X	Y	Z	
X		K	H		**W,H**		Y		

Breaking ciphertext of ‘CRQJIE’ using the plaintext that we have ‘ITSEI?’ which I think the ‘E’ is ‘F’ and ‘U’ is ‘H’ character.

Y	A	B	C	D	E	F	G	H	I
X		A	I	G/D	F				L
Y	J	K	L	M	N	O	P	Q	R
X	E	O			N	M	R	S	T
Y	S	T	U	V	W	X	Y	Z	
X		K	H		**W,H**		Y		

Now, we decide to break ‘ebgj’ which we recover the plaintext into ‘FA?E’. Which is ‘FACE,FADE,FARE,FATE,FAME’. But ‘R,T,M’ which not available because we have found the plaintext therefore left ‘C,D’.

Y	A	B	C	D	E	F	G	H	I
X		A	I	G/D	F		**C,D**		L
Y	J	K	L	M	N	O	P	Q	R
X	E	O			N	M	R	S	T
Y	S	T	U	V	W	X	Y	Z	
X		K	H		W,H		Y		

Now, we know that the cipher that of ‘rupjw’ is ‘THRE?’ So I found know that ‘?’ is ‘W’ which means ciphertext of ‘W’ is ‘W’, now we can update our ciphertext of F from ‘WTH’ to ‘TH’.

Y	A	B	C	D	E	F	G	H	I
X		A	I	G/D	F		C,D		L
Y	J	K	L	M	N	O	P	Q	R
X	E	O			N	M	R	S	T
Y	S	T	U	V	W	X	Y	Z	
X		K	H		**W**		Y		

Now, let break ciphertext of ‘osgu’ ‘M??H’ , which mean ‘M?CH’ or ‘M?DH’ mean ‘MUCH’ therefore ciphertext of ‘S’ is ‘U’

Y	A	B	C	D	E	F	G	H	I
X		A	I	G/D	F		C,D		L
Y	J	K	L	M	N	O	P	Q	R
X	E	O			N	M	R	S	T
Y	S	T	U	V	W	X	Y	Z	
X	**U**	K	H		W		Y		

Now we can try to decrypt the ciphertext of ‘hbsdurjpq’

Since the ciphertext of ‘D’ is either G or D meaning that the ciphertext of ‘hbsdurjpq’ is ‘?AUGHTERS’ or ‘?AUDHTERS’

Therefore, I think the only keyword is ‘LAUGHERS’ or ‘DAUGHTERS’

Y	A	B	C	D	E	F	G	H	I
X		A	I	G/D	F		C,D	**G/D**	L
Y	J	K	L	M	N	O	P	Q	R
X	E	O			N	M	R	S	T
Y	S	T	U	V	W	X	Y	Z	
X	U	K	H		W		Y		

To do confirmation on cipher D and H we take other ciphertext to decrypt such as ‘snhjp’

‘UNDER’ or ‘UNGER’ meaning that UNDER will be the correct plaintext, because no such as called UNGER. So we know that the ciphertext of ‘H’ is ‘D’

Y	A	B	C	D	E	F	G	H	I
X		A	I	G/D	F		C,D	**D**	L
Y	J	K	L	M	N	O	P	Q	R
X	E	O			N	M	R	S	T
Y	S	T	U	V	W	X	Y	Z	
X	U	K	H		W		Y		

Since we know the ciphertext of ‘H’ is plaintext of ‘D’, we update ciphertext of ‘G’

Y	A	B	C	D	E	F	G	H	I
X		A	I	G/D	F		C	D	L
Y	J	K	L	M	N	O	P	Q	R
X	E	O			N	M	R	S	T
Y	S	T	U	V	W	X	Y	Z	
X	U	K	H		W		Y		

After we have confirm the ciphertext of ‘H’ is ‘D’, we do confirmation on ciphertext ‘tcndq’ or ‘tcnd’

We found out that it’s ‘KING’ and ‘KINGS’ for cipher ‘tcndq’ since we know that ciphertext of ‘D’ is ‘G’ and it’s impossible that ciphertext of ‘D’ is ‘D’ since ciphertext of ‘H’ is ‘D’

Y	A	B	C	D	E	F	G	H	I
X		A	I	**G**	F		C	D	L
Y	J	K	L	M	N	O	P	Q	R
X	E	O			N	M	R	S	T
Y	S	T	U	V	W	X	Y	Z	
X	U	K	H		W		Y		

Since ‘WXYZ’ have the sequence there from the ciphertext and plaintext , we know that

Ciphertext of ‘W’= Plaintext of ‘W’

Ciphertext of ‘X’= Plaintext of ‘X’

Ciphertext of ‘Y’= Plaintext of ‘Y’

Ciphertext of ‘Z’= Plaintext of ‘Z’

Y	A	B	C	D	E	F	G	H	I
X		A	I	G	F		C	D	L
Y	J	K	L	M	N	O	P	Q	R
X	E	O			N	M	R	S	T
Y	S	T	U	V	W	X	Y	Z	
X	U	K	H		W	**X**	Y	**Z**	

Now, I need to find ciphertext of ‘V’ from ciphertext ‘ebvkpcrj’ , the plaintext is ‘FA?ORITE’ , We found out that ciphertext of ‘V’ is ‘V’

Ciphertext of ‘V’= Plaintext of ‘V’

Y	A	B	C	D	E	F	G	H	I
X		A	I	G	F		C	D	L
Y	J	K	L	M	N	O	P	Q	R
X	E	O			N	M	R	S	T
Y	S	T	U	V	W	X	Y	Z	
X	U	K	H	**V**	W	X	Y	Z	

After that, we left ciphertext of ‘A,F,G,L,M’

Ciphertext of “fjbsrcesi” is “?EAUTIFUL” which possible “BEAUTIFUL”. So, I know that ciphertext of ‘F’ is ‘B’

Y	A	B	C	D	E	F	G	H	I
X		A	I	G	F	**B**	C	D	L
Y	J	K	L	M	N	O	P	Q	R
X	E	O			N	M	R	S	T
Y	S	T	U	V	W	X	Y	Z	
X	U	K	H	V	W	X	Y	Z	

J,P,Q

Now, let break ciphertext of ‘L’ from ciphertext of ‘libyrucnd’ which I got the plaintext

“?LAYTHING” The plaintext that not discover yet is “J,P,Q” which mean

“JLAYTHING” , “PLAYTHING” , “QLAYTHING” which means that “PLAYTHING” is the correct one because the remaining two is not the correct words.

Therefore, ‘L’ is ‘P’

Y	A	B	C	D	E	F	G	H	I
X		A	I	G	F	B	C	D	L
Y	J	K	L	M	N	O	P	Q	R
X	E	O	**P**		N	M	R	S	T
Y	S	T	U	V	W	X	Y	Z	
X	U	K	H	V	W	X	Y	Z	

Now, left ciphertext of ‘A’ and ‘M’ that is not appear in cipher text so I make assumption. That ciphertext of ‘A’ can be ‘J/Q’ and ciphertext of ‘M’ can also be ‘J/Q’, since no ciphertext therefore I can’t do the cracking part.

Y	A	B	C	D	E	F	G	H	I
X	**J/Q**	A	I	G	F	B	C	D	L
Y	J	K	L	M	N	O	P	Q	R
X	E	O	P	**J/Q**	N	M	R	S	T
Y	S	T	U	V	W	X	Y	Z	
X	U	K	H	V	W	X	Y	Z	

After, I re-arrange the table.

Found out that the cipher text of ‘A’ and ‘M’ are ‘J’ and ‘O’

X	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Y	B	F	G	H	J	**E**	**D**	**U**	**C**	**A**	**T**	**I**	**O**	**N**	K	L	M	P	Q	R	S	V	W	X	Y	Z

Cipher	A	B	C	D	E	F	G	H	I
Plain	**J**	A	I	G	F	B	C	D	L
Cipher	J	K	L	M	N	O	P	Q	R
Plain	E	O	P	**O**	N	M	R	S	T
Cipher	S	T	U	V	W	X	Y	Z	
Plain	U	K	H	V	W	X	Y	Z	

CipherText

cn kihjn rcojq wujn wcqucnd qrcii ujiljh knj rujpj icvjh b tcnd wukqj hbsdurjpq wjpj bii fjbsrcesi fsr ruj yksndjqr wbq qk fjbsrcesi rubr ruj qsn crqjie wucgu ubq qjjn qk osgu wbq bqrkncqujh wujnjvjp cr quknj cn ujp ebgj gikqj fy ruj tcndq gbqrij iby b dpjbr hbpt ekpjqr bnh snhjp bn kih icojrpjj cn ruj ekpjqr wbq b wjii bnh wujn ruj hby wbq vjpy wbpo ruj tcndq gucih wjnr ksr cnrk ruj ekpjqr bnh qbr hkwn fy ruj qchj ke ruj gkki eksnrbcn bnh wujn quj wbq fkpjh quj rkkt b dkihjn fbii bnh rupjw cr sl kn ucdu bnh gbsdur cr bnh rucq fbii wbq ujp ebvkpcrj libyrucnd

