МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

НАВЧАЛЬНО-НАУКОВИЙ КОМПЛЕКС

«ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ»

Лабораторна робота № 2

з курсу «Технології захисту інформації»

Тема: «Дослідження сучасних симетричних систем шифрування»

Виконав:

студент IV курсу

групи ДА-32

Колінько Анжела

Київ – 2017

Варіант 11

1. Ключі для кожного шифру:

a. DES (ECB), key={59,56,152,55,21,32,247,94} (hex: 3b 38 98 37 15 20 f7 5e)

b. DES (CBC), key={59,56,152,55,21,32,247,94}

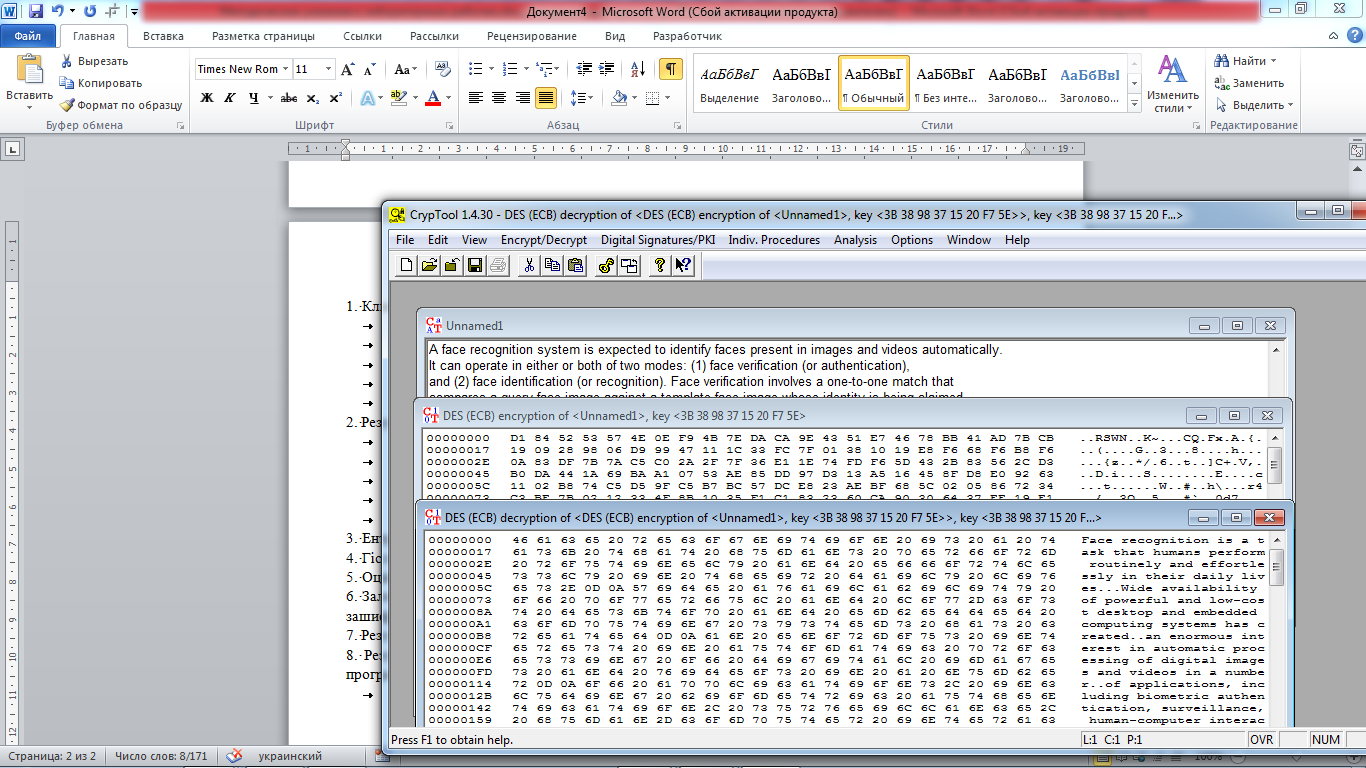
c. TripleDES (ECB), key[i]={59,56,152,55,21,32,247,94}

d. TripleDES (CBC), key[i]={59,56,152,55,21,32,247,94}

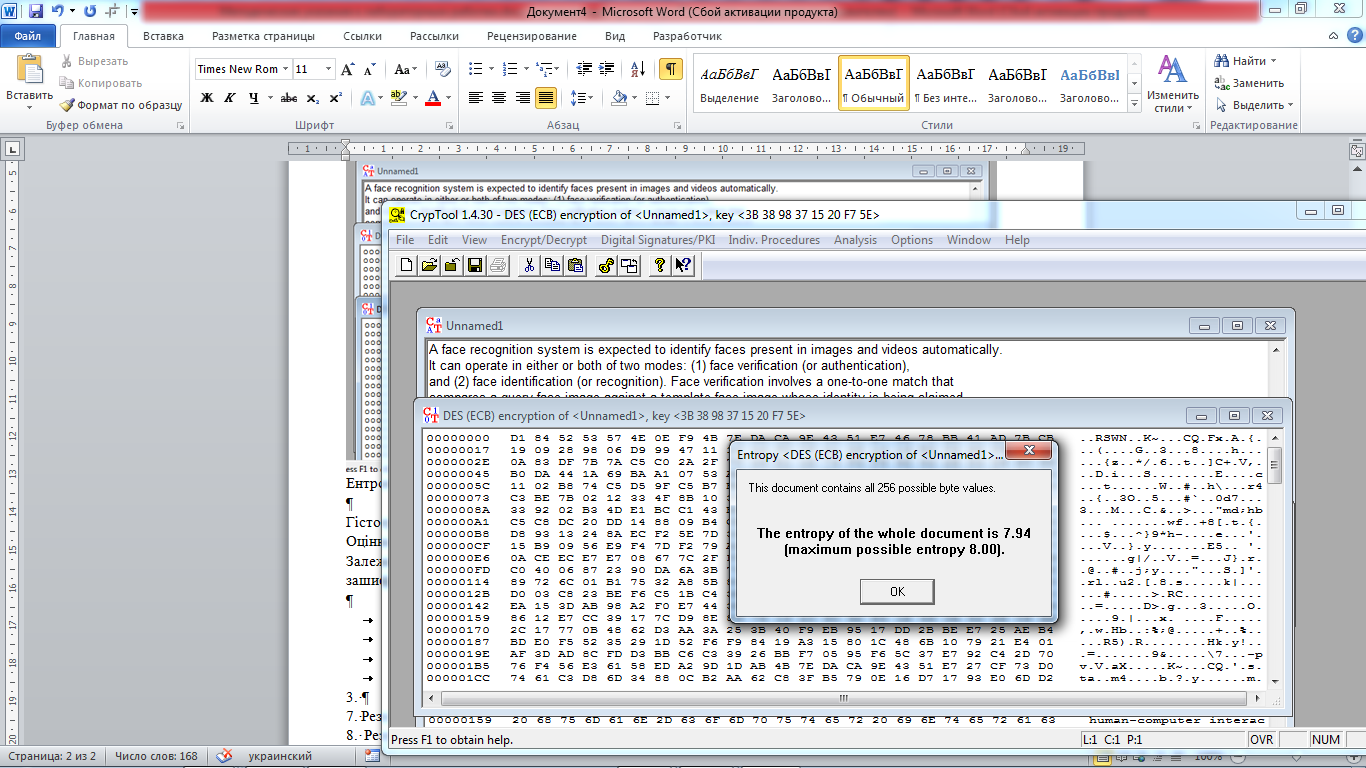
e. AES. key={59,56,152,55,21,32,247,94}

2. Результати шифрування и дешифрування текстового файлу в пакеті CrypTool з використаних алгоритмів

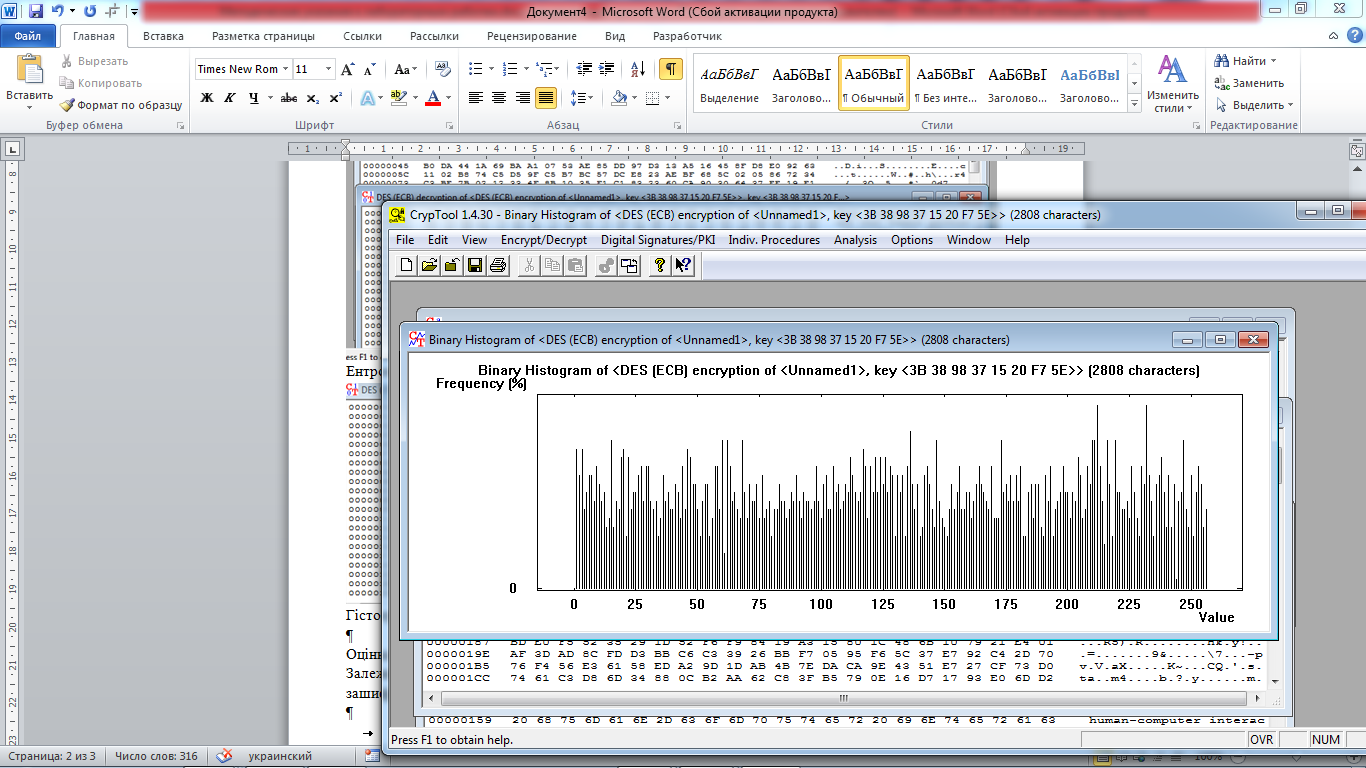
a. DES (ECB),



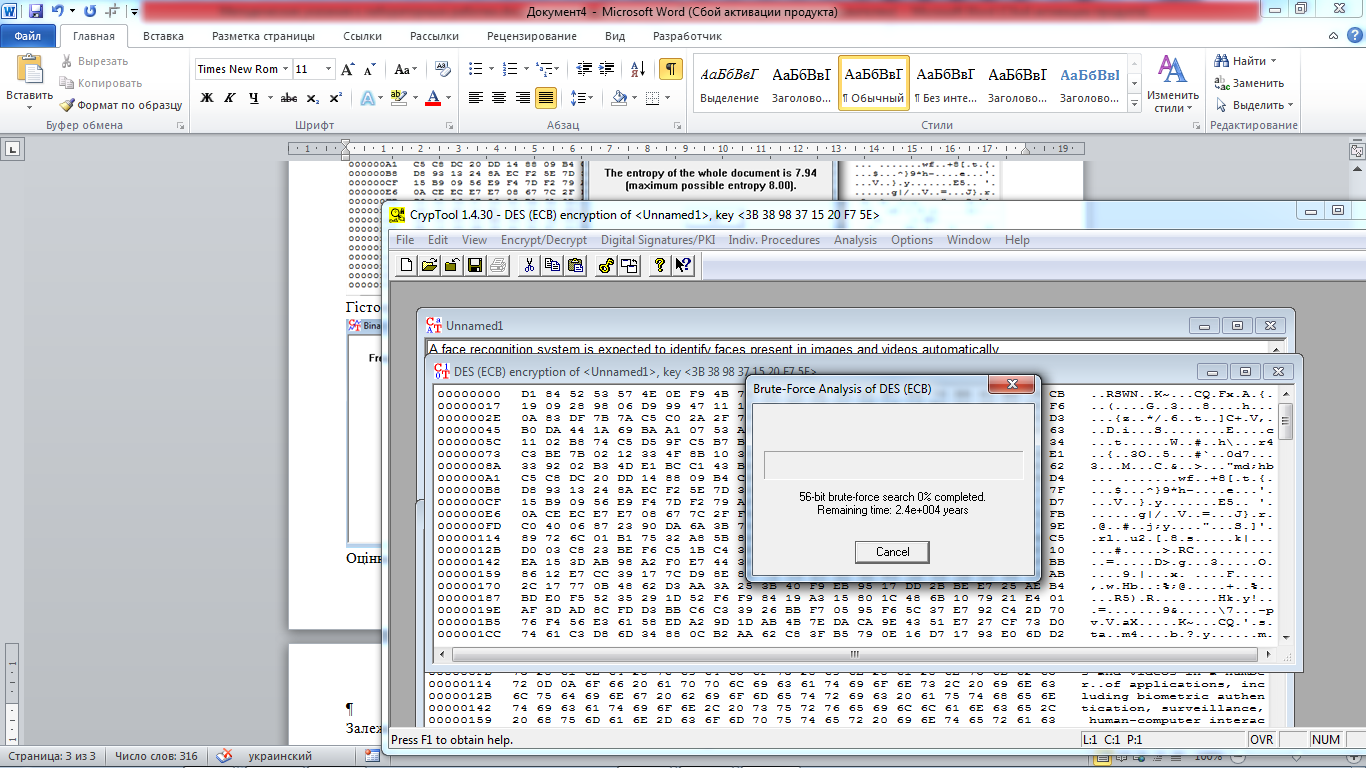
Ентропія, максимально можлива ентропія зашифрованих файлів:



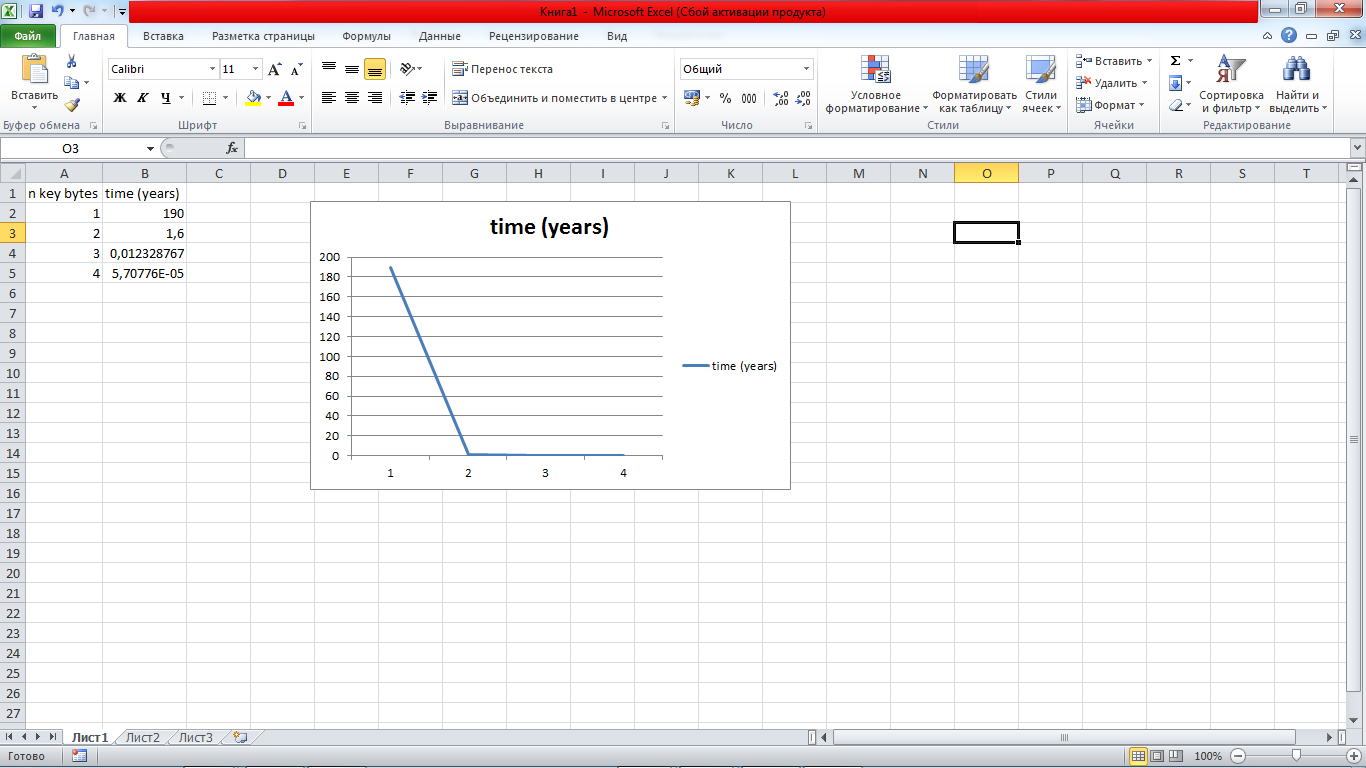
Гістограми розподілу частот символів зашифрованих файлів:



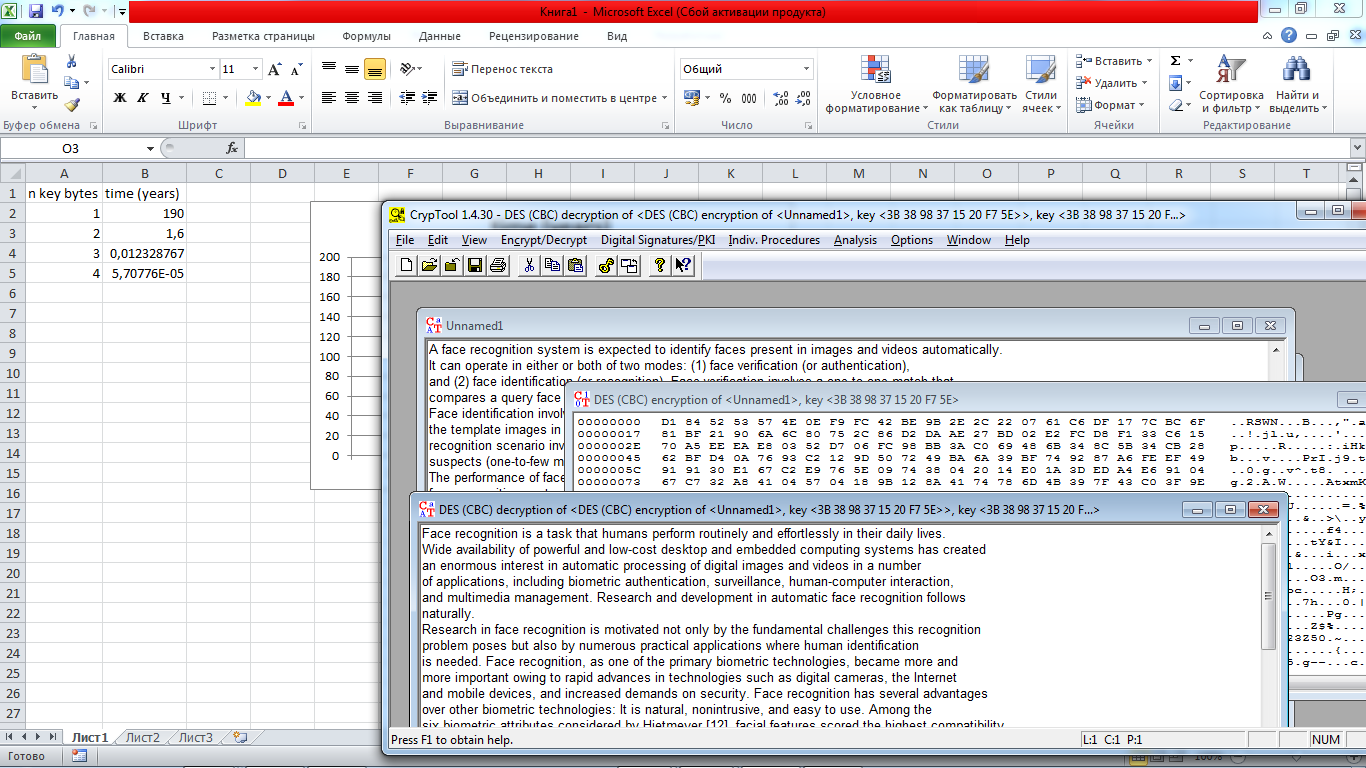
Оцінка часу для взлому зашифрованих фалів методом "грубої сили":



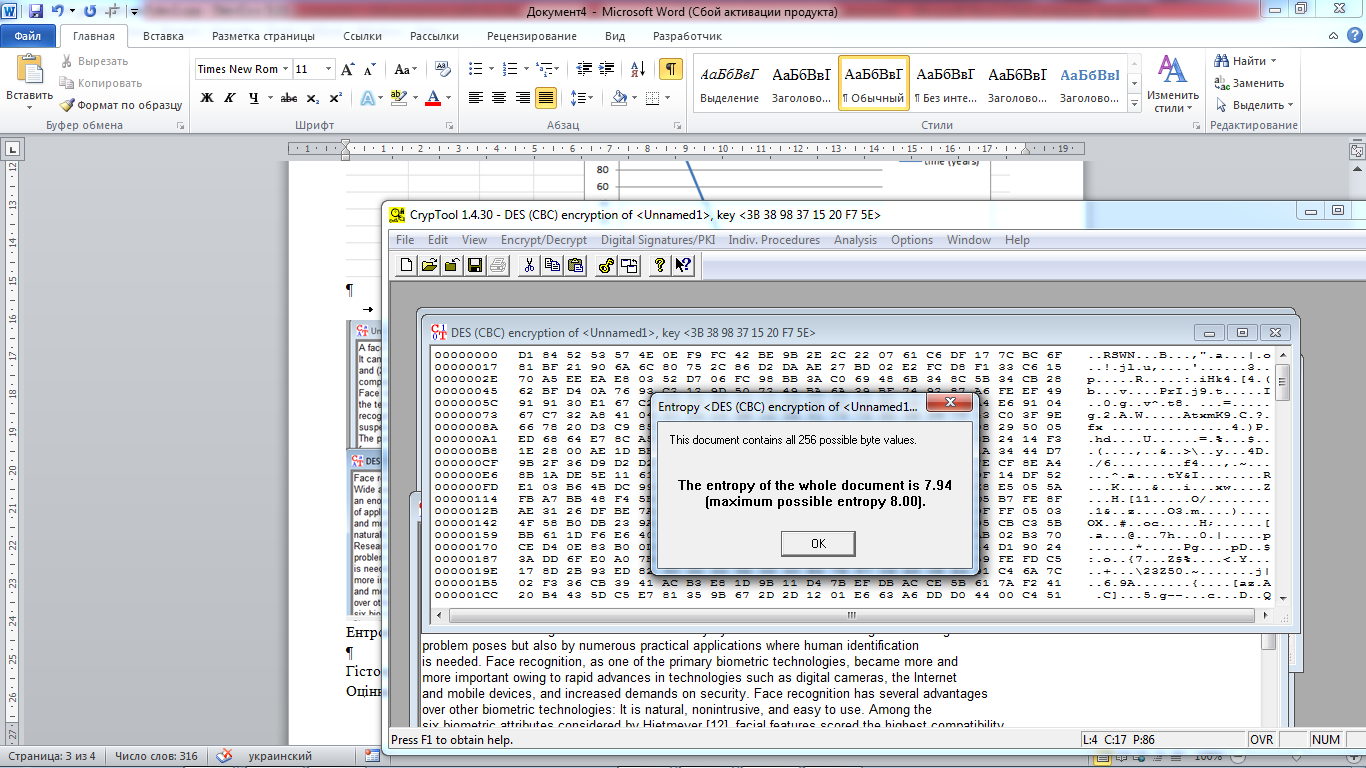
Залежність тривалості взлому досліджуваних шифрів від довжини ключа (результати криптоаналізу зашифрованих файлів при зменшеній довжині ключа):



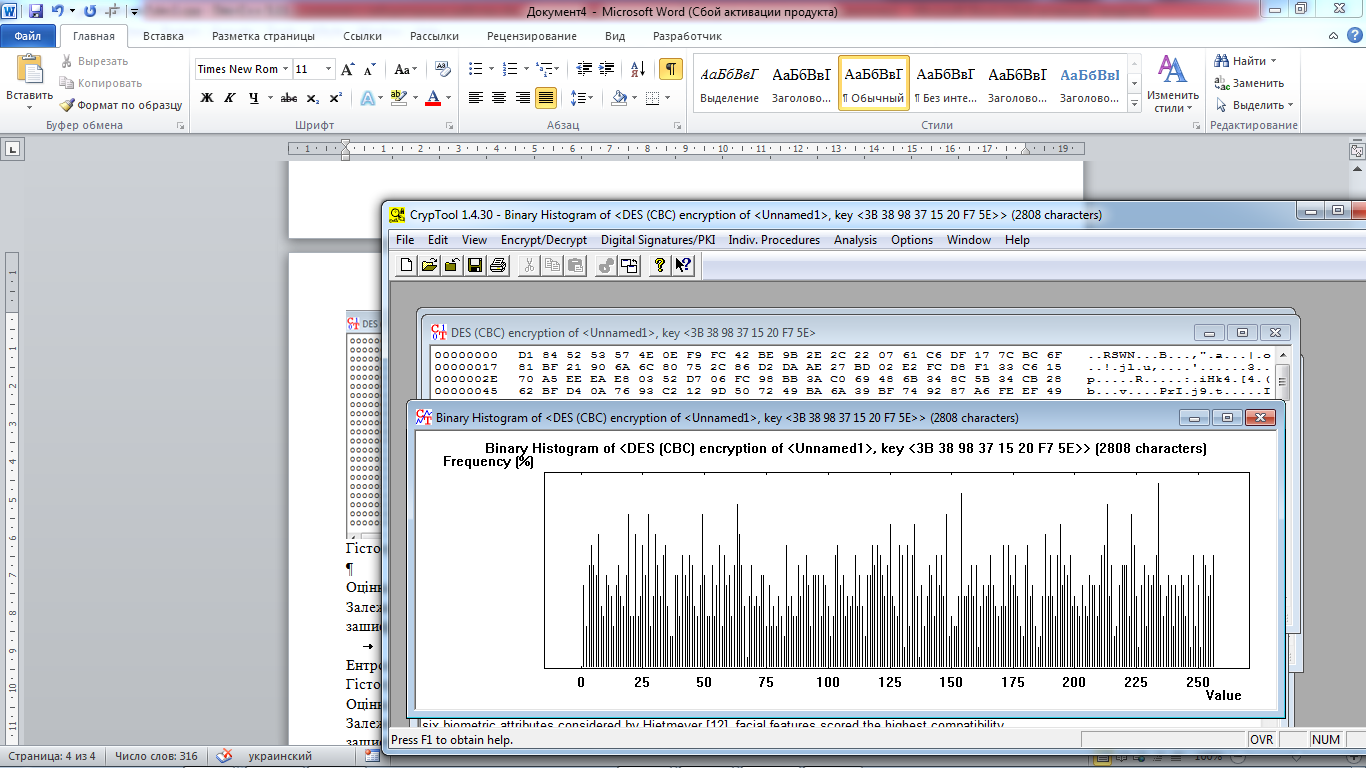
b. DES (CBC),



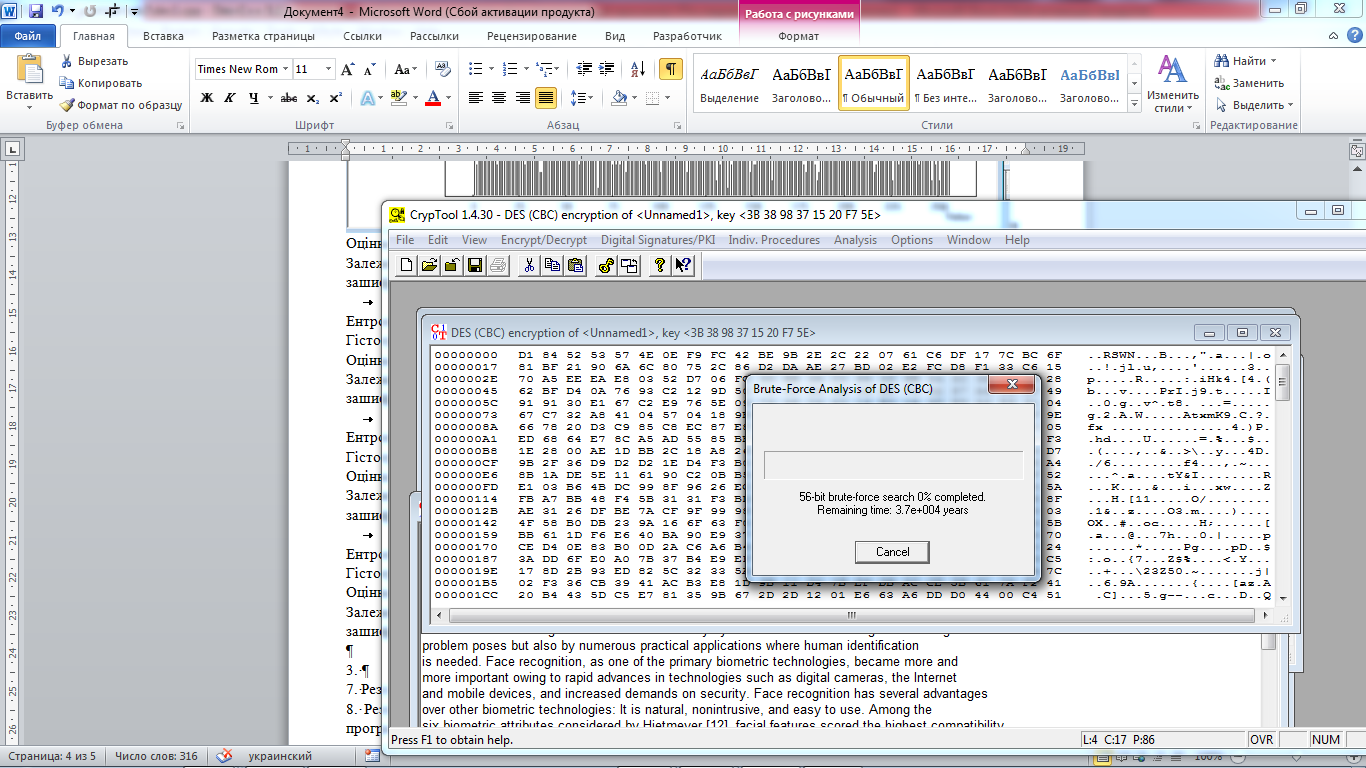
Ентропія, максимально можлива ентропія зашифрованих файлів:



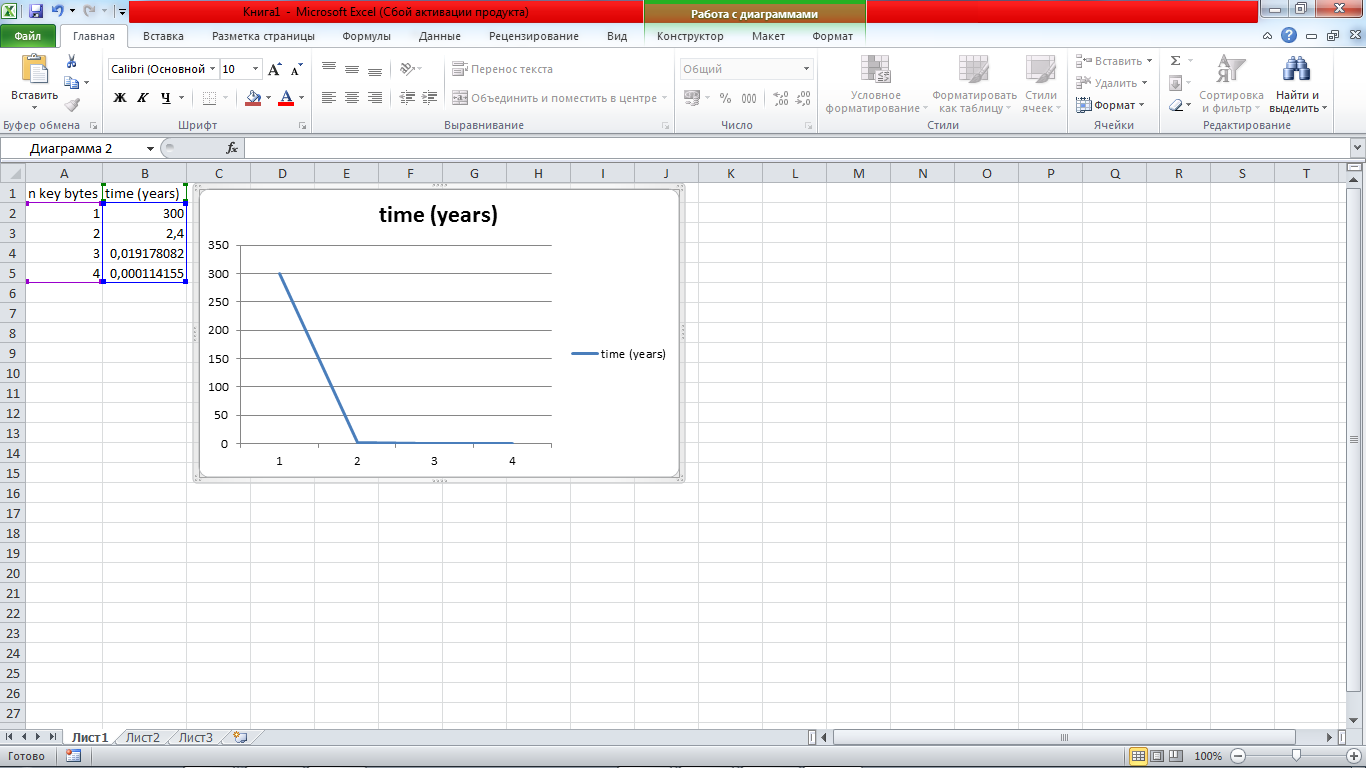
Гістограми розподілу частот символів зашифрованих файлів:



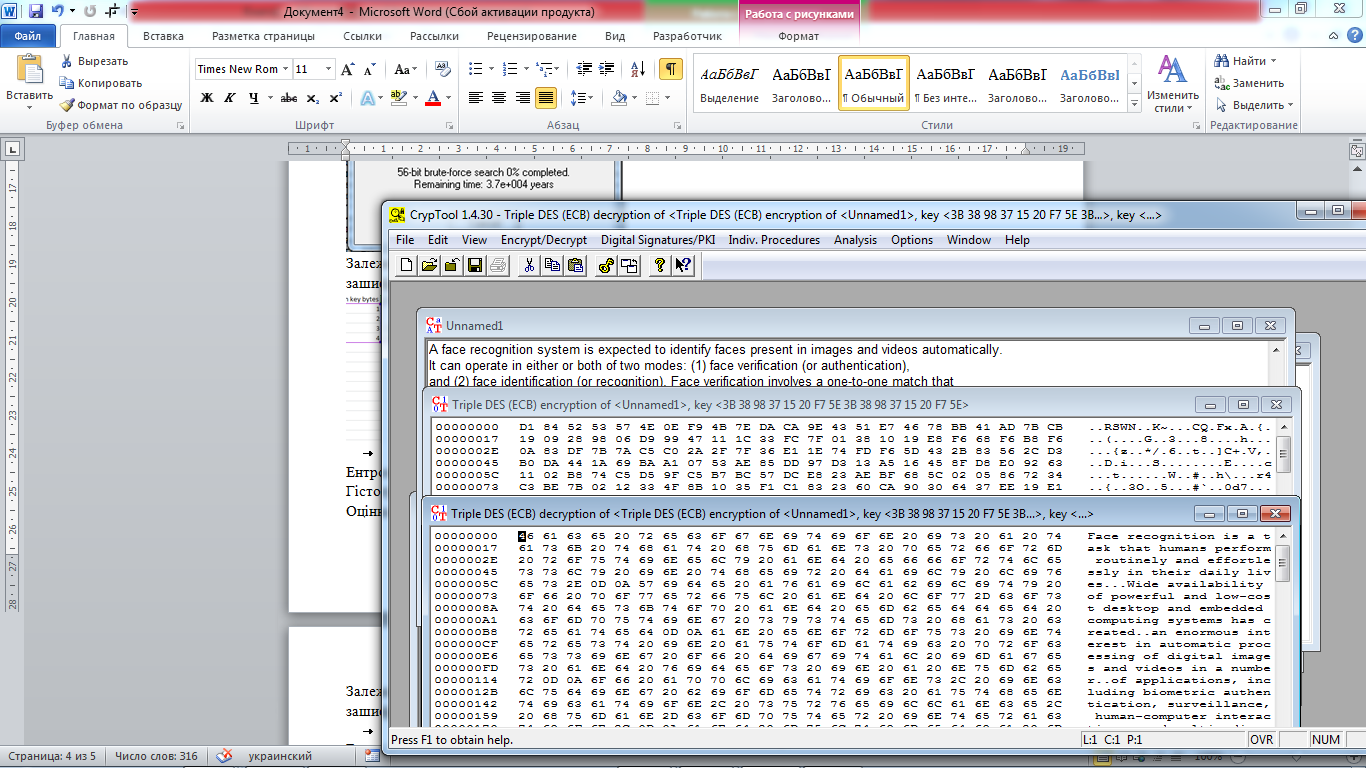
Оцінка часу для взлому зашифрованих фалів методом "грубої сили":



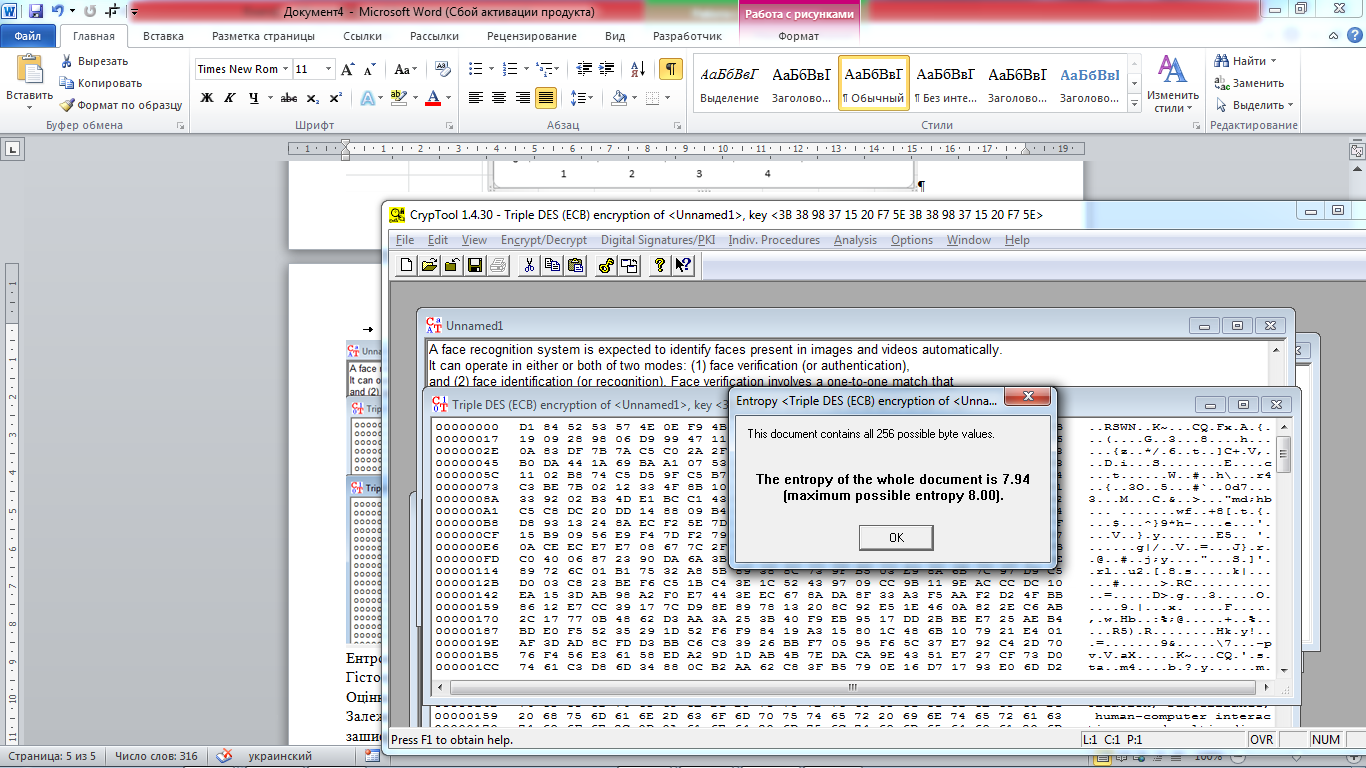
Залежність тривалості взлому досліджуваних шифрів від довжини ключа (результати криптоаналізу зашифрованих файлів при зменшеній довжині ключа):



c. TripleDES (ECB),



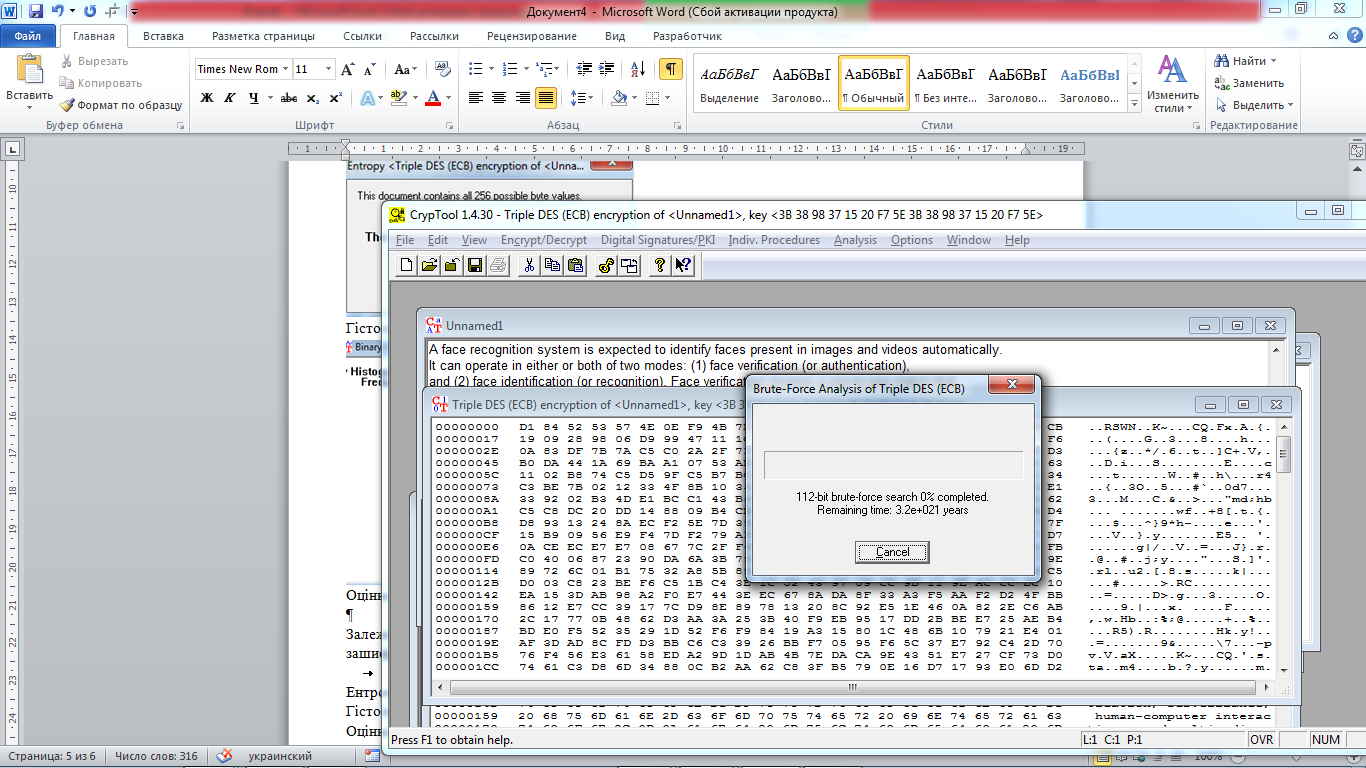
Ентропія, максимально можлива ентропія зашифрованих файлів:



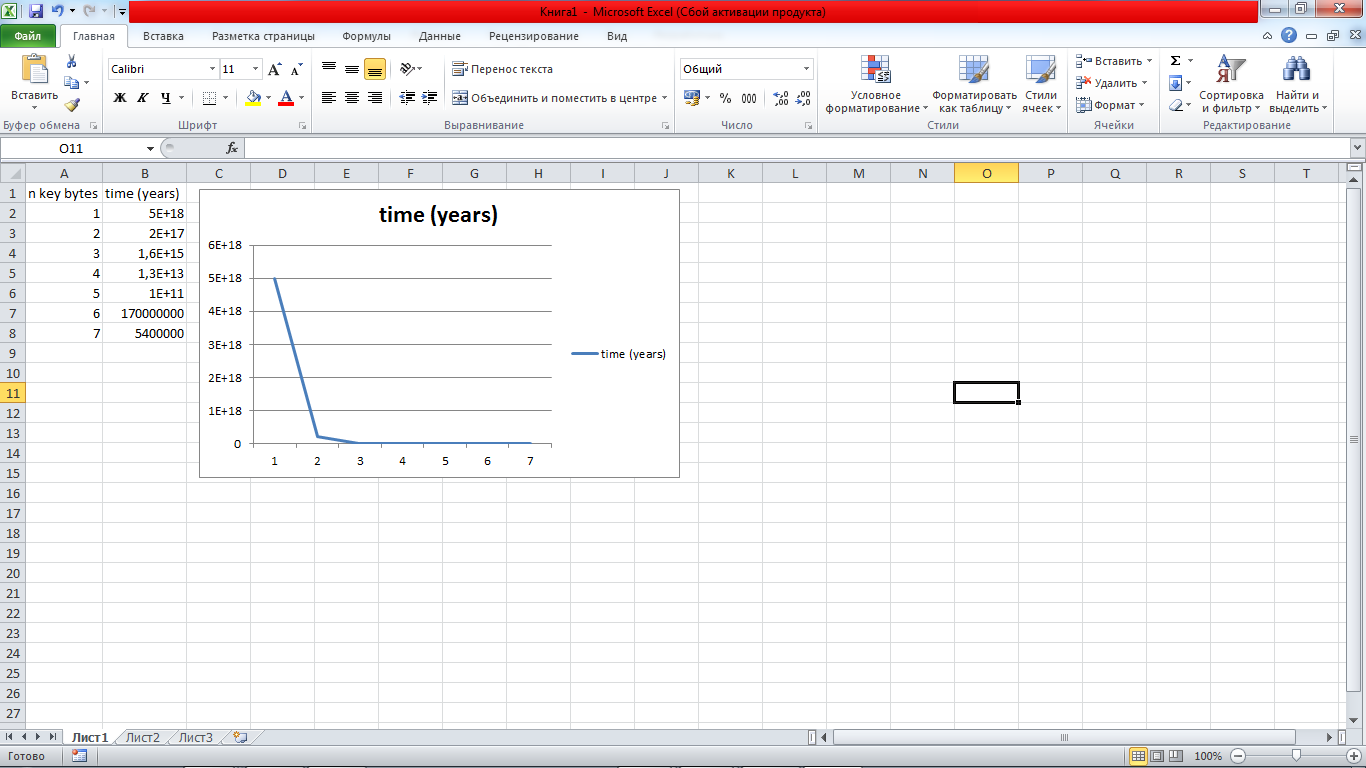
Гістограми розподілу частот символів зашифрованих файлів:



Оцінка часу для взлому зашифрованих фалів методом "грубої сили":



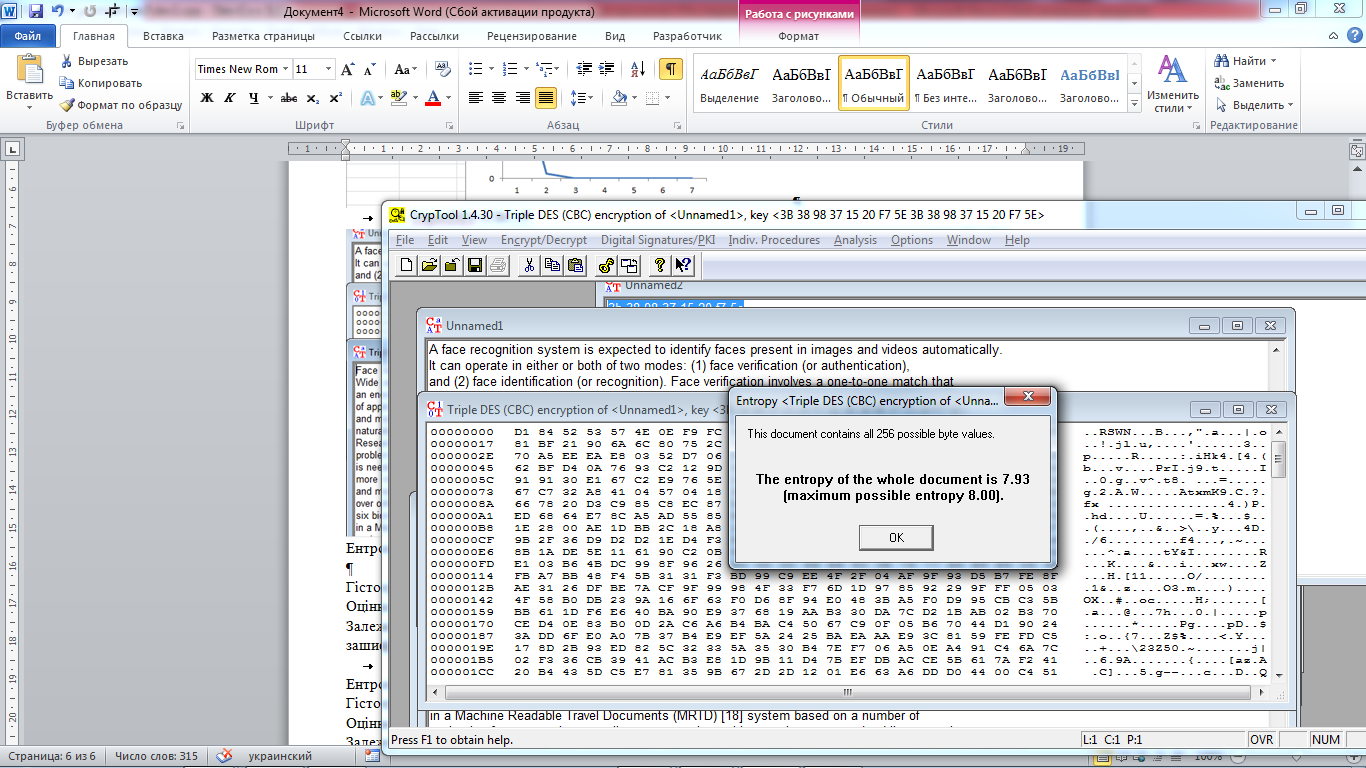
Залежність тривалості взлому досліджуваних шифрів від довжини ключа (результати криптоаналізу зашифрованих файлів при зменшеній довжині ключа):



d. TripleDES (CBC),



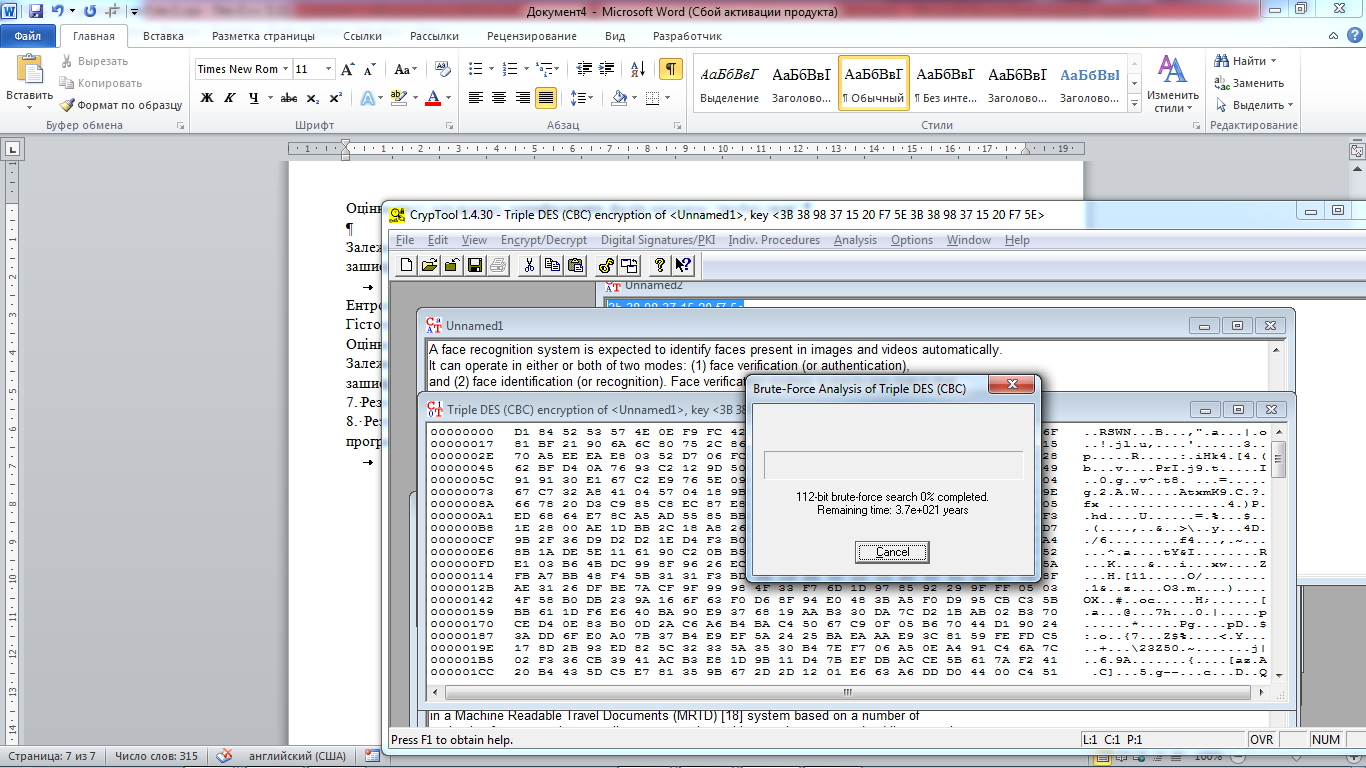
Ентропія, максимально можлива ентропія зашифрованих файлів:



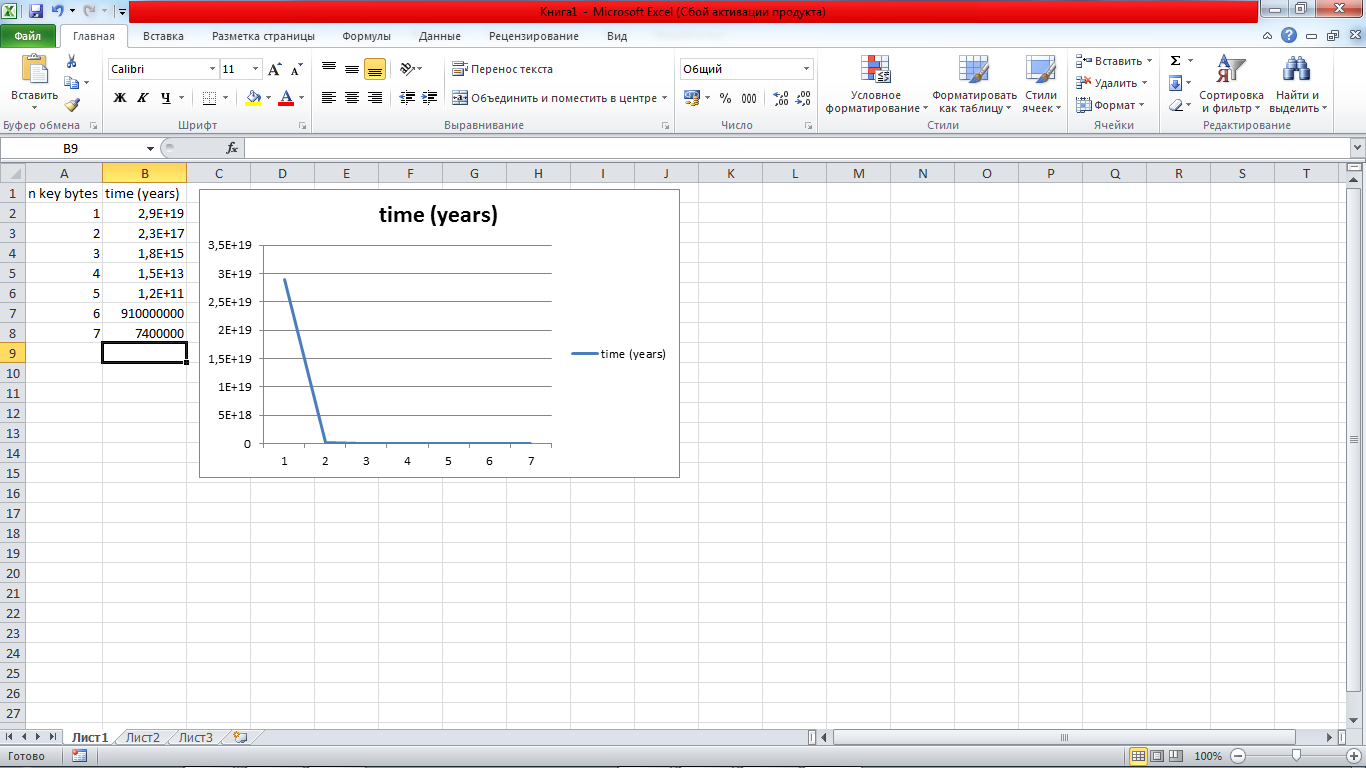
Гістограми розподілу частот символів зашифрованих файлів:



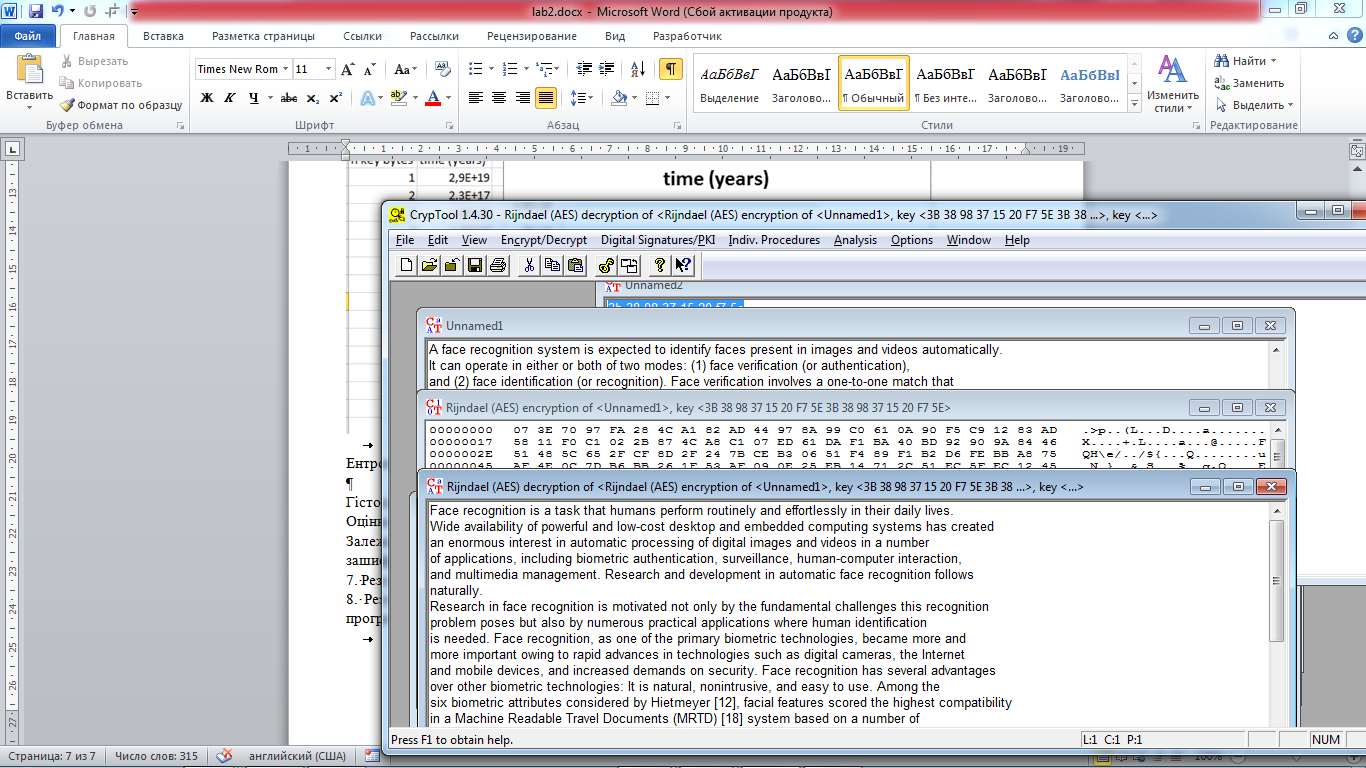
Оцінка часу для взлому зашифрованих фалів методом "грубої сили":



