МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

НАВЧАЛЬНО-НАУКОВИЙ КОМПЛЕКС

«ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ»

Лабораторна робота № 5

з курсу «Технології захисту інформації»

Тема: «Дослідження стеганографічних методів приховування даних»

Виконав:

студент IV курсу

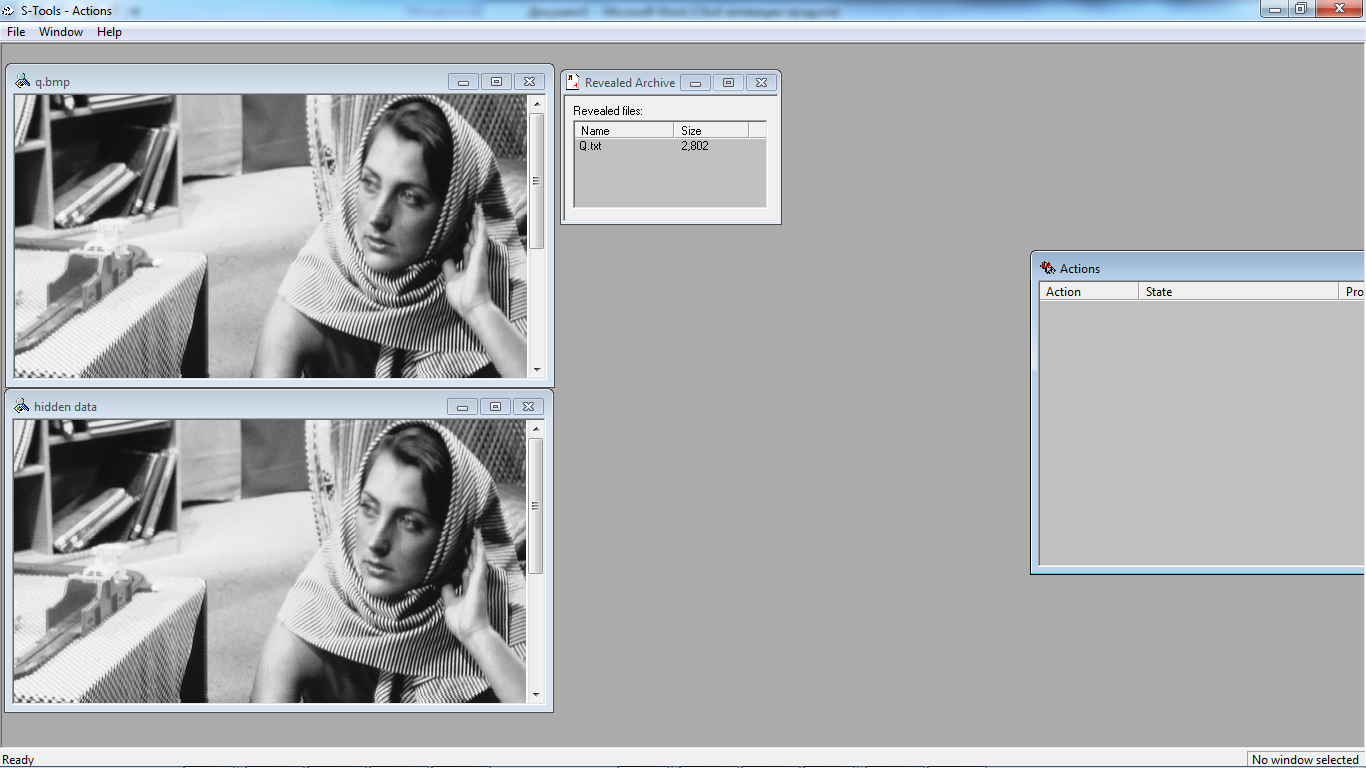
групи ДА-32

Колінько Анжела

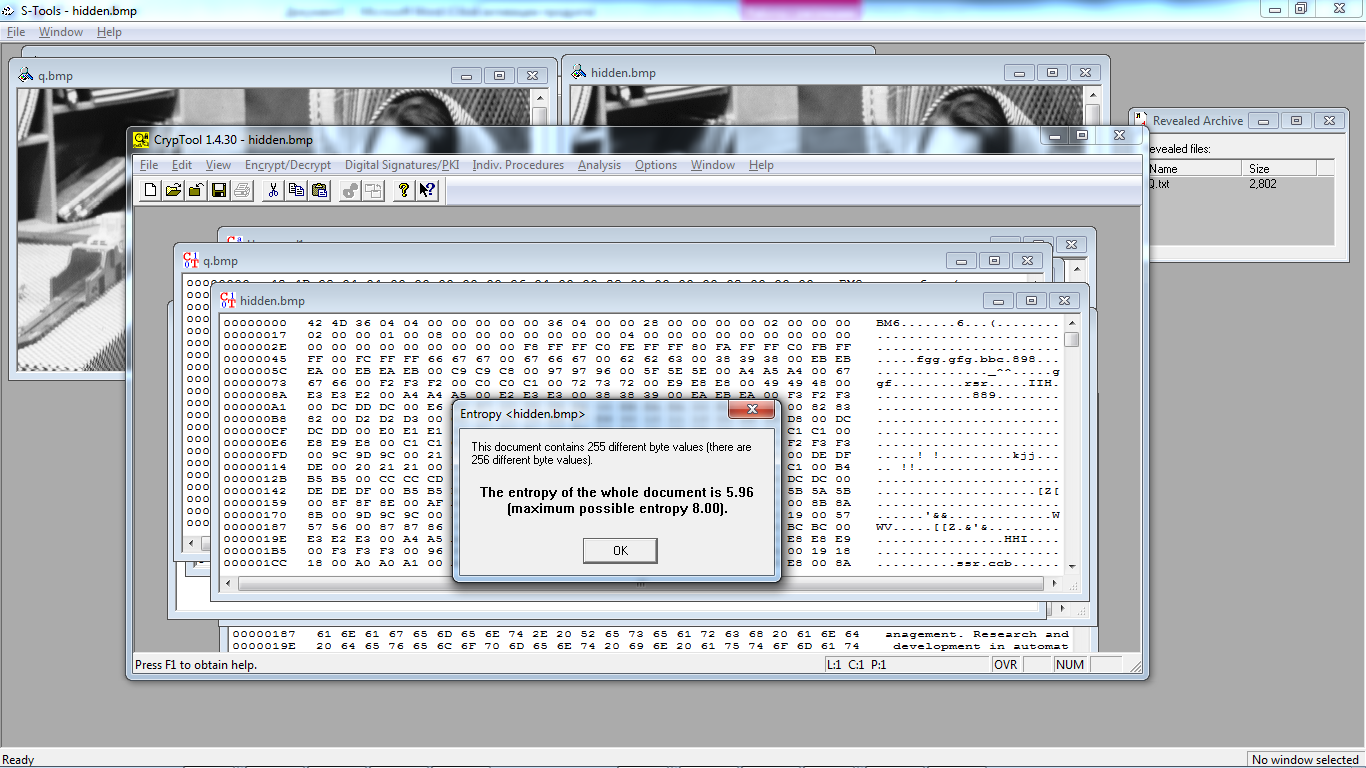
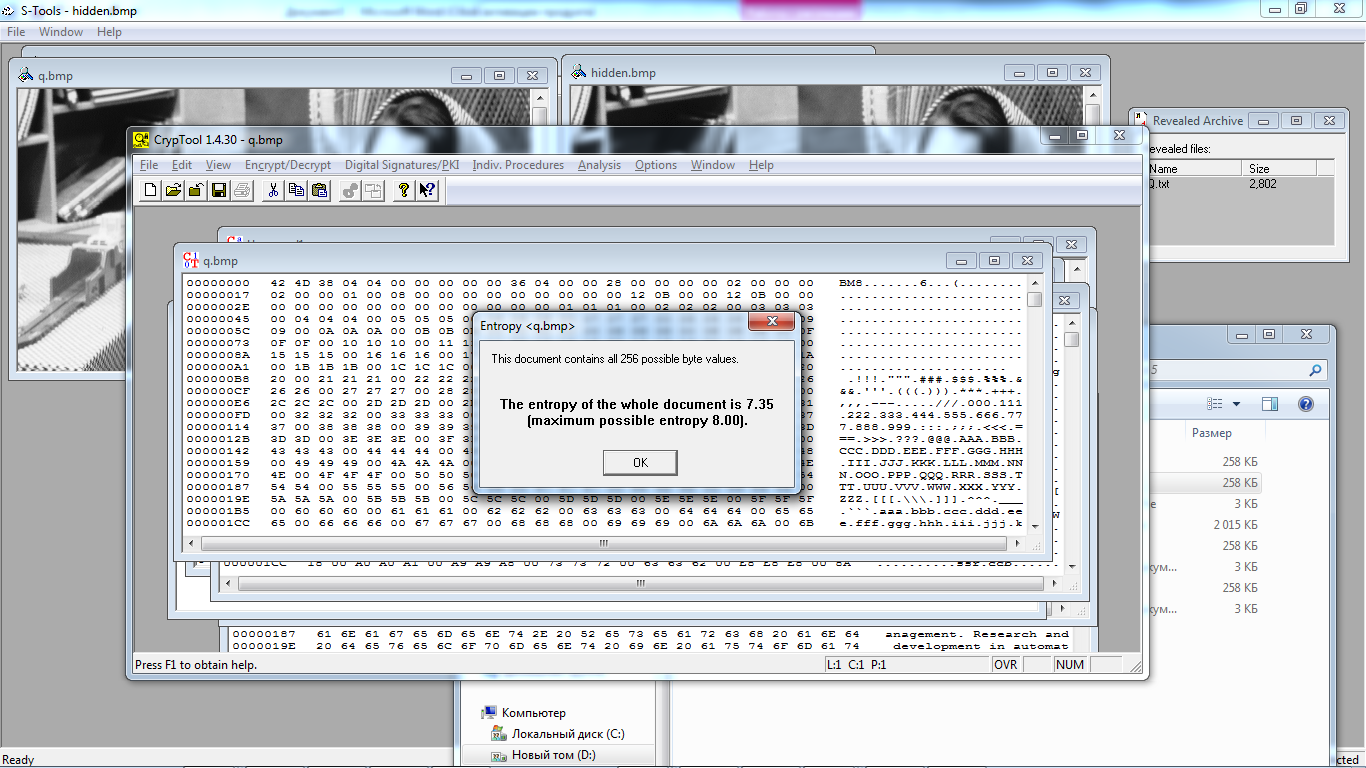
Київ – 2017