Plaintext

in olden times when wishing still helped one there lived a king whose daughters were all beautiful but the youngest was so beautiful that the sun itself which has seen so much was astonished whenever it shone in her face close by the kings castle lay a great dark forest and under an old limetree in the forest was a well and when the day was very warm the kings child went out into the forest and sat down by the side of the cool fountain and when she was bored she took a golden ball and threw it up on high and caught it and this ball was her favorite plaything

Q2

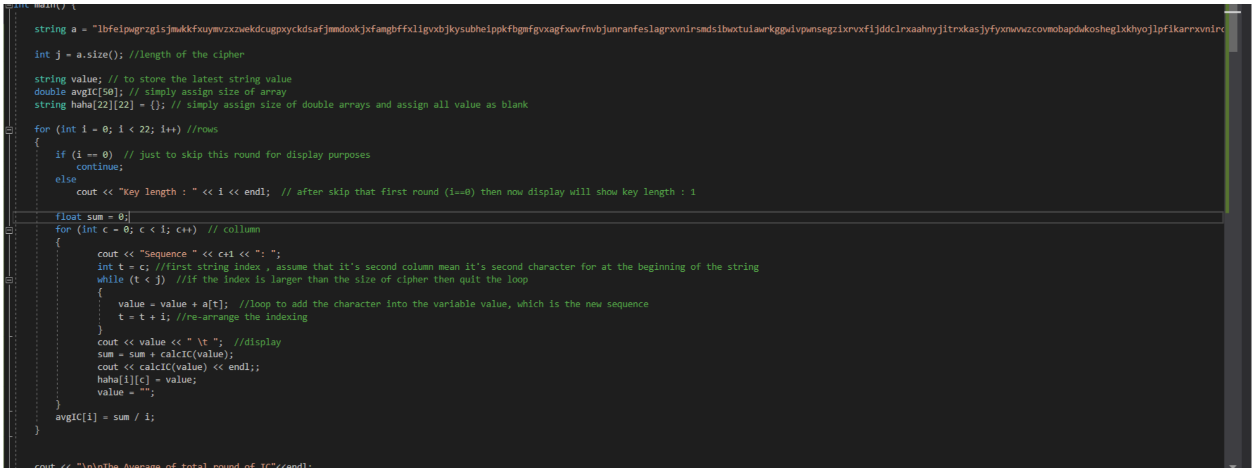
Ciphertext encrypt by vigenere cipher

lbfeipwgrzgisjmwkkfxuymvzxzwekdcugpxyckdsafjmmdoxkjxfamgbffxligvxbjkysubheippkfbgmfgvxagfxwvf
nvbjunranfeslagrxvnirsmdsibwxtuiawrkggwivpwnsegzixrvxfijddclrxaahnyjitrxkasjyfyxnwvwzcovmobapdw
kosheglxkhyojlpfikarrxvnircpafvdzytjixpsjdsheoilpqysdxsrrbjhyooisyhthkrikxjymzabkkfxbyatugtrwyssyexi
kcfixjlbpsnkkkvvimafrxvgpeizabkvwninrkkgvbwx

To break vigenere cipher

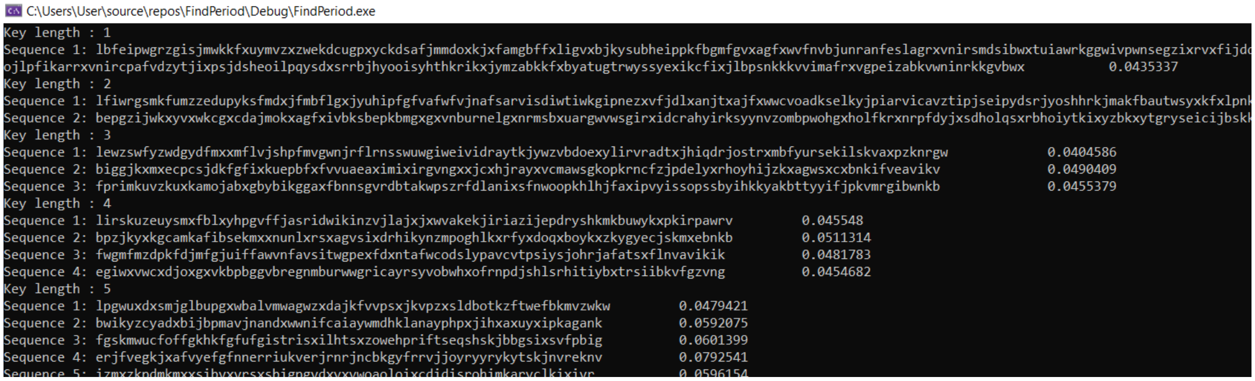
We guess the length of keyword(key) and generate list of sequences from the ciphertext given.

Code :



```
int main() {

    string a = "lbfeipwgrzgisjmwkkfxuymvzxzwekdcugpxyckdsafjmmdoxkjxfamgbffxligvxbjkysubheippkfbgmfgvxagfxwvfnvbjunranfeslagrxvnirsmdsibwxtuiawrkggwivpwnsegzixrvxfijddclrxaahnyjitrxkasjyfyxnwvwzcovmobapdwkosheglxkhyojlpfikarrxvnirc

    int j = a.size(); //length of the cipher

    string value; // to store the latest string value
    double avgIC[50]; // simply assign size of array
    string haha[22][22] = {}; // simply assign size of double arrays and assign all value as blank

    for (int i = 0; i < 22; i++) //rows
    {
        if (i == 0)  // just to skip this round for display purposes
            continue;
        else
            cout << "Key length : " << i << endl;  // after skip that first round (i==0) then now display will show key length : 1

        float sum = 0;
        for (int c = 0; c < i; c++)  // collumn
        {
                cout << "Sequence " << c+1 << ": ";
                int t = c; //first string index , assume that it's second column mean it's second character for at the beginning of the string
                while (t < j)  //if the index is larger than the size of cipher then quit the loop
                {
                    value = value + a[t];  //loop to add the character into the variable value, which is the new sequence
                    t = t + i; //re-arrange the indexing
                }
                cout << value << " \t ";  //display
                sum = sum + calcIC(value);
                cout << calcIC(value) << endl;;
                haha[i][c] = value;
                value = "";
        }
        avgIC[i] = sum / i;
    }


    cout << "\n\nThe Average of total round of IC"<<endl;
```

Output :