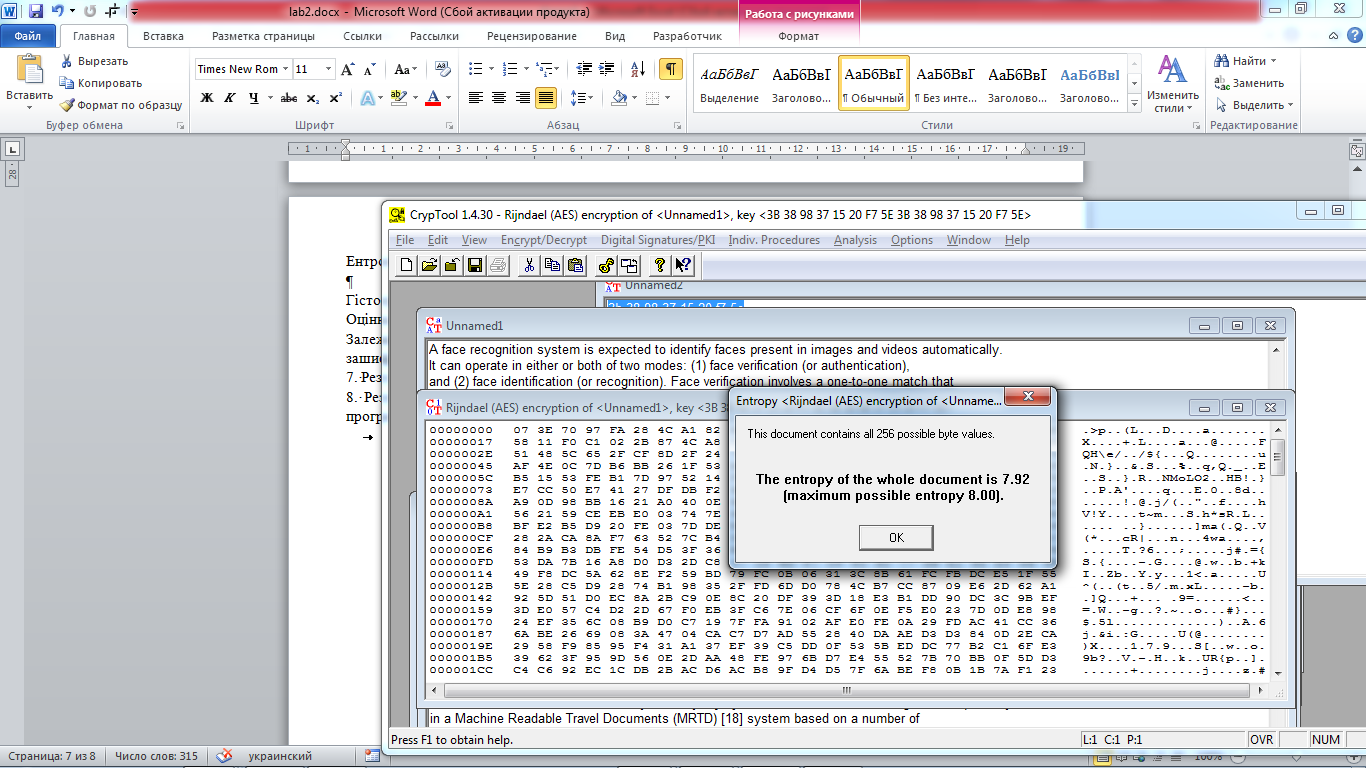
Залежність тривалості взлому досліджуваних шифрів від довжини ключа (результати криптоаналізу зашифрованих файлів при зменшеній довжині ключа):



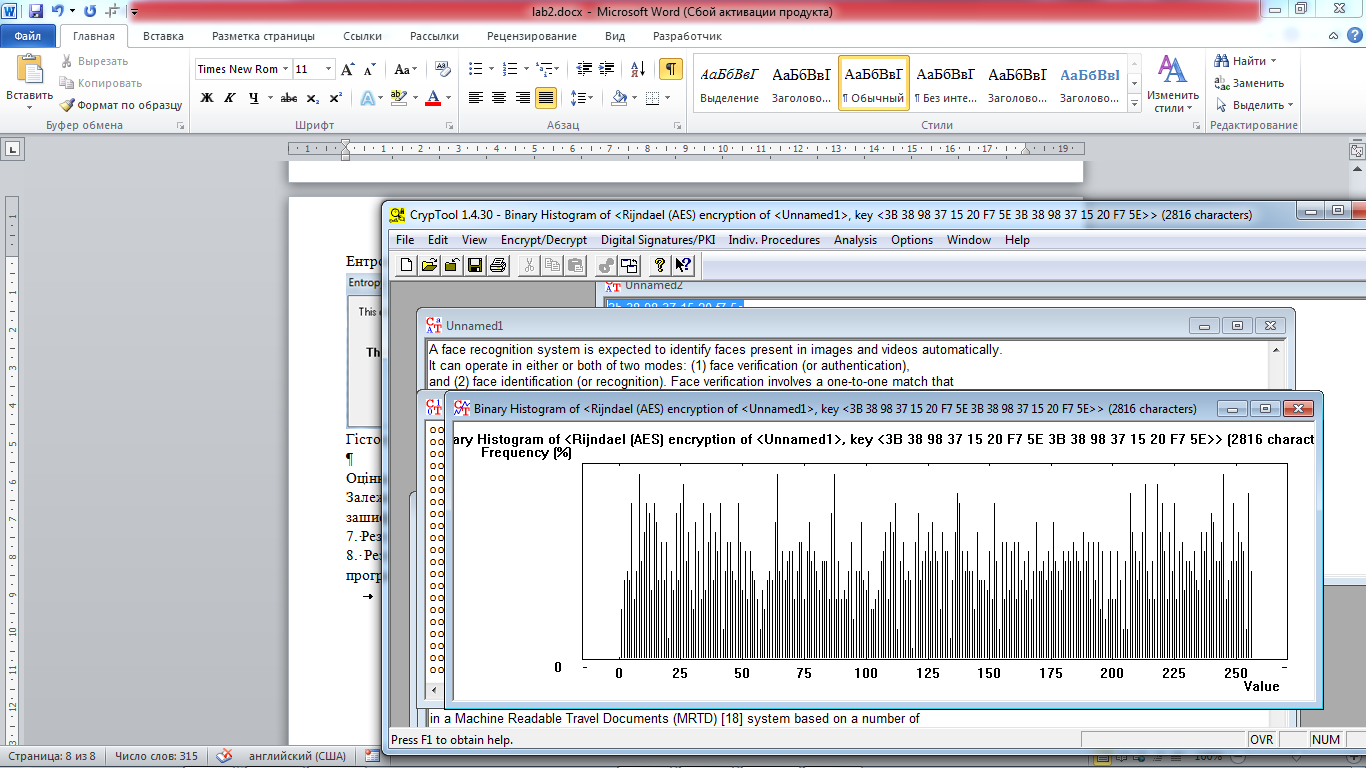
e. AES



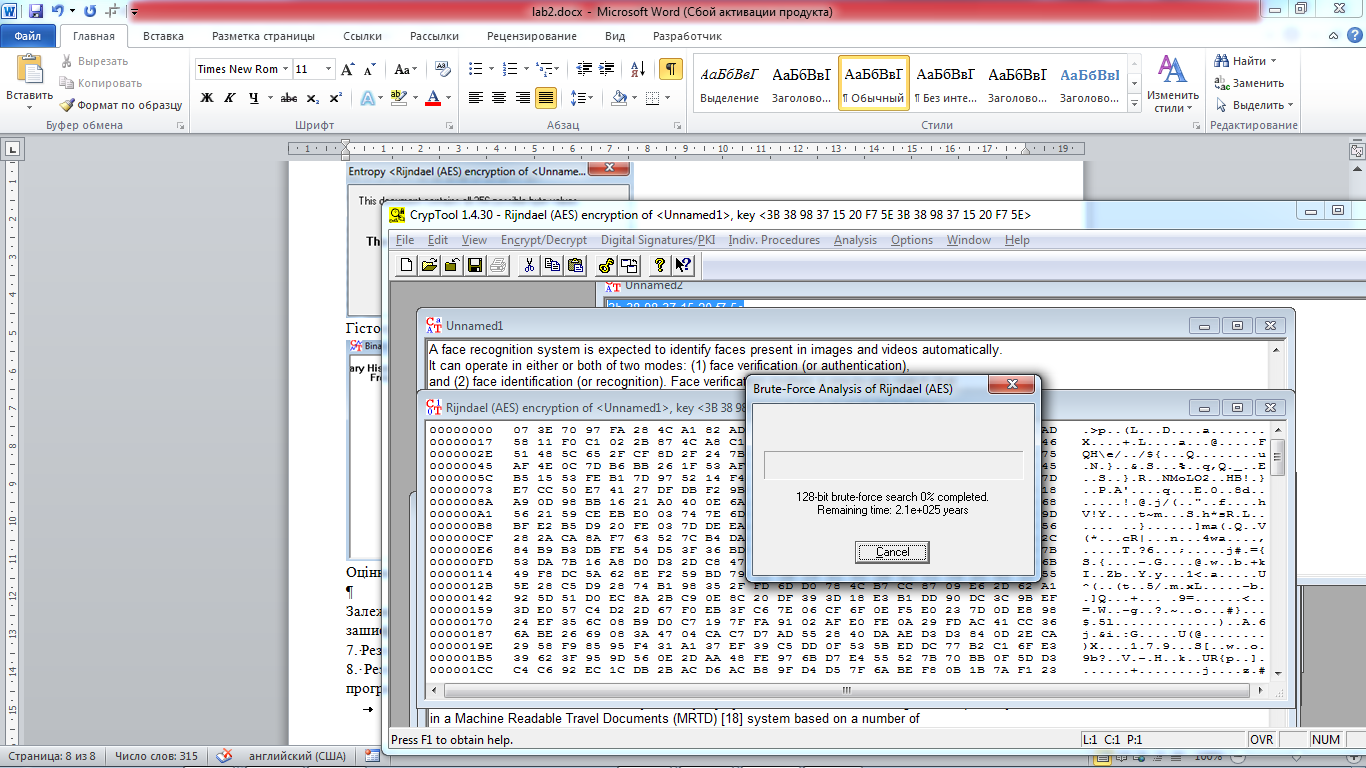
Ентропія, максимально можлива ентропія зашифрованих файлів:



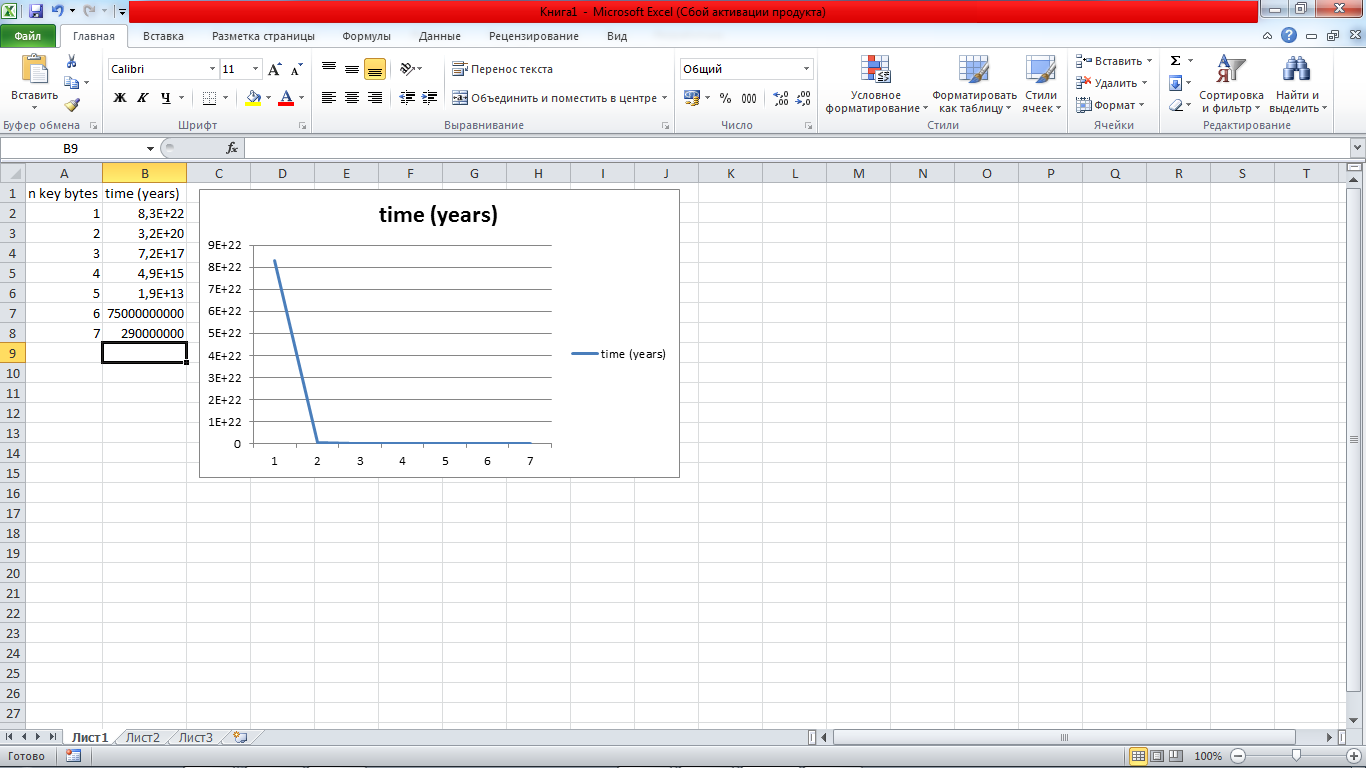
Гістограми розподілу частот символів зашифрованих файлів:



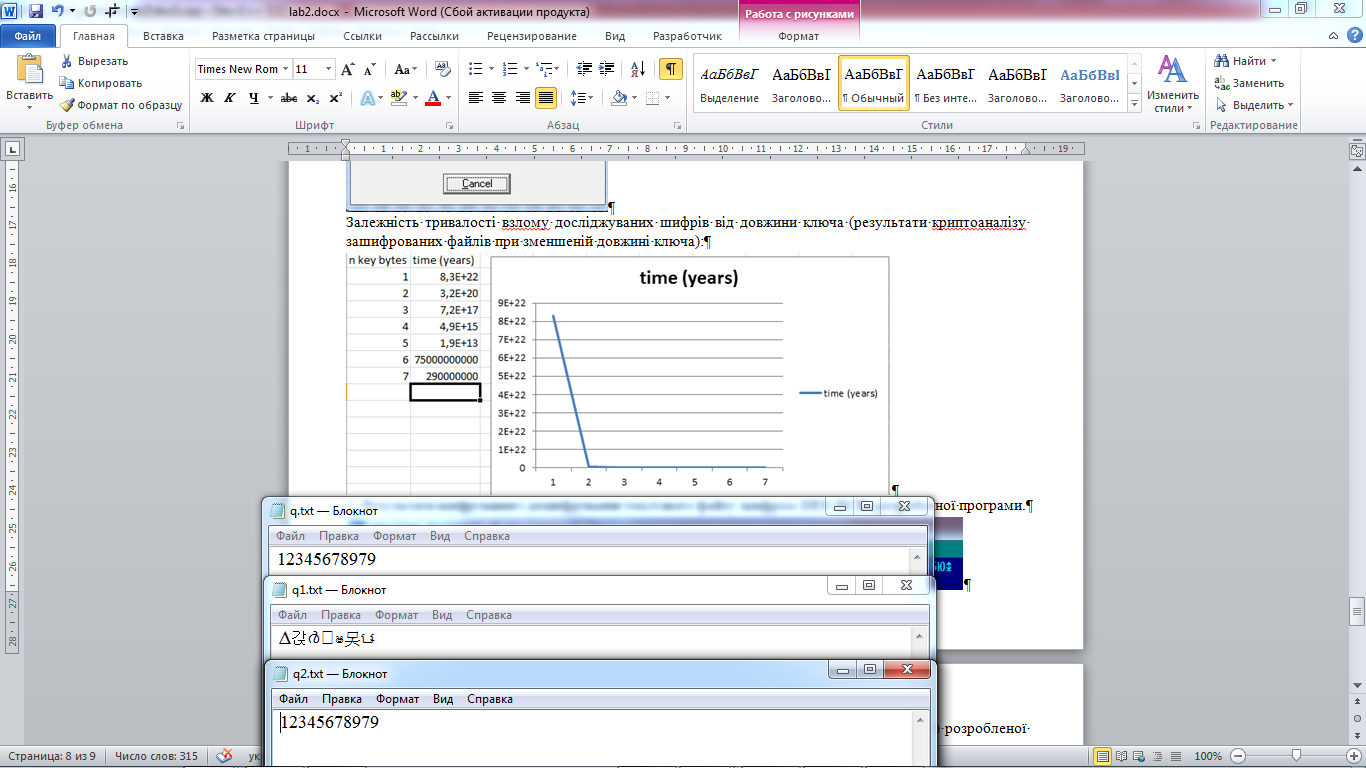
Оцінка часу для взлому зашифрованих фалів методом "грубої сили":