Варіант 11

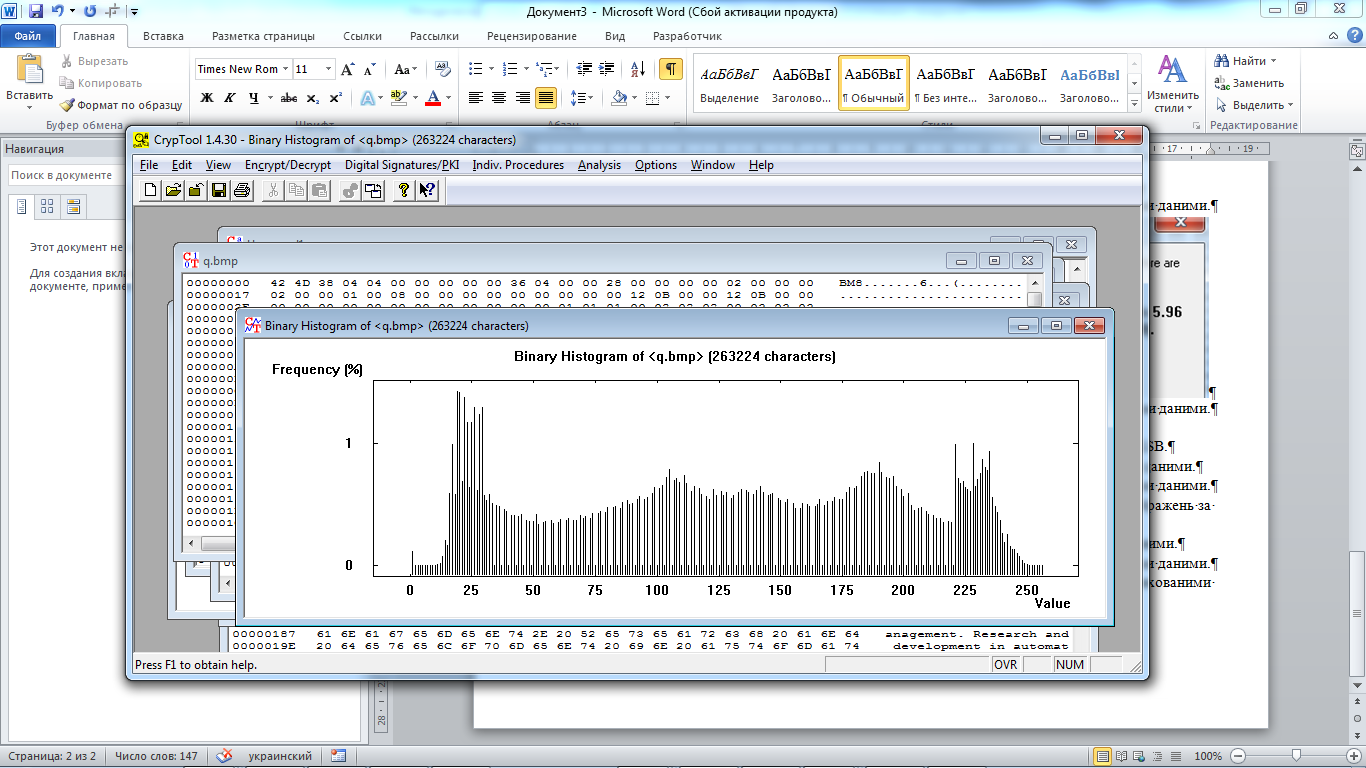
1. Обраний контейнер-зображення, контейнер-зображення з прихованими даними.