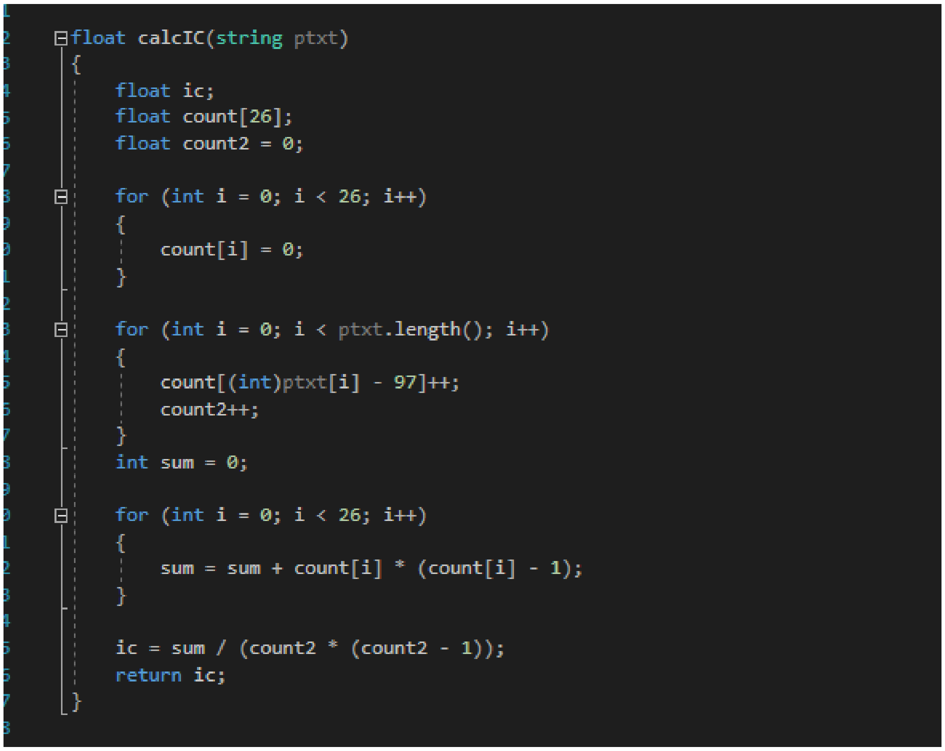


C:\Users\User\source\repos\FindPeriod\Debug\FindPeriod.exe

```
Key length : 1
Sequence 1: lbfeipwgrzgisjmwkkfxuymvzxzwekdcugpxyckdsafjmmdoxkjxfamgbffxligvxbjkysubheippkfbgmfgvxagfxwvfnvbjunranfeslagrxvnirsmdsibwxtuiawrkggwivpwnsegzixrvxfijdd
ojlpfikarrxvnircpafvdzytjixpsjdsheoilpqysdxsrrbjhyooisyhthkrikxjymzabkkfxbyatugtrwyssyexikcfixjlbpsnkkkvvimafrxvgpeizabkvwninrkkgvbwx          0.0435337
Key length : 2
Sequence 1: lfiwrgsmkfumzzedupyksfmdxjfmbflgxjyuhipfgfvafwfvjnafsarvisdiwtiwkgipnezxvfjdlxanjtxajfxwwcvoadkselkyjpiarvicavztipjseipydsrjyoshhrkjmakfbautwsyxkfxlpnk
Sequence 2: bepgzijwkxyvxwkcgxcdajmokxagfxivbksbepkbmgxgxvnburnelgxnrmsbxuargwvwsgirxidcrahyirksyynvzombpwohgxholfkrxnrpfdyjxsdholqsxrbhoiytkixyzbkxytgryseicijbskk
Key length : 3
Sequence 1: lewzswfyzwdgydfmxxmflvjshpfmvgwnjrflrnsswuwgiweividraytkjywzvbdoexylirvradtxjhiqdrjostrxmbfyursekilskvaxpzknrgw            0.0404586
Sequence 2: biggjkxmxecpcsjdkfgfixkuepbfxfvvuaeaximixirgvngxxjcxhjrayxvcmawsgkopkrncfzjpdelyxrhoyhijzkxagwsxcxbnkifveavikv            0.0490409
Sequence 3: fprimkuvzkuxkamojabxgbybikggaxfbnnsgvrdbtakwpszrfdlanixsfnwoopkhlhjfaxipvyissopssbyihkkyakbttyyifjpkvmrgibwnkb            0.0455379
Key length : 4
Sequence 1: lirskuzeuysmxfblxyhpgvffjasridwikinzvjlajxjxwvakekjiriazijepdryshkmkbuwykxpkirpawrv         0.045548
Sequence 2: bpzjkyxkgcamkafibsekmxxnunlxrsxagvsixdrhikynzmpoghlkxrfyxdoqxboykxzkygyecjskmxebnkb         0.0511314
Sequence 3: fwgmfmzdpkfdjmfgjuiffawvnfavsitwgpexfdxntafwcodslypavcvtpsiysjohrjafatsxflnvavikik         0.0481783
Sequence 4: egiwxvwcxdjoxgxvkbpbggvbregnmburwwgricayrsyvobwhxofrnpdjshlsrhitiybxtrsiibkvfgzvng         0.0454682
Key length : 5
Sequence 1: lpgwuxdxsmjglbupgxwbalvmwagwzxdajkfvvpsxjkvpzxsldbotkzftwefbkmvzwkw         0.0479421
Sequence 2: bwikyzcyadxbijbpmavjnandxwwnifcaiaywmdhklanayphpxjihxaxuyxipkagank         0.0592075
Sequence 3: fgskmwucfoffgkhkfgfufgistrisxilhtsxzowehpriftseqshskjbbgsixsvfpbig         0.0601399
Sequence 4: erjfvegkjxafvyefgfnnerriukverjrnrjncbkgyfrrvjjoyryyrykytskjnvreknv         0.0792541
Sequence 5: izmxzkpdmkmxxsibvxvrsxshignpgvdxvxvwoaoloixcdidisrohimkarvclkixivr         0.0596154
```

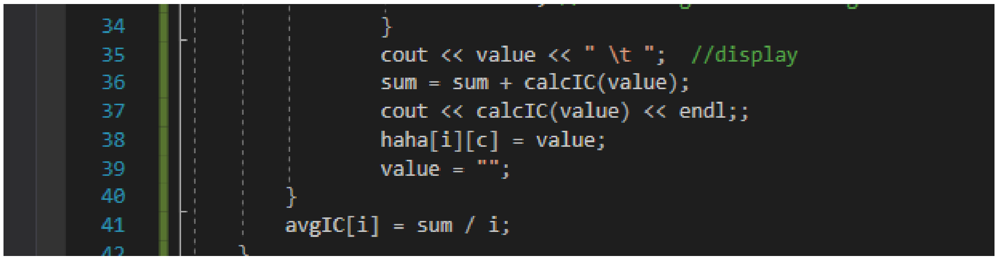
After we have generate list of sequences , we have to look for the IC value using the sequences that we generate from the ciphertext.

This is the function of calculate the IC value



```cpp
float calcIC(string ptxt)
{
    float ic;
    float count[26];
    float count2 = 0;

    for (int i = 0; i < 26; i++)
    {
        count[i] = 0;
    }

    for (int i = 0; i < ptxt.length(); i++)
    {
        count[(int)ptxt[i] - 97]++;
        count2++;
    }
    int sum = 0;

    for (int i = 0; i < 26; i++)
    {
        sum = sum + count[i] * (count[i] - 1);
    }

    ic = sum / (count2 * (count2 - 1));
    return ic;
}
```

Code for displaying the sequence

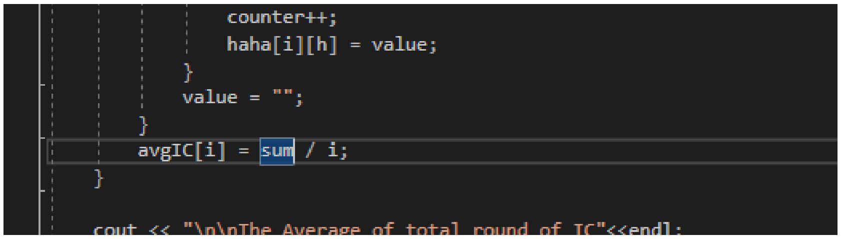


```cpp
                }
                cout << value << " \t ";  //display
                sum = sum + calcIC(value);
                cout << calcIC(value) << endl;;
                haha[i][c] = value;
                value = "";
            }
            avgIC[i] = sum / i;
        }
```

From here we can see that

calcIC(value) is calling the functions with the parameter called string which actually is the sequence value, which generate the IC values in our terminal.

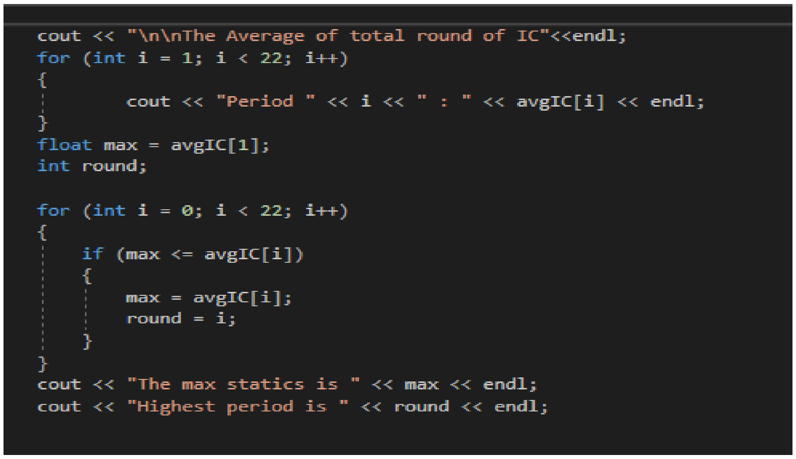
And, I store all the IC value reason able to calculate the value of averageIC.



```cpp
                counter++;
                haha[i][h] = value;
            }
            value = "";
        }
        avgIC[i] = sum / i;
    }

    cout << "\n\nThe Average of total round of IC"<<endl;
```

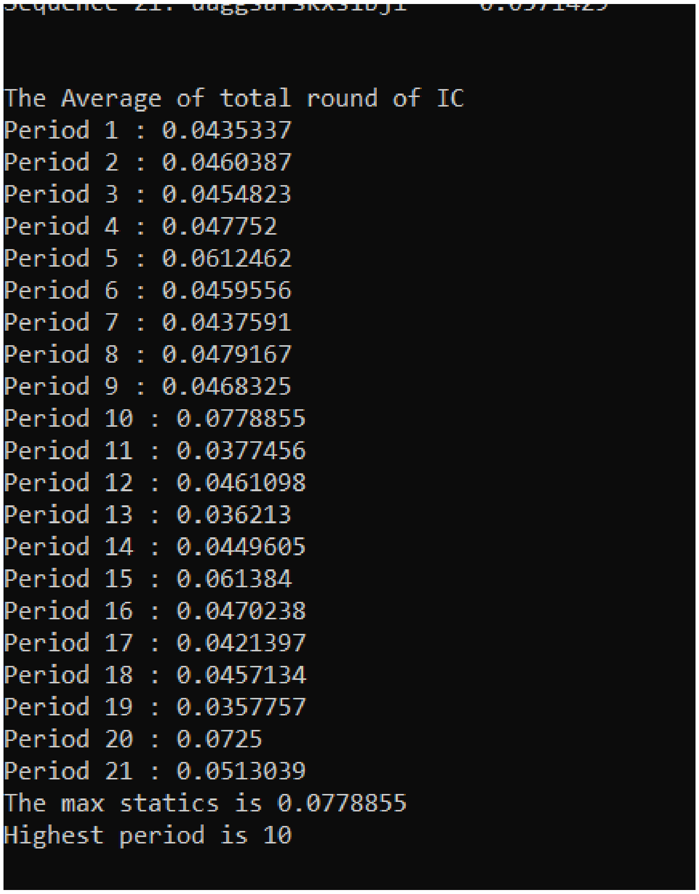
From code , we can see that sum = sum +calcIC(value)