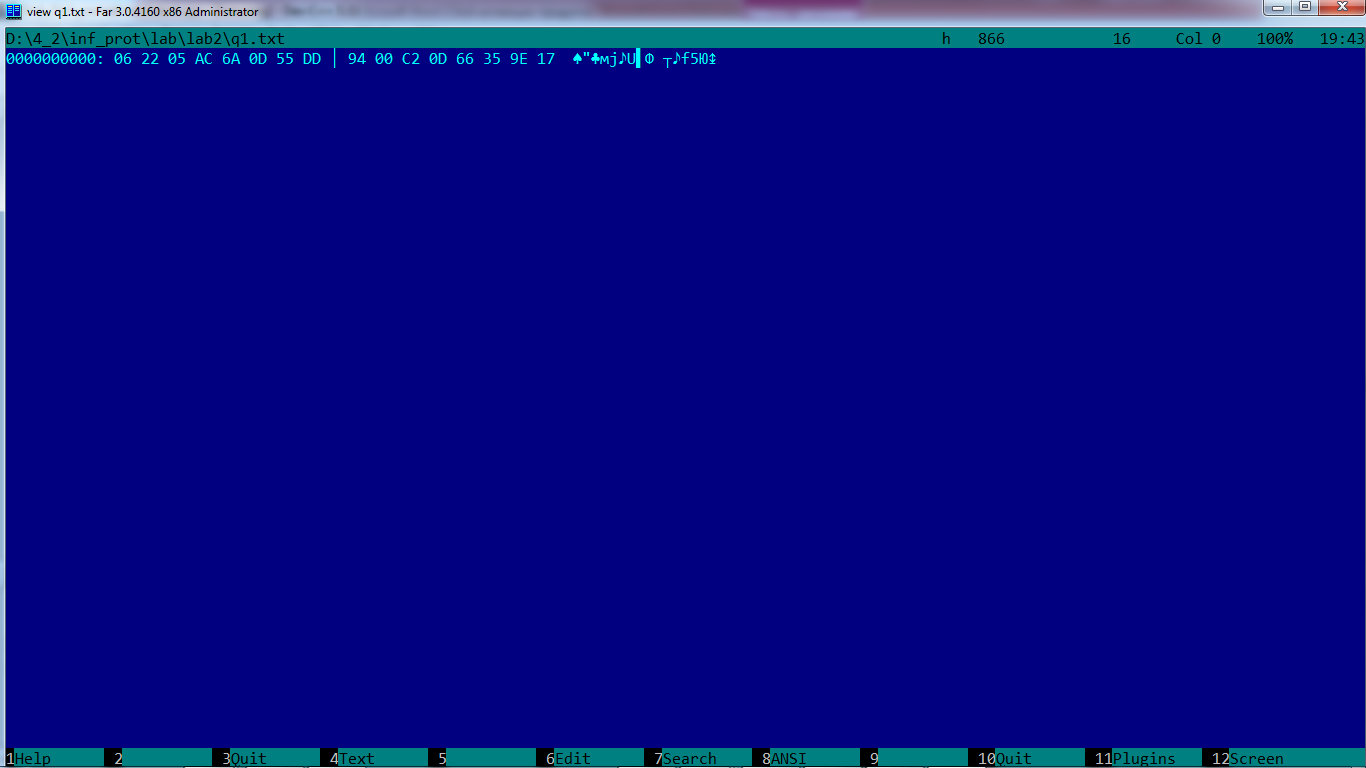


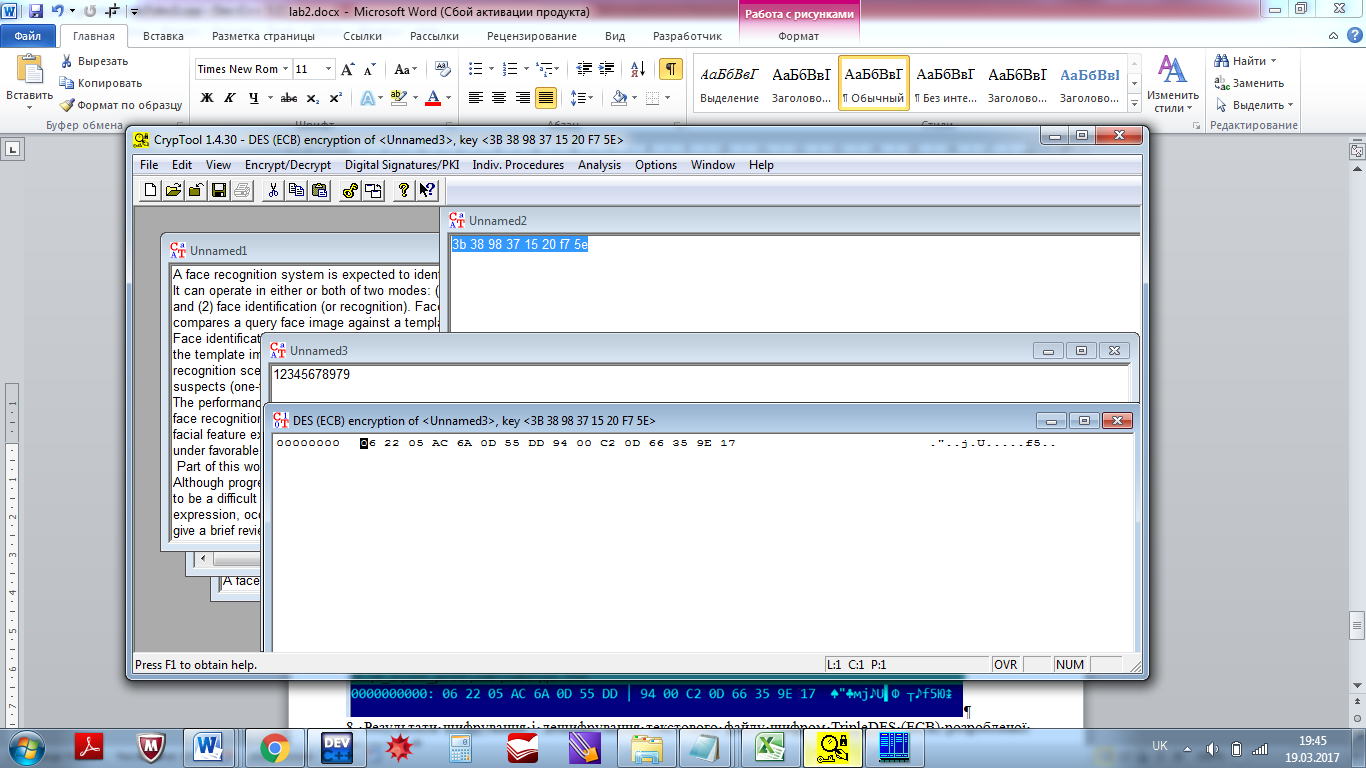
Залежність тривалості взлому досліджуваних шифрів від довжини ключа (результати криптоаналізу зашифрованих файлів при зменшеній довжині ключа):



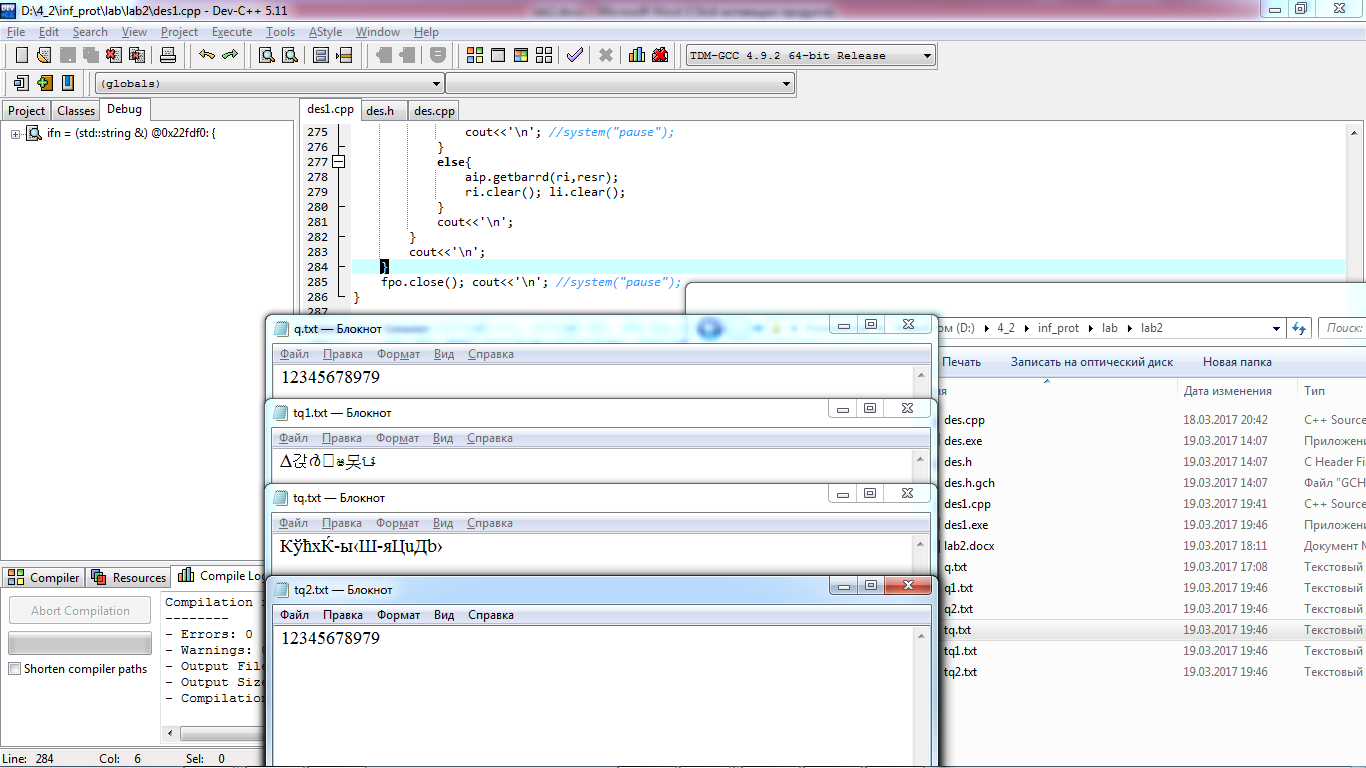
7. Результати шифрування і дешифрування текстового файлу шифром DES (ECB) розробленої програми.







8. Результати шифрування і дешифрування текстового файлу шифром TripleDES (ECB) розробленої програми.



Висновки: на лабораторній роботі розглядалися алгоритми симетричного шифрування і дешифрування даних з використанням шифрів DES (ECB), DES (CBC), TripleDES (ECB), TripleDES (CBC), AES за допомогою програмних засобів CrypTool та їх практична реалізація.

Додаток 1. Лістинг des1.cpp

#include<iostream>

#include<vector>

#include<string>

#include<fstream>

#include<cstdint>

#include"des.h"

#define bsz 8

using namespace std;

typedef unsigned short int usi;

typedef uint8\_t ui8;

struct b2b{

unsigned char b0:1;

unsigned char b1:1;

unsigned char b2:1;

unsigned char b3:1;

unsigned char b4:1;

unsigned char b5:1;

unsigned char b6:1;

unsigned char b7:1;

ui8 getb(ui8 i){

switch(i){

case 0: return b0;

case 1: return b1;

case 2: return b2;

case 3: return b3;

case 4: return b4;

case 5: return b5;

case 6: return b6;

case 7: return b7;

}

}

void putb(ui8 i,ui8 j){

switch(i){

case 0: b0=j; break;

case 1: b1=j; break;

case 2: b2=j; break;

case 3: b3=j; break;

case 4: b4=j; break;

case 5: b5=j; break;

case 6: b6=j; break;

case 7: b7=j; break;

}

}

};

union dtb{

unsigned char ucarr[bsz];

b2b lb[bsz/2];

b2b rb[bsz/2];

b2b barr[bsz];

void getarr(vector<bool>& bvecl,vector<bool>& bvecr){

for(ui8 k=0;k<bsz\*8;k++){

ui8 nbyte=k/8, nbit=k%8;

if(k<(ui8)56/2)

bvecl.push\_back(barr[nbyte].getb(nbit));

else if(k>=(ui8)56/2&&k<56)

bvecr.push\_back(barr[nbyte].getb(nbit));

}

}

void getbarr(vector<bool>& bvecl,vector<bool>& bvecr){

for(ui8 k=0;k<bsz\*8;k++){

ui8 nbyte=k/8, nbit=k%8;

if(k<(ui8)56/2)

barr[nbyte].putb(nbit,bvecl.at(k));

else if(k>=(ui8)56/2&&k<56)

barr[nbyte].putb(nbit,bvecr.at(k-(usi)56/2));

else

barr[nbyte].putb(nbit,0);

}

}

void getarrd(vector<bool>& bvecl,vector<bool>& bvecr){

for(ui8 k=0;k<bsz\*8;k++){

ui8 nbyte=k/8, nbit=k%8;

if(k<(ui8)bsz\*8/2)

bvecl.push\_back(barr[nbyte].getb(nbit));

else if(k>=bsz\*8/2)

bvecr.push\_back(barr[nbyte].getb(nbit));

}

}

void getbarrd(vector<bool>& bvecl,vector<bool>& bvecr){

for(ui8 k=0;k<bsz\*8;k++){

ui8 nbyte=k/8, nbit=k%8;

if(k<(ui8)bsz\*8/2)

barr[nbyte].putb(nbit,bvecl.at(k));

else if(k>=(ui8)bsz\*8/2)

barr[nbyte].putb(nbit,bvecr.at(k-(usi)bsz\*8/2));

}

}

void invb(){

for(ui8 i=0;i<bsz;i++){

for(ui8 j=0;j<8/2;j++){

ui8 u8tmp=barr[i].getb(7-j);

barr[i].putb(7-j,barr[i].getb(j));

barr[i].putb(j,u8tmp);

}

}

}

};

bool gff(string& ifn,string& dt,ifstream& fpi){

usi cnt=0;

fpi.open(ifn.c\_str(),ios::binary);

if(!fpi.is\_open()) {cout<<"\nerr: gff no input file"; system("pause"); return 0;}

else{

while(1){

unsigned char ch='\0';

fpi>>noskipws>>ch;

if(fpi.eof()) break;

dt+=ch;

//cout<<"\nch="<<ch<<" fdt.ucarr[cnt-1]="<<fdt.ucarr[cnt];

}

}

fpi.close();

return 1;

}

bool ptf(string& ofn,dtb& fdt,ofstream& fpo,ui8 fl){

for(ui8 i=0;i<bsz;i++){

if(fl==2){fpo<<fdt.ucarr[i]; cout<<hex<<(usi)fdt.ucarr[i]<<' ';}

cout<<' ';

for(ui8 j=0;j<8;j++){

if(fl==1) cout<<(usi)fdt.barr[i].getb(7-j);

else if(fl==0) cout<<(usi)fdt.barr[i].getb(j);

}

}

}

void desenc(string& srs, string& dst, dtb key,vector<dtb>& ksh){

dtb fdt; bool strt=true,strt1=true; ifstream fpi; ofstream fpo; ui8 gffres;

fpo.open(dst.c\_str(),ios::binary);

if(!fpo.is\_open()) {cout<<"\nerr:ptff no output file"; system("pause"); return;}

cout<<'\n';

cout<<"\nkey="; ptf(dst,key,fpo,1);

string dt; gff(srs,dt,fpi);

while(dt.length()!=0){

for(ui8 i=0;i<8;i++){

if(dt.length()>0){

cout<<"\nsizeof(dt)="<<dt.length()<<' '<<dt<<'\n'; //system("pause");

fdt.ucarr[i]=(unsigned char)((dt.c\_str())[0]);

cout<<"\ndt="<<hex<<(usi)fdt.ucarr[i]<<dec;

dt.erase(dt.begin());