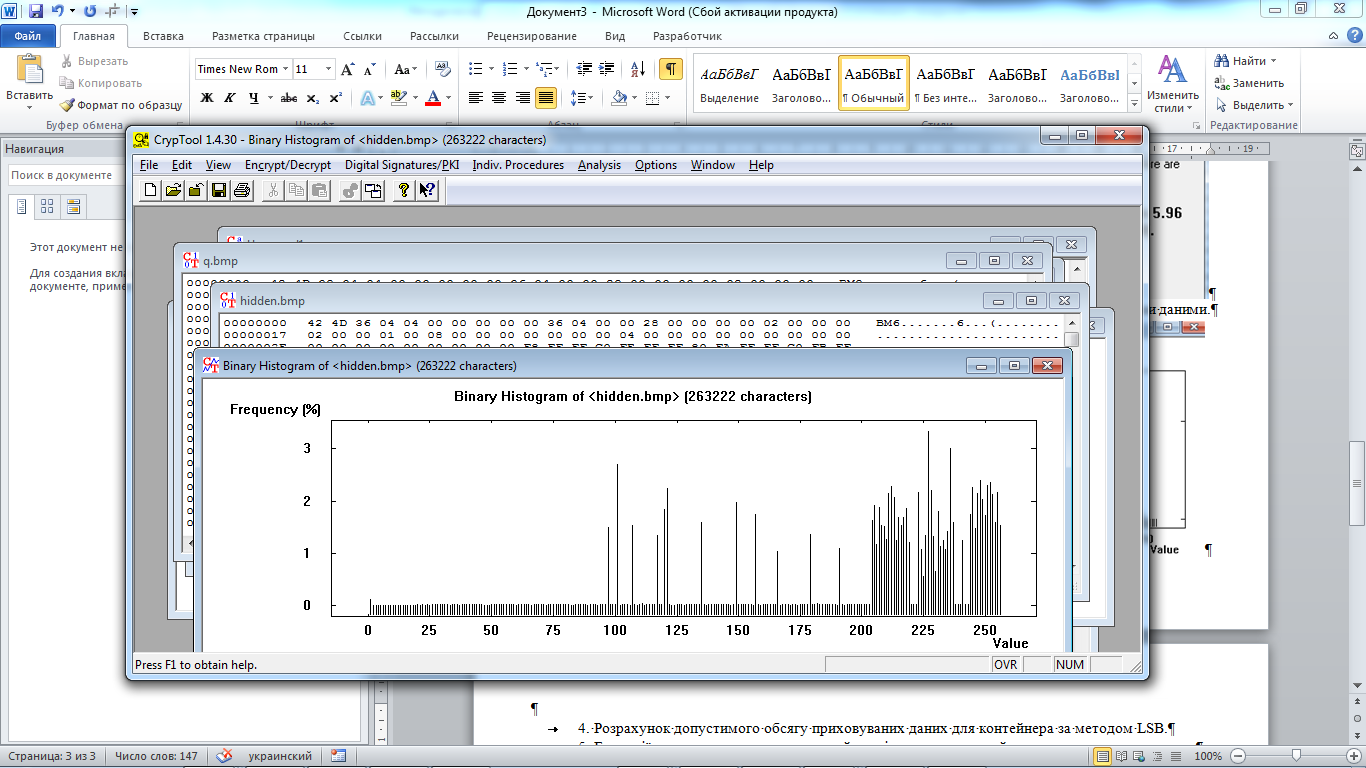


2. Ентропії порожнього контейнера-зображення і контейнера-зображення з прихованими даними.



3. Гістограми порожнього контейнера-зображення і контейнера-зображення з прихованими даними.