Therefore , we divide according to the keylength(total of sequence).



```
cout << "\n\nThe Average of total round of IC"<<endl;
for (int i = 1; i < 22; i++)
{
        cout << "Period " << i << " : " << avgIC[i] << endl;
}
float max = avgIC[1];
int round;

for (int i = 0; i < 22; i++)
{
    if (max <= avgIC[i])
    {
        max = avgIC[i];
        round = i;
    }
}
cout << "The max statics is " << max << endl;
cout << "Highest period is " << round << endl;
```

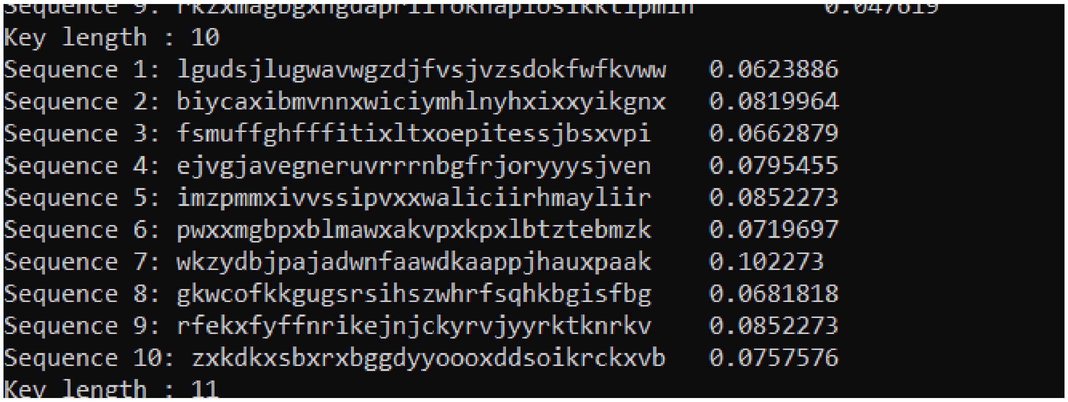
The code above is to find the highest IC value and we know which round it's the highest.



```
Sequence 21: uagg3uf3kx31bj1      0.0571429


The Average of total round of IC
Period 1 : 0.0435337
Period 2 : 0.0460387
Period 3 : 0.0454823
Period 4 : 0.047752
Period 5 : 0.0612462
Period 6 : 0.0459556
Period 7 : 0.0437591
Period 8 : 0.0479167
Period 9 : 0.0468325
Period 10 : 0.0778855
Period 11 : 0.0377456
Period 12 : 0.0461098
Period 13 : 0.036213
Period 14 : 0.0449605
Period 15 : 0.061384
Period 16 : 0.0470238
Period 17 : 0.0421397
Period 18 : 0.0457134
Period 19 : 0.0357757
Period 20 : 0.0725
Period 21 : 0.0513039
The max statics is 0.0778855
Highest period is 10
```

Therefore , we know that the highest period is 10, which we can believe/ believe higher chance that the keywork length is '10'

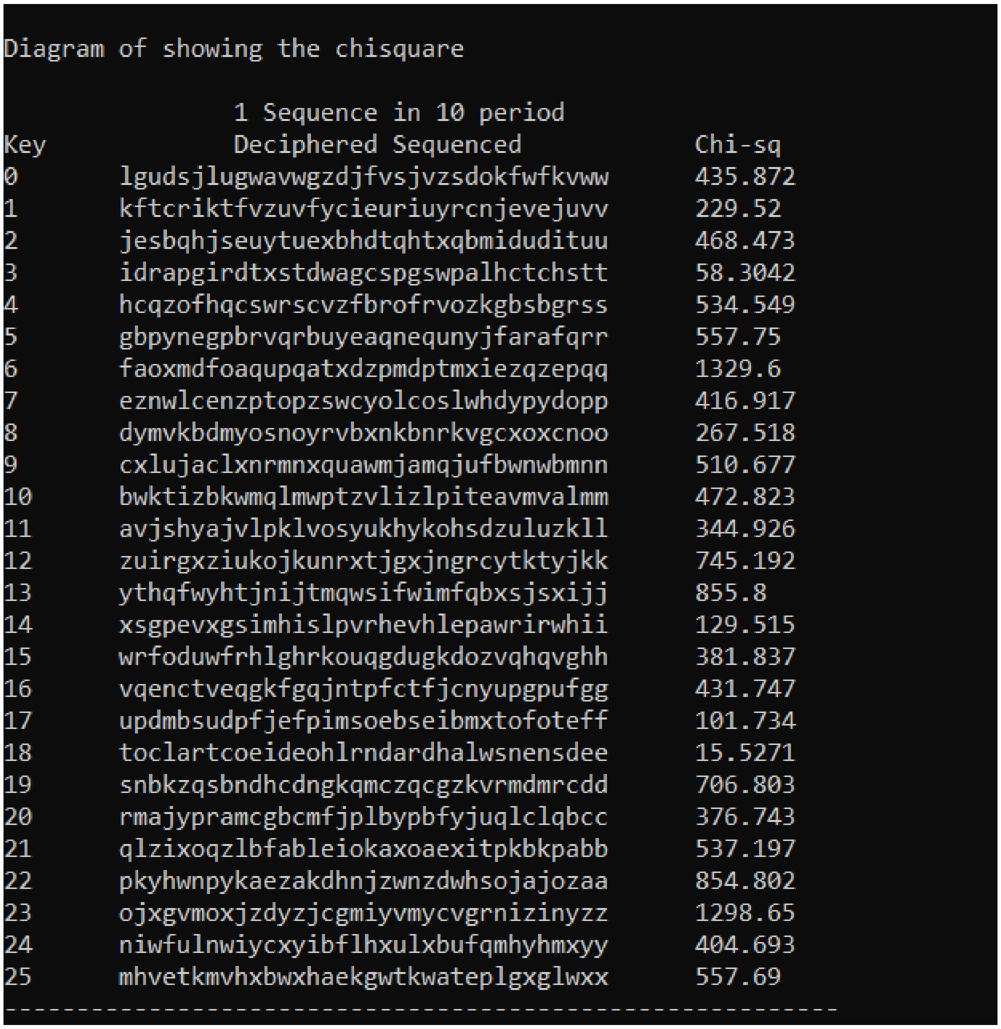
Since , we know that the sequence is ‘10’ , we look back to the diagram of key length 10



```
Sequence 9: rkzxmagbgxngdaprilfoknaplosikktipmin        0.047619
Key length : 10
Sequence 1: lgudsjlugwavwgzdjfvsjvzsdokfwfkvww   0.0623886
Sequence 2: biycaxibmvnnxwiciymhlnyhxixxyikgnx   0.0819964
Sequence 3: fsmuffghfffitixltxoepitessjbsxvpi    0.0662879
Sequence 4: ejvgjavegneruvrrrnbgfrjoryyysjven    0.0795455
Sequence 5: imzpmmxivvssipvxxwaliciirhmayliir    0.0852273
Sequence 6: pwxxmgbpxblmawxakvpxkpxlbtztebmzk    0.0719697
Sequence 7: wkzydbjpajadwnfaawdkaappjhauxpaak    0.102273
Sequence 8: gkwcofkkgugsrsihszwhrfsqhkbgisfbg    0.0681818
Sequence 9: rfekxfyffnrikejnjckyrvjyyrktknrkv    0.0852273
Sequence 10: zxkdkxsbxrxbggdyyoooxddsoikrckxvb   0.0757576
Key length : 11
```