}

else fdt.ucarr[i]=0;

}

dtb apc1,apc2,aip;

cout<<"\ndata="; ptf(dst,fdt,fpo,1); cout<<'\n';

for(ui8 i=0;i<sizeof(pc1);i++){

ui8 nbyte=i/8, nbit=i%8, nbyte1=(pc1[i]-1)/8,nbit1=(pc1[i]-1)%8;

apc1.barr[nbyte].putb(nbit,key.barr[nbyte1].getb(7-nbit1));

}

for(ui8 i=0;i<sizeof(ip);i++){

ui8 nbyte=i/8, nbit=i%8, nbyte1=(ip[i]-1)/8,nbit1=(ip[i]-1)%8;

aip.barr[nbyte].putb(nbit,fdt.barr[nbyte1].getb(7-nbit1));

}

for(ui8 i=0;i<sizeof(itrol);i++){//round;

for(ui8 j=0;j<itrol[i];j++){//num of shifts;

vector<bool> bvecl,bvecr; apc1.getarr(bvecl,bvecr);

/\*cout<<'\n'; for(ui8 l=0;l<bvecl.size();l++) cout<<bvecl.at(l); cout<<'\t';

for(ui8 l=0;l<bvecr.size();l++) cout<<bvecr.at(l); cout<<'\n'; system("pause");\*/

ui8 u8tmpl=bvecl.at(0), u8tmpr=bvecr.at(0);

for(ui8 k=0;k<bvecl.size()-1;k++){

bvecl.at(k)=bvecl.at(k+1);

bvecr.at(k)=bvecr.at(k+1);

}

bvecl.at(bvecl.size()-1)=u8tmpl; bvecr.at(bvecr.size()-1)=u8tmpr;

apc1.getbarr(bvecl,bvecr);

}

for(ui8 k=0;k<sizeof(pc2);k++){

ui8 nbyte=k/8, nbit=k%8, nbyte1=(pc2[k]-1)/8,nbit1=(pc2[k]-1)%8;

apc2.barr[nbyte].putb(nbit,apc1.barr[nbyte1].getb(nbit1));

}//apc2 is round key; apc1 is Ci, Di;

ksh.push\_back(apc2);

cout<<"\nkeysh="; ptf(dst,apc2,fpo,0);

vector<bool> li,ri; aip.getarrd(li,ri); dtb aep,asb,app;

//cout<<'\n'; for(ui8 l=0;l<li.size();l++) cout<<li.at(l); cout<<'\t';

//for(ui8 l=0;l<ri.size();l++) cout<<ri.at(l); cout<<'\n'; system("pause");

for(ui8 j=0;j<sizeof(ep);j++){

ui8 nbyte=j/8, nbit=j%8, nbyte1=(ep[j]-1)/8,nbit1=(ep[j]-1)%8;

aep.barr[nbyte].putb(nbit,ri.at(ep[j]-1));

}

cout<<"\naep="; ptf(dst,aep,fpo,0); cout<<'\n';

for(ui8 j=0;j<sizeof(ep);j++){//xor ki;

ui8 nbyte=j/8, nbit=j%8;

aep.barr[nbyte].putb(nbit,(apc2.barr[nbyte].getb(nbit)^aep.barr[nbyte].getb(nbit)));

}

cout<<"\naep xor ki ="; ptf(dst,aep,fpo,0);

//vector<bool> sbvl,sbvr; aep.getarrd(sbvl,sbvr);

//for(usi j=0;j<sbvr.size();j++) sbvl.push\_back(sbvr.at(j));

//cout<<"\n\t"; for(usi j=0;j<sbvl.size();j++) cout<<sbvl.at(j); cout<<' ';

for(ui8 j=0,scnt=0;j<sizeof(ep);j+=6,scnt++){

ui8 nbyte=j/8, nbit=j%8;

ui8 nbyte1=(j+1)/8, nbit1=(j+1)%8;

ui8 nbyte2=(j+2)/8, nbit2=(j+2)%8;

ui8 nbyte3=(j+3)/8, nbit3=(j+3)%8;

ui8 nbyte4=(j+4)/8, nbit4=(j+4)%8;

ui8 nbyte5=(j+5)/8, nbit5=(j+5)%8;

unsigned char uctmp;

/\*cout<<"x,y="<<aep.barr[nbyte].getb(nbit)\*2+aep.barr[nbyte5].getb(nbit5)<<' '

<<aep.barr[nbyte1].getb(nbit1)\*2\*2\*2+aep.barr[nbyte2].getb(nbit2)\*2\*2+

aep.barr[nbyte3].getb(nbit3)\*2+aep.barr[nbyte4].getb(nbit4);

cout<<'\n'<<(usi)aep.barr[nbyte].getb(nbit)<<(usi)aep.barr[nbyte1].getb(nbit1)<<(usi)aep.barr[nbyte2].getb(nbit2)

<<(usi)aep.barr[nbyte3].getb(nbit3)<<(usi)aep.barr[nbyte4].getb(nbit4)<<(usi)aep.barr[nbyte5].getb(nbit5)<<'\n';\*/

switch(scnt){

case 0:

uctmp=sb1[aep.barr[nbyte].getb(nbit)\*2+aep.barr[nbyte5].getb(nbit5)]

[aep.barr[nbyte1].getb(nbit1)\*2\*2\*2+aep.barr[nbyte2].getb(nbit2)\*2\*2+

aep.barr[nbyte3].getb(nbit3)\*2+aep.barr[nbyte4].getb(nbit4)];

break;

case 1:

uctmp=sb2[aep.barr[nbyte].getb(nbit)\*2+aep.barr[nbyte5].getb(nbit5)]

[aep.barr[nbyte1].getb(nbit1)\*2\*2\*2+aep.barr[nbyte2].getb(nbit2)\*2\*2+

aep.barr[nbyte3].getb(nbit3)\*2+aep.barr[nbyte4].getb(nbit4)];

break;

case 2:

uctmp=sb3[aep.barr[nbyte].getb(nbit)\*2+aep.barr[nbyte5].getb(nbit5)]

[aep.barr[nbyte1].getb(nbit1)\*2\*2\*2+aep.barr[nbyte2].getb(nbit2)\*2\*2+

aep.barr[nbyte3].getb(nbit3)\*2+aep.barr[nbyte4].getb(nbit4)];

break;

case 3:

uctmp=sb4[aep.barr[nbyte].getb(nbit)\*2+aep.barr[nbyte5].getb(nbit5)]

[aep.barr[nbyte1].getb(nbit1)\*2\*2\*2+aep.barr[nbyte2].getb(nbit2)\*2\*2+

aep.barr[nbyte3].getb(nbit3)\*2+aep.barr[nbyte4].getb(nbit4)];

break;

case 4:

uctmp=sb5[aep.barr[nbyte].getb(nbit)\*2+aep.barr[nbyte5].getb(nbit5)]

[aep.barr[nbyte1].getb(nbit1)\*2\*2\*2+aep.barr[nbyte2].getb(nbit2)\*2\*2+

aep.barr[nbyte3].getb(nbit3)\*2+aep.barr[nbyte4].getb(nbit4)];

break;

case 5:

uctmp=sb6[aep.barr[nbyte].getb(nbit)\*2+aep.barr[nbyte5].getb(nbit5)]

[aep.barr[nbyte1].getb(nbit1)\*2\*2\*2+aep.barr[nbyte2].getb(nbit2)\*2\*2+

aep.barr[nbyte3].getb(nbit3)\*2+aep.barr[nbyte4].getb(nbit4)];

break;

case 6:

uctmp=sb7[aep.barr[nbyte].getb(nbit)\*2+aep.barr[nbyte5].getb(nbit5)]

[aep.barr[nbyte1].getb(nbit1)\*2\*2\*2+aep.barr[nbyte2].getb(nbit2)\*2\*2+

aep.barr[nbyte3].getb(nbit3)\*2+aep.barr[nbyte4].getb(nbit4)];

break;

case 7:

uctmp=sb8[aep.barr[nbyte].getb(nbit)\*2+aep.barr[nbyte5].getb(nbit5)]

[aep.barr[nbyte1].getb(nbit1)\*2\*2\*2+aep.barr[nbyte2].getb(nbit2)\*2\*2+

aep.barr[nbyte3].getb(nbit3)\*2+aep.barr[nbyte4].getb(nbit4)];

break;

}

if(scnt%2==0){

asb.ucarr[scnt/2]=uctmp<<4;

}

else{

asb.ucarr[scnt/2]+=uctmp;

}

cout<<"\nuctmp="<<(usi)uctmp<<' '<<hex<<(usi)uctmp<<dec;;

}

cout<<"\nasb="; ptf(dst,asb,fpo,1);

for(ui8 j=0;j<sizeof(pp);j++){

ui8 nbyte=j/8, nbit=j%8, nbyte1=(pp[j]-1)/8,nbit1=(pp[j]-1)%8;

app.barr[nbyte].putb(nbit,asb.barr[nbyte1].getb(7-nbit1));

}

cout<<"\napp="; ptf(dst,app,fpo,0);

//cout<<'\n'; for(ui8 j=0;j<li.size();j++) cout<<li.at(j); cout<<' '; //system("pause");

vector<bool> resr;

for(ui8 j=0;j<li.size();j++) resr.push\_back(app.barr[j/8].getb(j%8)^li.at(j));

cout<<"\nnli="; for(ui8 j=0;j<li.size();j++) cout<<ri.at(j);

cout<<"\nnri="; for(ui8 j=0;j<resr.size();j++) cout<<resr.at(j);

//cout<<"\nsizrof(ri)="<<ri.size()<<" sizeof(resr)="<<resr.size(); cout<<' '; system("pause");

if(i==15){

aip.getbarrd(resr,ri); dtb aip1;

cout<<"\nbinv="; ptf(dst,aip,fpo,0);

for(ui8 j=0;j<sizeof(ip1);j++){

ui8 nbyte=j/8, nbit=j%8, nbyte1=(ip1[j]-1)/8,nbit1=(ip1[j]-1)%8;

aip1.barr[nbyte].putb(nbit,aip.barr[nbyte1].getb(nbit1));

}

aip1.invb(); strt1=true; ptf(dst,aip1,fpo,2);

cout<<"\nainv="; ptf(dst,aip1,fpo,1);

cout<<'\n'; //system("pause");

}

else{

aip.getbarrd(ri,resr);

ri.clear(); li.clear();

}

cout<<'\n';

}

cout<<'\n';

}

fpo.close(); cout<<'\n'; //system("pause");

}

void desdec(string& srs, string& dst, dtb key,vector<dtb>& ksh){

dtb fdt; bool strt=true,strt1=true; ifstream fpi; ofstream fpo; ui8 gffres;

fpo.open(dst.c\_str(),ios::binary);

if(!fpo.is\_open()) {cout<<"\nerr:ptff no output file"; system("pause"); return;}

cout<<'\n';

cout<<"\nkey="; ptf(dst,key,fpo,1);

string dt; gff(srs,dt,fpi);

while(dt.length()!=0){

for(ui8 i=0;i<8;i++){

if(dt.length()>0){

cout<<"\nsizeof(dt)="<<dt.length()<<' '<<dt<<'\n'; //system("pause");

fdt.ucarr[i]=(unsigned char)((dt.c\_str())[0]);

cout<<"\ndt="<<hex<<(usi)fdt.ucarr[i]<<dec;

dt.erase(dt.begin());

}

else fdt.ucarr[i]=0;

}

dtb apc1,apc2,aip;

cout<<"\ndata="; ptf(dst,fdt,fpo,1);

for(ui8 i=0;i<sizeof(pc1);i++){

ui8 nbyte=i/8, nbit=i%8, nbyte1=(pc1[i]-1)/8,nbit1=(pc1[i]-1)%8;

apc1.barr[nbyte].putb(nbit,key.barr[nbyte1].getb(7-nbit1));

}

for(ui8 i=0;i<sizeof(ip);i++){

ui8 nbyte=i/8, nbit=i%8, nbyte1=(ip[i]-1)/8,nbit1=(ip[i]-1)%8;

aip.barr[nbyte].putb(nbit,fdt.barr[nbyte1].getb(7-nbit1));

}

for(ui8 i=0;i<sizeof(itrol);i++){//round;

for(ui8 j=0;j<itrol[i];j++){//num of shifts;

vector<bool> bvecl,bvecr; apc1.getarr(bvecl,bvecr);

/\*cout<<'\n'; for(ui8 l=0;l<bvecl.size();l++) cout<<bvecl.at(l); cout<<'\t';

for(ui8 l=0;l<bvecr.size();l++) cout<<bvecr.at(l); cout<<'\n'; system("pause");\*/

ui8 u8tmpl=bvecl.at(bvecl.size()-1), u8tmpr=bvecr.at(bvecr.size()-1);

for(ui8 k=bvecl.size()-2;k!=0;k--){

bvecl.at(k+1)=bvecl.at(k);

bvecr.at(k+1)=bvecr.at(k);

}

bvecl.at(0)=u8tmpl; bvecr.at(0)=u8tmpr;

apc1.getbarr(bvecl,bvecr);

}

for(ui8 k=0;k<sizeof(pc2);k++){

ui8 nbyte=k/8, nbit=k%8, nbyte1=(pc2[k]-1)/8,nbit1=(pc2[k]-1)%8;

apc2.barr[nbyte].putb(nbit,apc1.barr[nbyte1].getb(nbit1));

}//apc2 is round key; apc1 is Ci, Di;

dtb rndkey=ksh.at(ksh.size()-1); ksh.pop\_back();

cout<<"\nkeysh="; ptf(dst,rndkey,fpo,0);

vector<bool> li,ri; aip.getarrd(li,ri); dtb aep,asb,app;

//cout<<'\n'; for(ui8 l=0;l<li.size();l++) cout<<li.at(l); cout<<'\t';

//for(ui8 l=0;l<ri.size();l++) cout<<ri.at(l); cout<<'\n'; system("pause");

for(ui8 j=0;j<sizeof(ep);j++){

ui8 nbyte=j/8, nbit=j%8, nbyte1=(ep[j]-1)/8,nbit1=(ep[j]-1)%8;

aep.barr[nbyte].putb(nbit,ri.at(ep[j]-1));

}

cout<<"\naep="; ptf(dst,aep,fpo,0); cout<<'\n';

for(ui8 j=0;j<sizeof(ep);j++){//xor ki;

ui8 nbyte=j/8, nbit=j%8;

aep.barr[nbyte].putb(nbit,(rndkey.barr[nbyte].getb(nbit)^aep.barr[nbyte].getb(nbit)));

}

cout<<"\naep xor ki ="; ptf(dst,aep,fpo,0);

//vector<bool> sbvl,sbvr; aep.getarrd(sbvl,sbvr);

//for(usi j=0;j<sbvr.size();j++) sbvl.push\_back(sbvr.at(j));

//cout<<"\n\t"; for(usi j=0;j<sbvl.size();j++) cout<<sbvl.at(j); cout<<' ';

for(ui8 j=0,scnt=0;j<sizeof(ep);j+=6,scnt++){

ui8 nbyte=j/8, nbit=j%8;

ui8 nbyte1=(j+1)/8, nbit1=(j+1)%8;

ui8 nbyte2=(j+2)/8, nbit2=(j+2)%8;

ui8 nbyte3=(j+3)/8, nbit3=(j+3)%8;

ui8 nbyte4=(j+4)/8, nbit4=(j+4)%8;

ui8 nbyte5=(j+5)/8, nbit5=(j+5)%8;

unsigned char uctmp;

/\*cout<<"x,y="<<aep.barr[nbyte].getb(nbit)\*2+aep.barr[nbyte5].getb(nbit5)<<' '