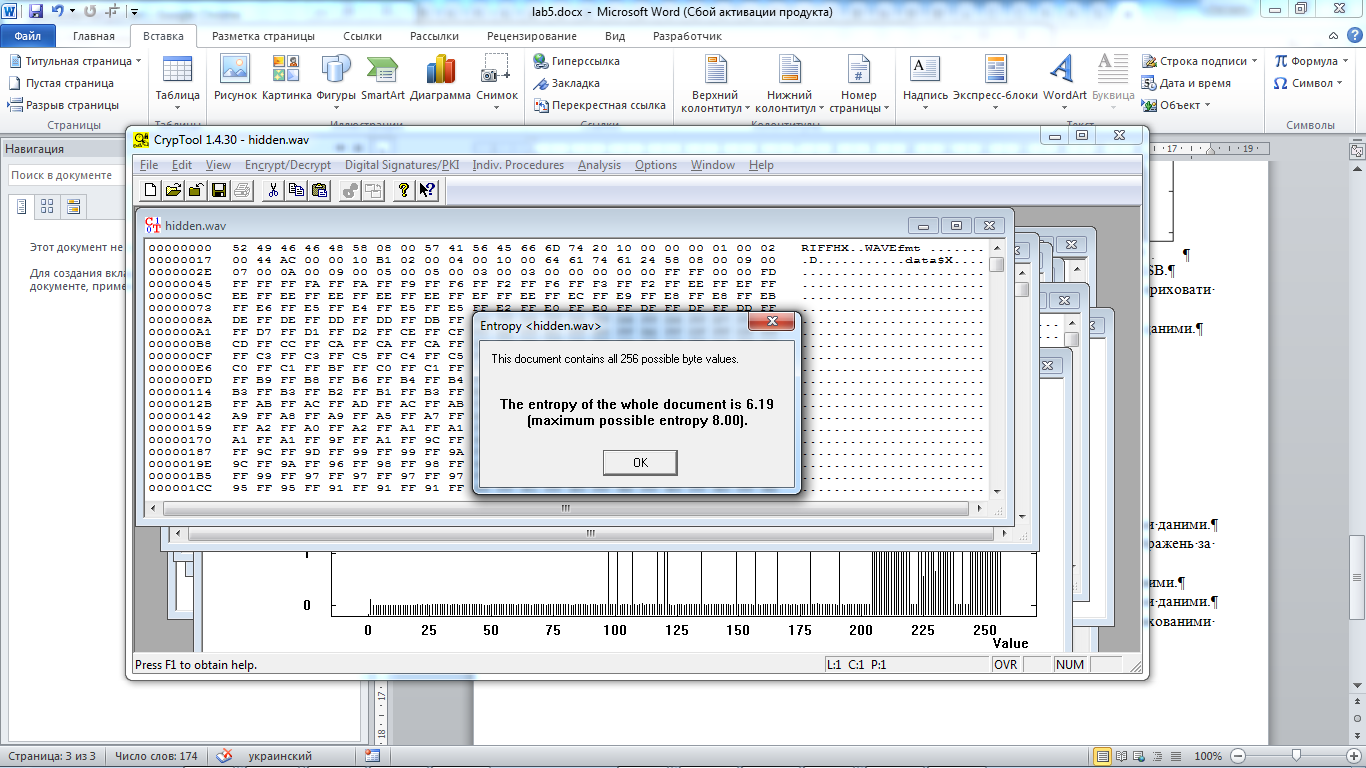