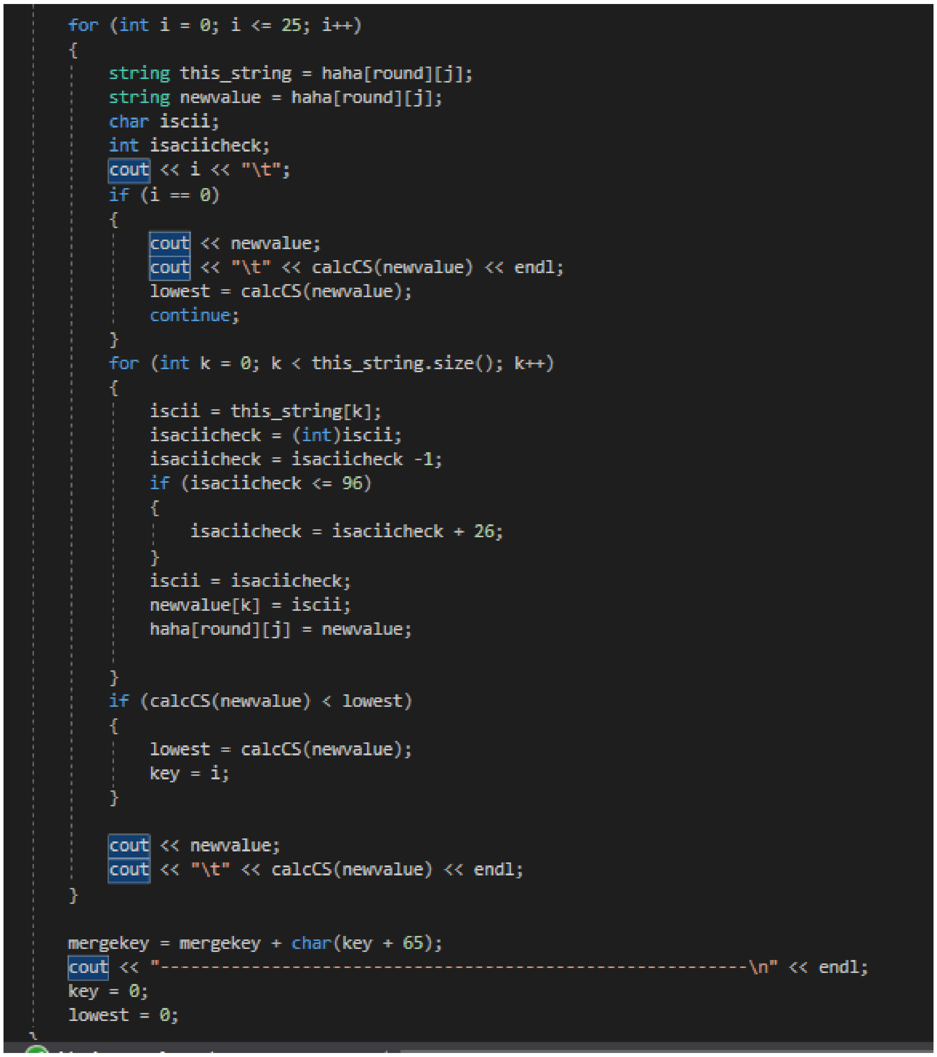
Using the 10th Key length of all the 10 sequence .

For sequence1 , which shift 26 times using 0 to 25



```
Diagram of showing the chisquare

                1 Sequence in 10 period
Key             Deciphered Sequenced            Chi-sq
0       lgudsjlugwavwgzdjfvsjvzsdokfwfkvww      435.872
1       kftcriktfvzuvfycieuriuyrcnjevejuvv      229.52
2       jesbqhjseuytuexbhdtqhtxqbmiduditu u     468.473
3       idrapgirdtxstdwagcspgswpalhctchstt      58.3042
4       hcqzofhqcswrscvzfbrofrvozkgbsbgrss      534.549
5       gbpynegpbrvqrbuyeaqnequnyjfarafqrr      557.75
6       faoxmdfoaqupqatxdzpmdptmxiezqzepqq      1329.6
7       eznwlcenzptopzswcyolcoslwhdypydopp      416.917
8       dymvkbdmyosnoyrvbxnkbnrkvgcxoxcnoo      267.518
9       cxlujaclxnrmnxquawmjamqjufbwnwbmnn      510.677
10      bwktizbkwmqlmwptzvlizlpiteavmvalmm      472.823
11      avjshyajvlpklvosyukhykohsdzuluzkll      344.926
12      zuirgxziukojkunrxtjgxjngrcytktyjkk      745.192
13      ythqfwyhtjnijtmqwsifwimfqbxsjsxijj      855.8
14      xsgpevxgsimhislpvrhevhlepawrirwhii      129.515
15      wrfoduwfrhlghrkouqgdugkdozvqhqvghh      381.837
16      vqenctveqgkfgqjntpfctfjcnyupgpufgg      431.747
17      updmbsudpfjefpimsoebseibmxtofoteff      101.734
18      toclartcoeideohlrndardhalwsnensdee      15.5271
19      snbkzqsbndhcdngkqmczqcgzkvrmdmrcdd      706.803
20      rmajypramcgbcmfjplbypbfyjuqlclqbcc      376.743
21      qlzixoqzlbfableiokaxoaexitpkbkpabb      537.197
22      pkyhwnpykaezakdhnjzwnzdwhsojajozaa      854.802
23      ojxgvmoxjzdyzjcgmiyvmycvgrniziny zz     1298.65
24      niwfulnwiycxyibflhxulxbufqmhyhmxyy      404.693
25      mhvetkmvhxbwxhaekgwtkwateplgxglwxx      557.69
---------------------------------------------------------
```

Which shows here , and again



```
for (int i = 0; i <= 25; i++)
{
    string this_string = haha[round][j];
    string newvalue = haha[round][j];
    char iscii;
    int isaciicheck;
    cout << i << "\t";
    if (i == 0)
    {
        cout << newvalue;
        cout << "\t" << calcCS(newvalue) << endl;
        lowest = calcCS(newvalue);
        continue;
    }
    for (int k = 0; k < this_string.size(); k++)
    {
        iscii = this_string[k];
        isaciicheck = (int)iscii;
        isaciicheck = isaciicheck -1;
        if (isaciicheck <= 96)
        {
            isaciicheck = isaciicheck + 26;
        }
        iscii = isaciicheck;
        newvalue[k] = iscii;
        haha[round][j] = newvalue;

    }
    if (calcCS(newvalue) < lowest)
    {
        lowest = calcCS(newvalue);
        key = i;
    }

    cout << newvalue;
    cout << "\t" << calcCS(newvalue) << endl;
}

mergekey = mergekey + char(key + 65);
cout << "----------------------------------------------------------\n" << endl;
key = 0;
lowest = 0;
```

This is the algorithm to do the shifting of 26 times .

This shifting mean value of ‘a’ in ASCII is 97, shift 26 times mean + 1, starting from 0 to 25 in total of 26 times.

For first shift 97 + 0 =a

For second shift 97 + 1 = b

And so on.

If value such as X which equivalent to 120

For first shifting 120 + 0 = x

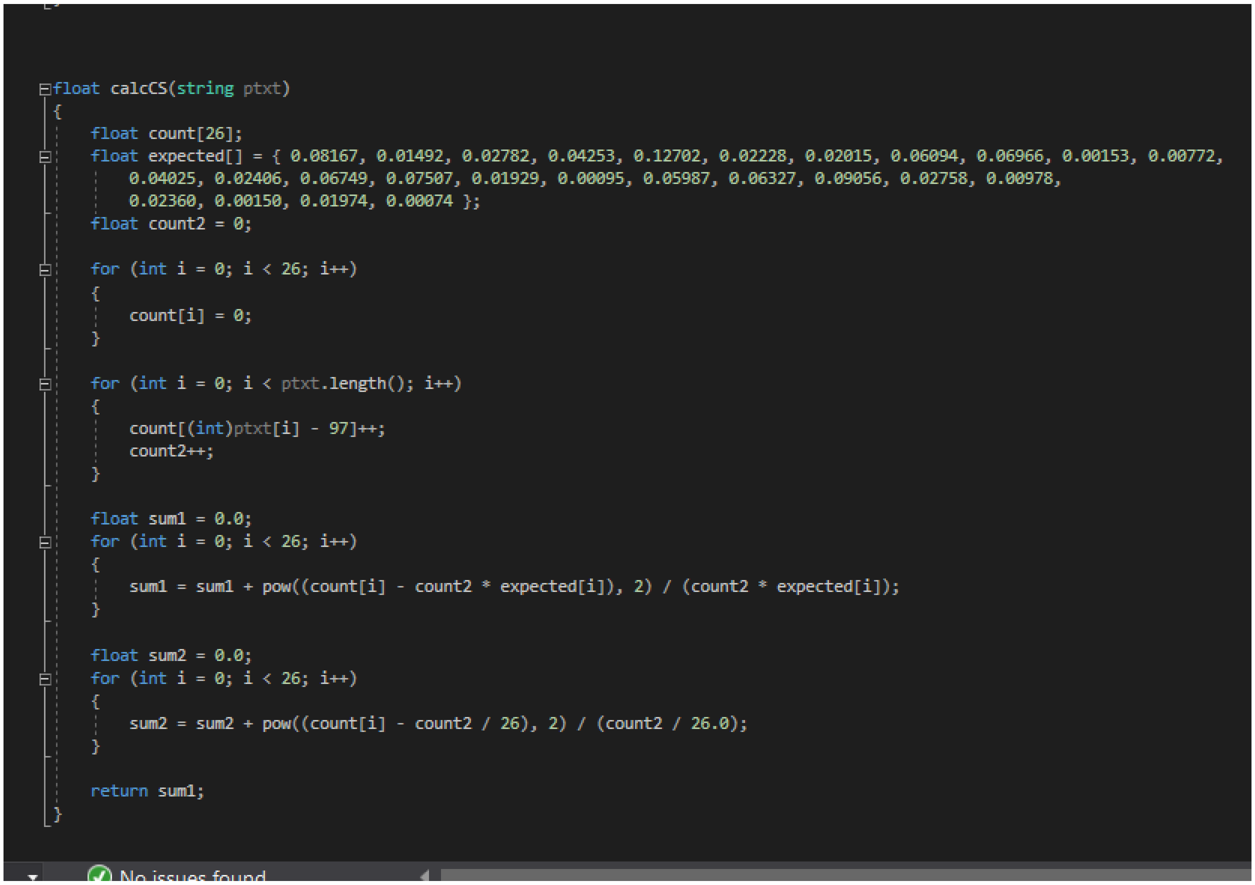
Second shifting 120 + 1 = y

Third shifting 120 + 2 = z

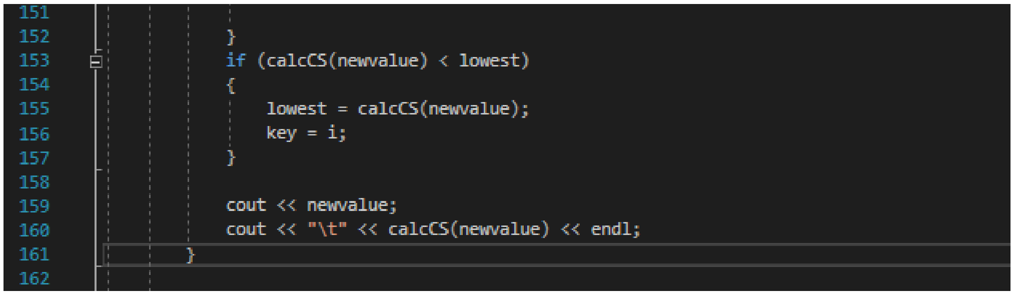
Forth shifting 120 + 3 = [(123 MOD 122) + 96] which is 97 = a