<<aep.barr[nbyte1].getb(nbit1)\*2\*2\*2+aep.barr[nbyte2].getb(nbit2)\*2\*2+

aep.barr[nbyte3].getb(nbit3)\*2+aep.barr[nbyte4].getb(nbit4);

cout<<'\n'<<(usi)aep.barr[nbyte].getb(nbit)<<(usi)aep.barr[nbyte1].getb(nbit1)<<(usi)aep.barr[nbyte2].getb(nbit2)

<<(usi)aep.barr[nbyte3].getb(nbit3)<<(usi)aep.barr[nbyte4].getb(nbit4)<<(usi)aep.barr[nbyte5].getb(nbit5)<<'\n';\*/

switch(scnt){

case 0:

uctmp=sb1[aep.barr[nbyte].getb(nbit)\*2+aep.barr[nbyte5].getb(nbit5)]

[aep.barr[nbyte1].getb(nbit1)\*2\*2\*2+aep.barr[nbyte2].getb(nbit2)\*2\*2+

aep.barr[nbyte3].getb(nbit3)\*2+aep.barr[nbyte4].getb(nbit4)];

break;

case 1:

uctmp=sb2[aep.barr[nbyte].getb(nbit)\*2+aep.barr[nbyte5].getb(nbit5)]

[aep.barr[nbyte1].getb(nbit1)\*2\*2\*2+aep.barr[nbyte2].getb(nbit2)\*2\*2+

aep.barr[nbyte3].getb(nbit3)\*2+aep.barr[nbyte4].getb(nbit4)];

break;

case 2:

uctmp=sb3[aep.barr[nbyte].getb(nbit)\*2+aep.barr[nbyte5].getb(nbit5)]

[aep.barr[nbyte1].getb(nbit1)\*2\*2\*2+aep.barr[nbyte2].getb(nbit2)\*2\*2+

aep.barr[nbyte3].getb(nbit3)\*2+aep.barr[nbyte4].getb(nbit4)];

break;

case 3:

uctmp=sb4[aep.barr[nbyte].getb(nbit)\*2+aep.barr[nbyte5].getb(nbit5)]

[aep.barr[nbyte1].getb(nbit1)\*2\*2\*2+aep.barr[nbyte2].getb(nbit2)\*2\*2+

aep.barr[nbyte3].getb(nbit3)\*2+aep.barr[nbyte4].getb(nbit4)];

break;

case 4:

uctmp=sb5[aep.barr[nbyte].getb(nbit)\*2+aep.barr[nbyte5].getb(nbit5)]

[aep.barr[nbyte1].getb(nbit1)\*2\*2\*2+aep.barr[nbyte2].getb(nbit2)\*2\*2+

aep.barr[nbyte3].getb(nbit3)\*2+aep.barr[nbyte4].getb(nbit4)];

break;

case 5:

uctmp=sb6[aep.barr[nbyte].getb(nbit)\*2+aep.barr[nbyte5].getb(nbit5)]

[aep.barr[nbyte1].getb(nbit1)\*2\*2\*2+aep.barr[nbyte2].getb(nbit2)\*2\*2+

aep.barr[nbyte3].getb(nbit3)\*2+aep.barr[nbyte4].getb(nbit4)];

break;

case 6:

uctmp=sb7[aep.barr[nbyte].getb(nbit)\*2+aep.barr[nbyte5].getb(nbit5)]

[aep.barr[nbyte1].getb(nbit1)\*2\*2\*2+aep.barr[nbyte2].getb(nbit2)\*2\*2+

aep.barr[nbyte3].getb(nbit3)\*2+aep.barr[nbyte4].getb(nbit4)];

break;

case 7:

uctmp=sb8[aep.barr[nbyte].getb(nbit)\*2+aep.barr[nbyte5].getb(nbit5)]

[aep.barr[nbyte1].getb(nbit1)\*2\*2\*2+aep.barr[nbyte2].getb(nbit2)\*2\*2+

aep.barr[nbyte3].getb(nbit3)\*2+aep.barr[nbyte4].getb(nbit4)];

break;

}

if(scnt%2==0){

asb.ucarr[scnt/2]=uctmp<<4;

}

else{

asb.ucarr[scnt/2]+=uctmp;

}

cout<<"\nuctmp="<<(usi)uctmp<<' '<<hex<<(usi)uctmp<<dec;;

}

cout<<"\nasb="; ptf(dst,asb,fpo,1);

for(ui8 j=0;j<sizeof(pp);j++){

ui8 nbyte=j/8, nbit=j%8, nbyte1=(pp[j]-1)/8,nbit1=(pp[j]-1)%8;

app.barr[nbyte].putb(nbit,asb.barr[nbyte1].getb(7-nbit1));

}

cout<<"\napp="; ptf(dst,app,fpo,0);

//cout<<'\n'; for(ui8 j=0;j<li.size();j++) cout<<li.at(j); cout<<' '; //system("pause");

vector<bool> resr;

for(ui8 j=0;j<li.size();j++) resr.push\_back(app.barr[j/8].getb(j%8)^li.at(j));

cout<<"\nnli="; for(ui8 j=0;j<li.size();j++) cout<<ri.at(j);

cout<<"\nnri="; for(ui8 j=0;j<resr.size();j++) cout<<resr.at(j);

//cout<<"\nsizrof(ri)="<<ri.size()<<" sizeof(resr)="<<resr.size(); cout<<' '; system("pause");

if(i==15){

aip.getbarrd(resr,ri); dtb aip1;

cout<<"\nbinv="; ptf(dst,aip,fpo,0);

for(ui8 j=0;j<sizeof(ip1);j++){

ui8 nbyte=j/8, nbit=j%8, nbyte1=(ip1[j]-1)/8,nbit1=(ip1[j]-1)%8;

aip1.barr[nbyte].putb(nbit,aip.barr[nbyte1].getb(nbit1));

}

aip1.invb(); strt1=true; ptf(dst,aip1,fpo,2);

cout<<"\nainv="; ptf(dst,aip1,fpo,1);

cout<<'\n'; //system("pause");

}

else{

aip.getbarrd(ri,resr);

ri.clear(); li.clear();

}

cout<<'\n';

}

cout<<'\n';

}

fpo.close();

}

void tdesenc(string tsrs, string tdst,dtb tkey,vector<dtb>& tksh,dtb tkey1,vector<dtb>& tksh1,dtb tkey2,vector<dtb>& tksh2){

desenc(tsrs, tdst, tkey, tksh);

desenc(tdst, tsrs, tkey1, tksh1);

desenc(tsrs, tdst, tkey2, tksh2);

}

void tdesdec(string tdst,string tddst,dtb tkey,vector<dtb>& tksh,dtb tkey1,vector<dtb>& tksh1,dtb tkey2,vector<dtb>& tksh2){

desdec(tdst, tddst, tkey2,tksh2);

desdec(tddst, tdst, tkey1,tksh1);

desdec(tdst, tddst, tkey,tksh);

}

int main(){

string srs="q.txt",dst="q1.txt",ddst="q2.txt"; dtb key={59,56,152,55,21,32,247,94}; vector<dtb> ksh;

desenc(srs, dst, key, ksh);

desdec(dst, ddst, key,ksh);

string tsrs="tq.txt",tdst="tq1.txt",tddst="tq2.txt";

dtb tkey={59,56,152,55,21,32,247,94}, tkey1={59,56,152,55,21,32,247,94}, tkey2={59,56,152,55,21,32,247,94};

vector<dtb> tksh,tksh1,tksh2;

tdesenc(tsrs,tdst,tkey,tksh,tkey1,tksh1,tkey2,tksh2);

tdesdec(tdst,tddst,tkey,tksh,tkey1,tksh1,tkey2,tksh2);

return 0;

}

Додаток 2. Лістинг des.h

#ifndef DES\_H

#define DES\_H

#include<cstdint>

typedef uint8\_t ui8;

ui8 pc1[]{57, 49, 41 ,33 ,25 ,17 ,9,

1 ,58 ,50 ,42 ,34 ,26 ,18,

10 ,2 ,59 ,51 ,43 ,35 ,27,

19 ,11 ,3 ,60 ,52 ,44 ,36,

63, 55 ,47 ,39 ,31 ,23 ,15,

7 ,62 ,54 ,46 ,38 ,30 ,22,

14 ,6 ,61 ,53 ,45 ,37 ,29,

21 ,13 ,5 ,28 ,20 ,12 ,4

};

ui8 itrol[]{1,1,2,2,2,2,2,2,1,2,2,2,2,2,2,1};

ui8 pc2[]{14 ,17 ,11 ,24 ,1 ,5,

3 ,28 ,15 ,6 ,21 ,10,

23 ,19 ,12 ,4 ,26 ,8,

16 ,7 ,27 ,20 ,13 ,2,

41 ,52 ,31 ,37 ,47 ,55,

30 ,40 ,51 ,45 ,33 ,48,

44 ,49 ,39 ,56 ,34 ,53,

46 ,42 ,50 ,36 ,29 ,32

};

ui8 ip[]{58 ,50 ,42 ,34 ,26 ,18 ,10 ,2,

60 ,52 ,44 ,36 ,28 ,20 ,12 ,4,

62 ,54 ,46 ,38 ,30 ,22 ,14 ,6,

64 ,56 ,48 ,40 ,32 ,24 ,16 ,8,

57 ,49 ,41 ,33 ,25 ,17 ,9 ,1,

59 ,51 ,43 ,35 ,27 ,19 ,11 ,3,

61 ,53 ,45 ,37 ,29 ,21 ,13 ,5,

63 ,55 ,47 ,39 ,31 ,23 ,15 ,7

};

ui8 ep[]{32, 1, 2, 3, 4, 5,

4 , 5 , 6, 7, 8, 9,

8 , 9 , 10, 11, 12, 13,

12, 13, 14, 15, 16, 17,

16, 17, 18, 19, 20, 21,

20, 21, 22, 23, 24, 25,

24, 25, 26, 27, 28, 29,

28, 29, 30, 31, 32, 1

};

ui8 sb1[][16]{{14 ,4 ,13 ,1 ,2 ,15 ,11 ,8 ,3 ,10 ,6 ,12 ,5 ,9 ,0 ,7},

{0 ,15 ,7 ,4 ,14 ,2 ,13 ,1 ,10 ,6 ,12 ,11 ,9 ,5 ,3 ,8},

{4 ,1 ,14 ,8 ,13 ,6 ,2 ,11 ,15 ,12 ,9 ,7 ,3 ,10 ,5 ,0},

{15 ,12 ,8 ,2 ,4 ,9 ,1 ,7 ,5 ,11 ,3 ,14 ,10 ,0 ,6 ,13}};

ui8 sb2[][16]{{15 ,1 ,8 ,14 ,6 ,11 ,3 ,4 ,9 ,7 ,2 ,13 ,12 ,0 ,5 ,10},

{3 ,13 ,4 ,7 ,15 ,2 ,8 ,14 ,12 ,0 ,1 ,10 ,6 ,9 ,11 ,5},

{0 ,14 ,7 ,11 ,10 ,4 ,13 ,1 ,5 ,8 ,12 ,6 ,9 ,3 ,2 ,15},

{13 ,8 ,10 ,1, 3 ,15 ,4 ,2 ,11 ,6 ,7 ,12 ,0 ,5 ,14 ,9}};

ui8 sb3[][16]{{10 ,0 ,9 ,14 ,6 ,3 ,15 ,5 ,1 ,13 ,12 ,7 ,11 ,4 ,2 ,8},

{13 ,7 ,0 ,9 ,3 ,4 ,6 ,10 ,2 ,8 ,5 ,14 ,12 ,11 ,15 ,1},

{13 ,6 ,4 ,9 ,8 ,15, 3 ,0 ,11 ,1 ,2 ,12 ,5 ,10 ,14 ,7},

{1 ,10 ,13 ,0 ,6 ,9 ,8, 7 ,4 ,15 ,14 ,3 ,11 ,5, 2, 12}};

ui8 sb4[][16]{{7 ,13 ,14 ,3 ,0 ,6 ,9 ,10 ,1 ,2 ,8 ,5 ,11 ,12 ,4 ,15},

{13 ,8 ,11 ,5 ,6 ,15 ,0 ,3 ,4 ,7 ,2 ,12 ,1 ,10 ,14 ,9},

{10 ,6 ,9 ,0 ,12 ,11 ,7 ,13 ,15, 1 ,3 ,14 ,5 ,2 ,8 ,4},

{3 ,15 ,0 ,6 ,10 ,1 ,13 ,8 ,9 ,4 ,5 ,11 ,12 ,7 ,2 ,14}};

ui8 sb5[][16]{{2 ,12 ,4 ,1 ,7 ,10 ,11 ,6 ,8 ,5 ,3 ,15 ,13 ,0 ,14 ,9},

{14 ,11 ,2 ,12 ,4 ,7 ,13 ,1 ,5 ,0 ,15 ,10 ,3 ,9 ,8 ,6},

{4 ,2 ,1 ,11 ,10 ,13 ,7 ,8 ,15 ,9 ,12 ,5 ,6 ,3 ,0 ,14},

{11 ,8 ,12 ,7 ,1 ,14 ,2 ,13 ,6 ,15 ,0 ,9 ,10 ,4 ,5 ,3}};

ui8 sb6[][16]{{12 ,1 ,10 ,15 ,9 ,2 ,6 ,8 ,0 ,13 ,3 ,4 ,14 ,7 ,5 ,11},

{10 ,15 ,4 ,2 ,7 ,12 ,9 ,5 ,6 ,1 ,13 ,14 ,0 ,11 ,3 ,8},

{9 ,14 ,15 ,5 ,2 ,8 ,12 ,3 ,7 ,0 ,4 ,10 ,1 ,13 ,11 ,6},

{4 ,3 ,2 ,12 ,9 ,5 ,15 ,10 ,11 ,14 ,1 ,7 ,6 ,0 ,8 ,13}};

ui8 sb7[][16]{{4 ,11 ,2 ,14 ,15 ,0 ,8 ,13 ,3 ,12 ,9 ,7 ,5 ,10 ,6 ,1},

{13 ,0 ,11 ,7 ,4 ,9 ,1 ,10 ,14 ,3 ,5 ,12 ,2 ,15 ,8 ,6},

{1 ,4 ,11 ,13 ,12 ,3 ,7 ,14 ,10 ,15 ,6 ,8 ,0 ,5 ,9, 2},

{6 ,11 ,13 ,8 ,1 ,4 ,10 ,7 ,9 ,5 ,0 ,15 ,14 ,2 ,3 ,12}};

ui8 sb8[][16]{{13, 2 ,8 ,4 ,6 ,15 ,11 ,1 ,10 ,9 ,3 ,14 ,5 ,0 ,12 ,7},

{1, 15 ,13 ,8 ,10 ,3 ,7 ,4, 12 ,5 ,6 ,11 ,0 ,14 ,9 ,2},

{7, 11 ,4 ,1 ,9 ,12 ,14 ,2 ,0 ,6 ,10 ,13 ,15 ,3 ,5 ,8},

{2, 1 ,14 ,7 ,4 ,10 ,8 ,13 ,15 ,12 ,9 ,0 ,3 ,5 ,6 ,11}};

ui8 pp[]{16 , 7 ,20 ,21,

29 ,12 ,28 ,17,

1 ,15 ,23 ,26,

5 ,18 ,31 ,10,

2 ,8 ,24 ,14,

32 ,27 ,3 ,9,

19 ,13 ,30 ,6,

22 ,11 ,4 ,25

};

ui8 ip1[]{40 ,8 ,48 ,16 ,56 ,24 ,64 ,32,

39 ,7 ,47 ,15 ,55 ,23 ,63 ,31,

38 ,6 ,46 ,14 ,54 ,22 ,62 ,30,

37 ,5 ,45 ,13 ,53 ,21 ,61 ,29,

36 ,4 ,44 ,12 ,52 ,20 ,60 ,28,

35 ,3 ,43 ,11 ,51 ,19 ,59 ,27,

34 ,2 ,42 ,10 ,50 ,18 ,58 ,26,

33 ,1 ,41 ,9 ,49 ,17 ,57 ,25

};

#endif