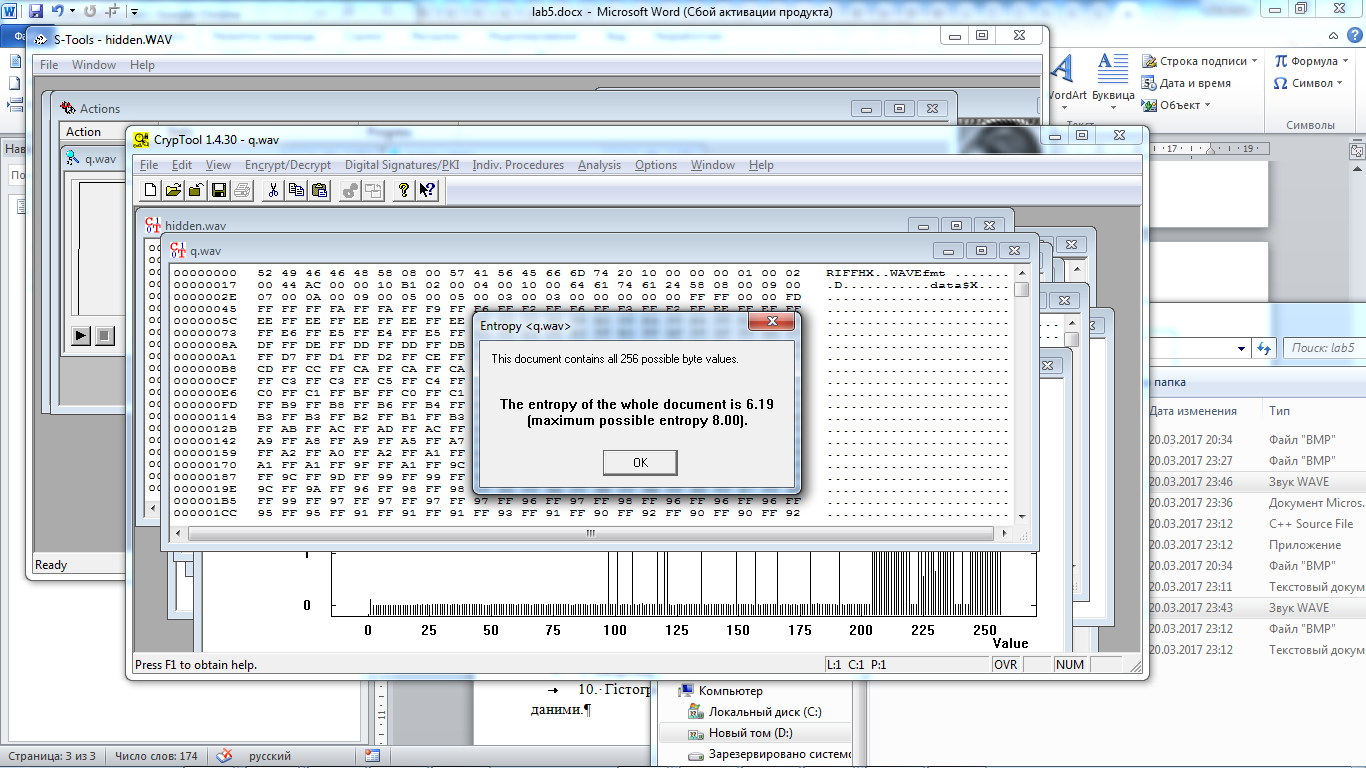