And after finish the shifting steps, I start to calculate value of CHISQUARE

Below code is the function to calculate chisquare



```
float calcCS(string ptxt)
{
    float count[26];
    float expected[] = { 0.08167, 0.01492, 0.02782, 0.04253, 0.12702, 0.02228, 0.02015, 0.06094, 0.06966, 0.00153, 0.00772,
        0.04025, 0.02406, 0.06749, 0.07507, 0.01929, 0.00095, 0.05987, 0.06327, 0.09056, 0.02758, 0.00978,
        0.02360, 0.00150, 0.01974, 0.00074 };
    float count2 = 0;

    for (int i = 0; i < 26; i++)
    {
        count[i] = 0;
    }

    for (int i = 0; i < ptxt.length(); i++)
    {
        count[(int)ptxt[i] - 97]++;
        count2++;
    }

    float sum1 = 0.0;
    for (int i = 0; i < 26; i++)
    {
        sum1 = sum1 + pow((count[i] - count2 * expected[i]), 2) / (count2 * expected[i]);
    }

    float sum2 = 0.0;
    for (int i = 0; i < 26; i++)
    {
        sum2 = sum2 + pow((count[i] - count2 / 26), 2) / (count2 / 26.0);
    }

    return sum1;
}
```



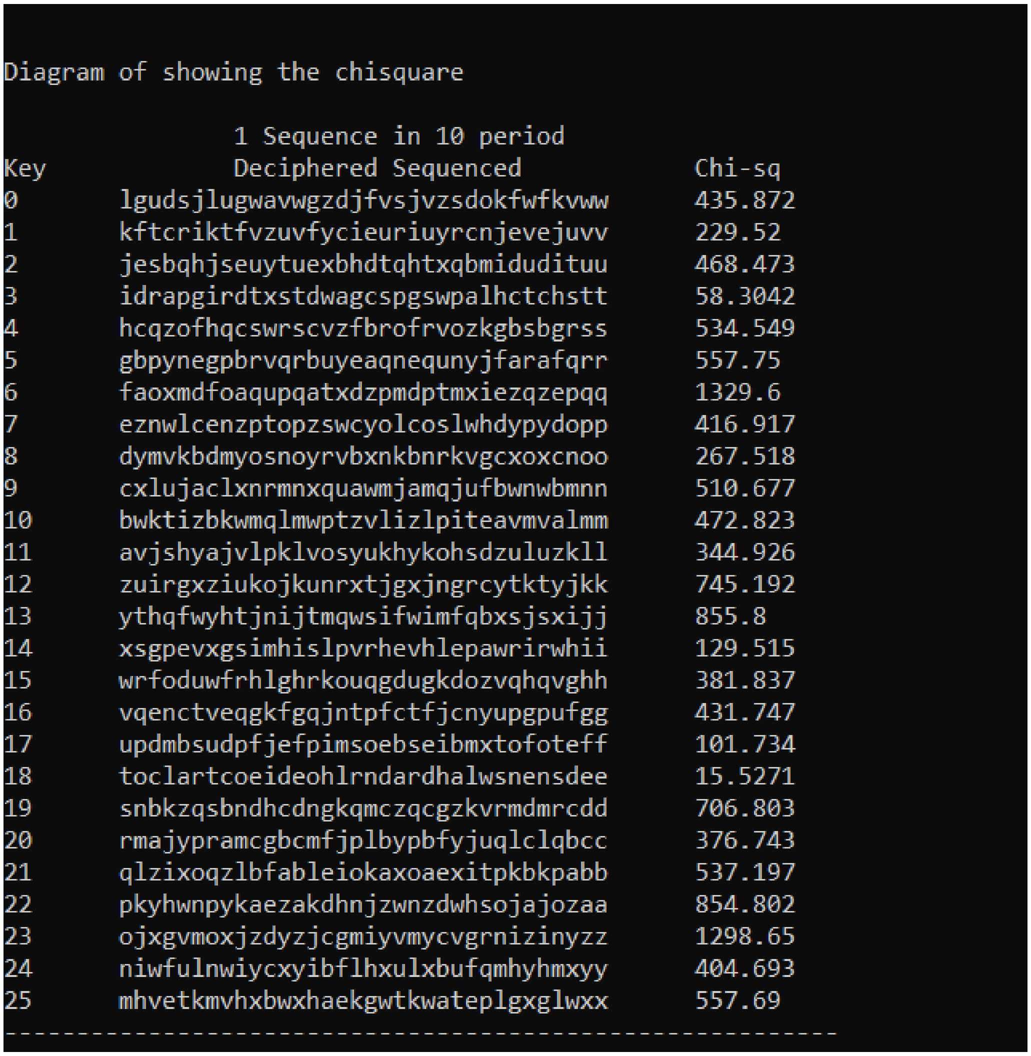
```
            }
            if (calcCS(newvalue) < lowest)
            {
                lowest = calcCS(newvalue);
                key = i;
            }

            cout << newvalue;
            cout << "\t" << calcCS(newvalue) << endl;
        }
```

Variable of “newvalue” is the value of after performing the shifting

We can just call that function with the (shifted value), and we have too look for the lowest chi-square value

In our scenario, let pick the first sequence in the period of 10.

From there all I have to do is to pick all lowest CHI-SQ value in all the sequence in 10th period.



```
Diagram of showing the chisquare

                1 Sequence in 10 period
Key             Deciphered Sequenced            Chi-sq
0       lgudsjlugwavwgzdjfvsjvzsdokfwfkvww      435.872
1       kftcriktfvzuvfycieuriuyrcnjevejuvv      229.52
2       jesbqhjseuytuexbhdtqhtxqbmiduditjuu     468.473
3       idrapgirdtxstdwagcspgswpalhctchstt      58.3042
4       hcqzofhqcswrscvzfbrofrvozkgbsbgrss      534.549
5       gbpynegpbrvqrbuyeaqnequnyjfarafqrr      557.75
6       faoxmdfoaqupqatxdzpmdptmxiezqzepqq      1329.6
7       eznwlcenzptopzswcyolcoslwhdypydopp      416.917
8       dymvkbdmyosnoyrvbxnkbnrkvgcxoxcnoo      267.518
9       cxlujaclxnrmnxquawmjamqjufbwnwbmnn      510.677
10      bwktizbkwmqlmwptzvlizlpiteavmvalmm      472.823
11      avjshyajvlpklvosyukhykohsdzuluzkll      344.926
12      zuirgxziukojkunrxtjgxjngrcytktyjkk      745.192
13      ythqfwyhtjnijtmqwsifwimfqbxsjsxijj      855.8
14      xsgpevxgsimhislpvrhevhlepawrirwhii      129.515
15      wrfoduwfrhlghrkouqgdugkdozvqhqvghh      381.837
16      vqenctveqgkfgqjntpfctfjcnyupgpufgg      431.747
17      updmbsudpfjefpimsoebseibmxtofoteff      101.734
18      toclartcoeideohlrndardhalwsnensdee      15.5271
19      snbkzqsbndhcdngkqmczqcgzkvrmdmrcdd      706.803
20      rmajypramcgbcmfjplbypbfyjuqlclqbcc      376.743
21      qlzixoqzlbfableiokaxoaexitpkbkpabb      537.197
22      pkyhwnpykaezakdhnjzwnzdwhsojajozaa      854.802
23      ojxgvmoxjzdyzjcgmiyvmycvgrniziinyzz     1298.65
24      niwfulnwiycxyibflhxulxbufqmhyhmxyy      404.693
25      mhvetkmvhxbwxhaekgwtkwateplgxglwxx      557.69
----------------------------------------------------------
```

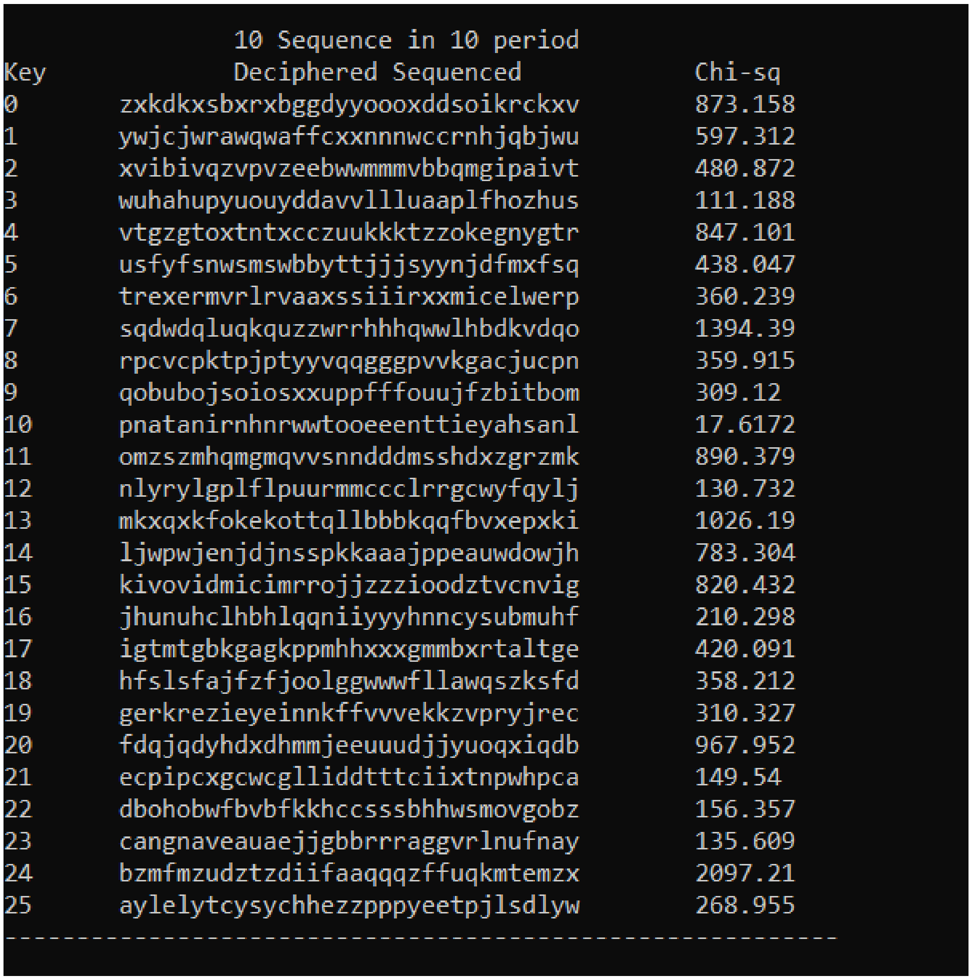
From the output diagram shown, we found out that the first character of the key is ‘S’ since it’s stating key 16 and it’s the lowest which have 15.5271 value in the 1st sequence .

We can see that the highest chi-sq value is at key 18

A	B	C	D	E	F	G	H	I	J	K	l	M	N	O	P	Q	R	**S**	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	**18**	19	20	21	22	23	24	25

Next, we skip all the 2,3,4,5,6,….9th sequence till 10 sequence .

From the 10th sequence which pointing to the character of ‘K’ and



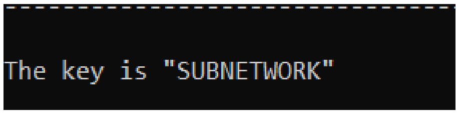
```
                10 Sequence in 10 period
Key             Deciphered Sequenced            Chi-sq
0       zxkdkxsbxrxbggdyyoooxddsoikrckxv        873.158
1       ywjcjwrawqwaffcxxnnnwccrnhjqbjwu        597.312
2       xvibivqzvpvzeebwwmmmvbbqmgipaivt        480.872
3       wuhahupyuouyddavvllluaaplfhozhus        111.188
4       vtgzgtoxtntxcczuukkktzzokegnygtr        847.101
5       usfyfsnwsmswbbyttjjjsyynjdfmxfsq        438.047
6       trexermvrlrvaaxssiiirxxmicelwerp        360.239
7       sqdwdqluqkquzzwrrhhhqwwlhbdkvdqo        1394.39
8       rpcvcpktpjptyyvqqgggpvvkgacjucpn        359.915
9       qobubojsoiosxxuppfffouujfzbitbom        309.12
10      pnatanirnhnrwwtooeeenttieyahsanl        17.6172
11      omzszmhqmgmqvvsnndddmsshdxzgrzmk        890.379
12      nlyrylgplflpuurmmccclrrgcwyfqylj        130.732
13      mkxqxkfokekottqllbbbkqqfbvxepxki        1026.19
14      ljwpwjenjdjnsspkkaaajppeauwdowjh        783.304
15      kivovidmicimrrojjzzzioodztvcnvig        820.432
16      jhunuhclhbhlqqniiyyyhnncysubmuhf        210.298
17      igtmtgbkgagkppmhhxxxgmmbxrtaltge        420.091
18      hfslsfajfzfjoolggwwwfllawqszksfd        358.212
19      gerkrezieyeinnkffvvvekkzvpryjrec        310.327
20      fdqjqdyhdxdhmmjeeuuudjjyuoqxiqdb        967.952
21      ecpipcxgcwcglliddtttciixtnpwhpca        149.54
22      dbohobwfbvbfkkhccsssbhhwsmovgobz        156.357
23      cangnaveauaejjgbbrrraggvrlnufnay        135.609
24      bzmfmzudztzdiifaaqqqzffuqkmtemzx        2097.21
25      aylelytcysychhezzpppyeetpjlsdlyw        268.955
-----------------------------------------------------------
```

Reason it’s ‘K’ because the 10th sequence it state that the lowest value of Chi-SQ is 17.6172 which pointing to key 10

A	B	C	D	E	F	G	H	I	J	K	l	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

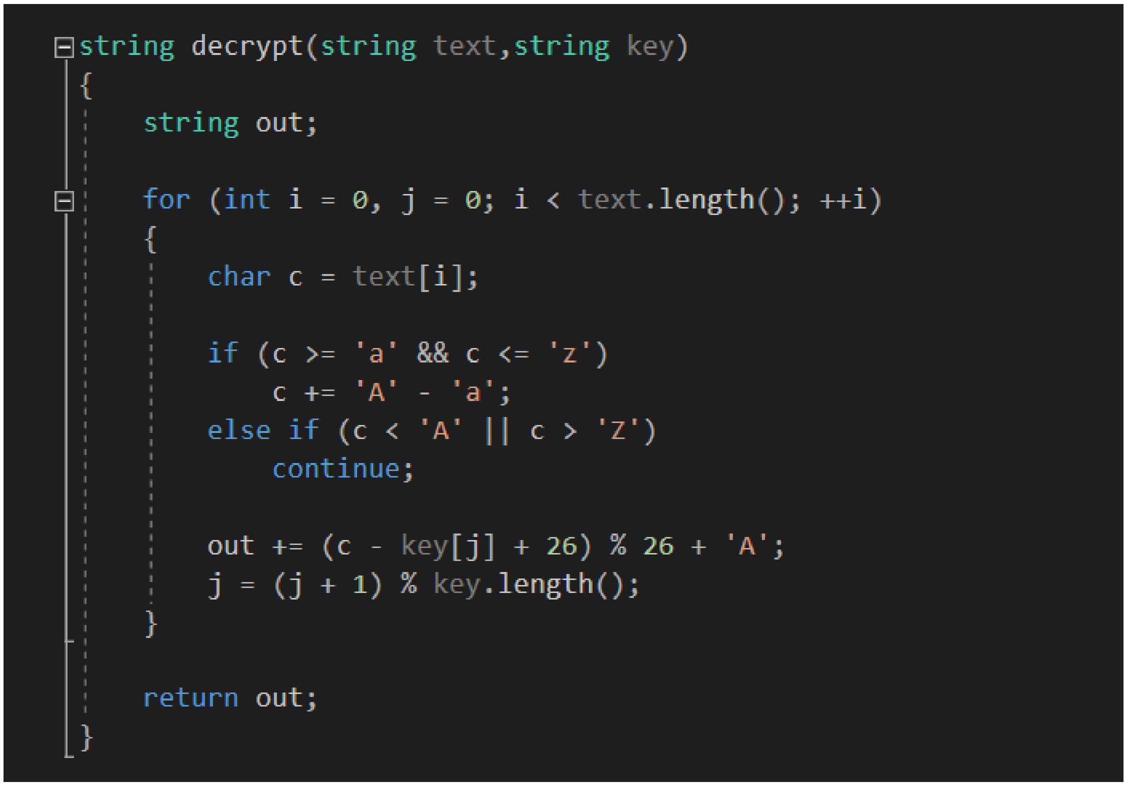
And the key ‘10’ is represent as character of ‘K’

So after we merge those key into a single character into a string, and the string is “SUBNETWORK” which is our key.