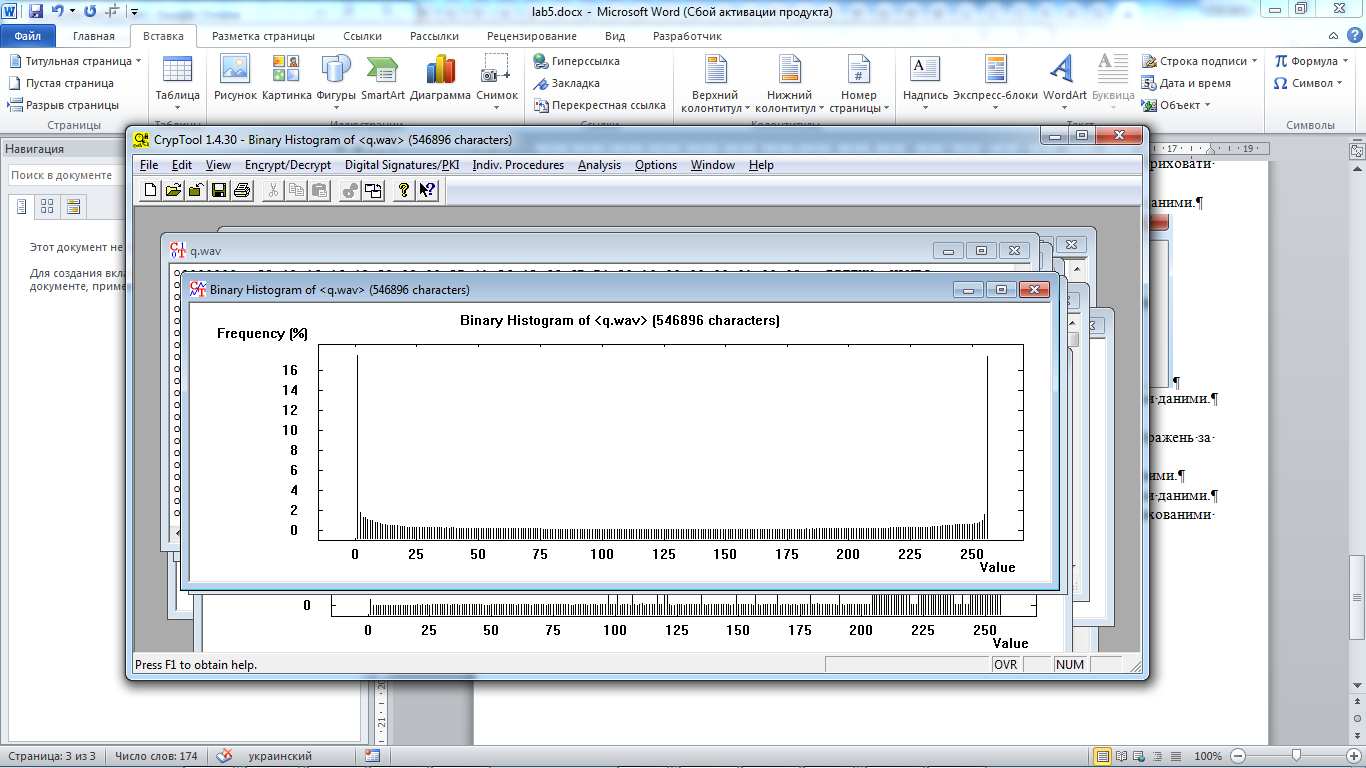
4. Розрахунок допустимого обсягу приховуваних даних для контейнера за методом LSB.