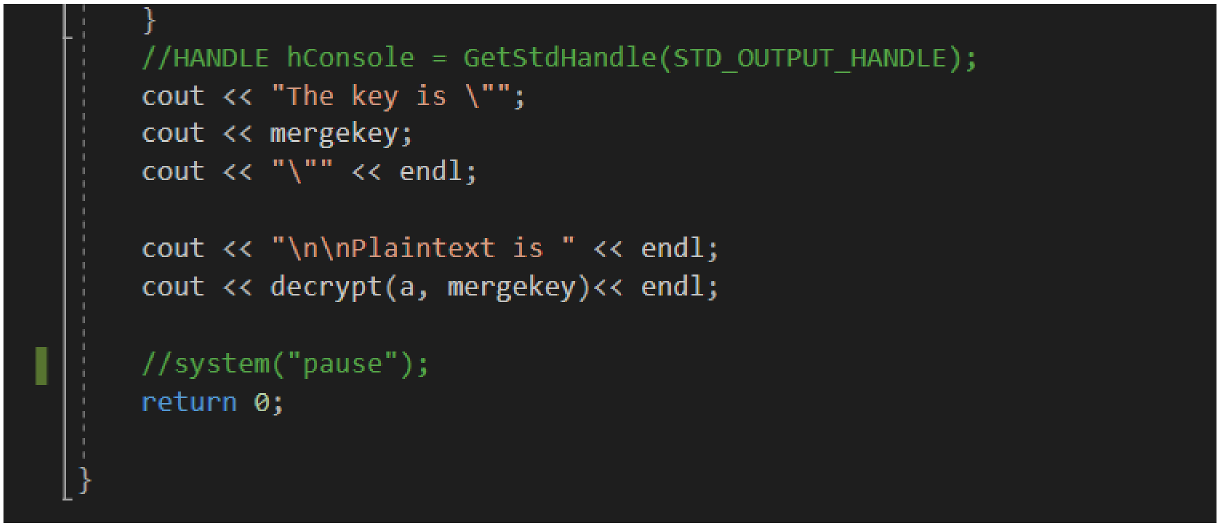
```
--------------------------------

The key is "SUBNETWORK"
```

I have attached an algorithm to decrypt the entire ciphertext to plaintext

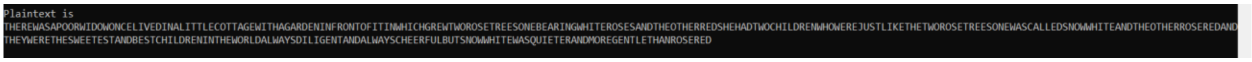


```
string decrypt(string text,string key)
{
    string out;

    for (int i = 0, j = 0; i < text.length(); ++i)
    {
        char c = text[i];

        if (c >= 'a' && c <= 'z')
            c += 'A' - 'a';
        else if (c < 'A' || c > 'Z')
            continue;

        out += (c - key[j] + 26) % 26 + 'A';
        j = (j + 1) % key.length();
    }

    return out;
}
```

So it accept 2 parameter, first parameter the variable named ‘text’ which is the “ciphertext”, whereby the variable named “key” which hold value of “SUBNETWORK”



```
    }
    //HANDLE hConsole = GetStdHandle(STD_OUTPUT_HANDLE);
    cout << "The key is \"";
    cout << mergekey;
    cout << "\"" << endl;

    cout << "\n\nPlaintext is " << endl;
    cout << decrypt(a, mergekey)<< endl;

    //system("pause");
    return 0;

}
```



```
Plaintext is
THEREWASAPOORWIDOWONCELIVEDINALITTLECOTTAGEWITHAGARDENINFRONTOFITINWHICHGREWTWOROSETREESONEBEARINGWHITEROSESANDTHEOTHERREDSHEHADTWOCHILDRENWHOWEREJUSTLIKETHETWOROSETREESONEWASCALLEDSNOWWHITEANDTHEOTHERROSEREDAND
THEYWERETHESWEETESTANDBESTCHILDRENINTHEWORLDALWAYSDILIGENTANDALWAYSCHEERFULBUTSNOWWHITEWASQUIETERANDMOREGENTLETHANROSERED
```

So the plaintext is

“THERE WAS A POOR WIDOW ONCE LIVED IN A LITTLE COTTAGE WITH A GARDEN IN FRONT OF IT IN WHICH GREW TWO ROSE TREES ONE BEARING WHITE ROSES AND THE OTHER RED SHE HAD TWO CHILDREN WHO WERE JUST LIKE THE TWO ROSE TREES ONE WAS CALLED SNOW WHITE AND THE OTHER ROSE RED AND THEY WERE THE SWEE TEST AND BEST CHILDREN IN THE WORLD ALWAYS DILIGENT AND ALWAYS CHEERFUL BUT SNOW WHITE WAS QUIETER AND MORE GENTLE THAN ROSERED”

Key is

“SUBNETWORK”

Ciphertext is

```
“lbfeipwgrzgisjmwkkfxuymvzxzwekdcugpxyckdsafjmmdoxkjxfamgbffxligvxbjkysubheippkfbgmfgvxag
fxwvfnvbjunranfeslagrxvnirsmdsibwxtuiawrkggwivpwnsegzixrvxfijddclrxaahnyjitrxkasjyfyxnwvw
zcovmobapdwkosheglxkhyojlpfikarrxvnircpafvdzytjixpsjdsheoilpqysdxsrrbjhyooisyhthkrikxjymz
abkkfxbyatugtrwyssyexikcfixjlbpsnkkkvvimafrxvgpeizabkvwninrkkgvbwx”
```

Difference between Monoalphabetic cipher and Vigenere cipher

Monoalphabetic cipher is basically replacing original character(plaintext) to other character which can be use statistical analysis to decode the ciphertext back to plaintext.

Example 1 :

X	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Y	**K**	**E**	**Y**	**W**	**O**	**R**	**D**	A	B	C	F	G	H	I	J	L	M	N	P	Q	S	T	U	V	X	Z

We can use above key “keyword” to do our encryption.

Let say the plaintext of “HELLOWORLD”

The ciphertext will be AOGGJUJNGW

Example 2 :

Or we can change the value of key and it position

X	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Y	M	N	O	P	Q	R	V	W	X	Z	**S**	**T**	**U**	**D**	**Y**	A	B	C	E	F	G	H	I	J	K	L

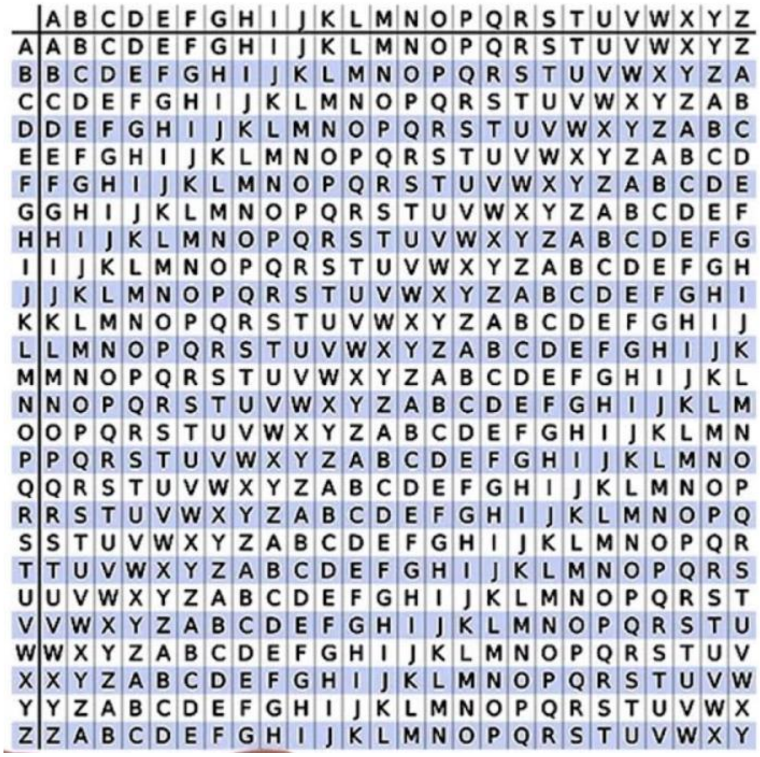
Same , we encrypt back the plaintext of “HELLOWORLD”

Ciphertext = WQTTYIYCTP

it can be decrypt easily using statistical analysis.

More ciphertext more accurate and faster to find the plaintext.

Vigenere cipher, cannot be break using statistical analysis reason is it use a table to do encoding and decoding (if receiver know the key to encrypt)



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Assume, Bob encrypt some plaintext into a ciphertext, and want to send to Alice

Bob want to send “BABY” which is our plaintext, using the key “SAM”

Plaintext=“BABY” Key= “SAM”

Ciphertext = “TANQ”, from the scenario we can see that the BABY which consist of 2 character of ‘B’ but the ciphertext now is ‘T’ & ‘N’.

Assume, alice know the key and Bob will send “TANQ” to Alice.

Alice will refer back to that table and do decoding which able to decrypt the ciphertext into plaintext, and Alice able to get plaintext of “BABY”.