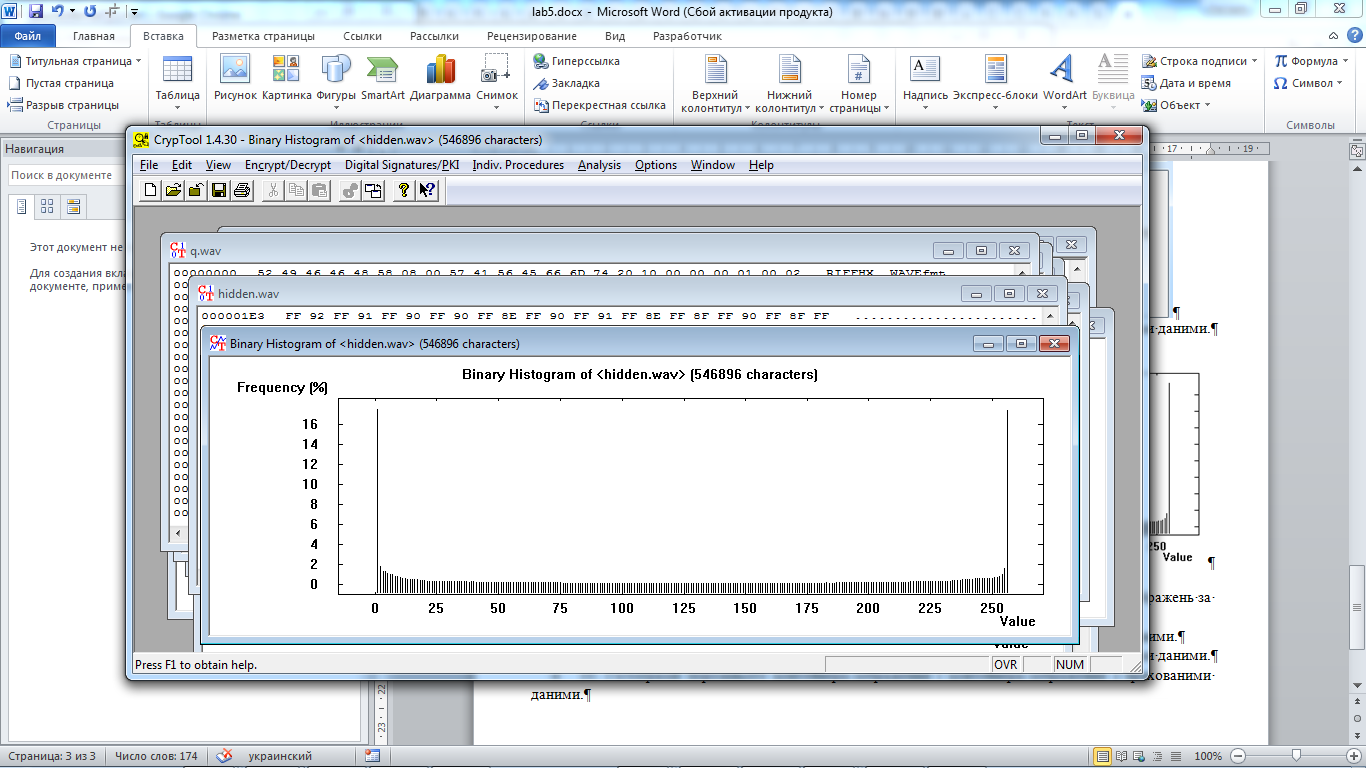
Розмір q.bmp – 263 224 байт. Якщо за методом LSB в кожному байті заміняти 1 біт, то можна приховати відповідно 263 224 біт = 32903 байт (≈32 KB).

5. Ентропії порожнього звукового контейнера і звукового контейнера з прихованими даними.



6. Гістограми порожнього звукового контейнера і звукового контейнера з прихованими даними.





7. Лістинг програми для приховування та вилучення даних в файлах напівтонових зображень за методом LSB.

#include<iostream>

#include<string>

#include<fstream>

#define otgo 11

using namespace std;

typedef unsigned short int usi;

unsigned int otd,ml;

union otgd {

unsigned char ucharr[sizeof(unsigned int)];

unsigned int uin;

void setuch(usi i,unsigned char uch){

ucharr[i]=uch;

}

};

int main(){

string srs="q.bmp",tsrs="q.txt",dst="q1.bmp",ddst="q1.txt";

ifstream fpi,tfpi; ofstream fpo;

fpi.open(srs.c\_str(),ios::binary); if(!fpi.is\_open()) {cout<<"\nerr: gff no input file"; return 0;}

tfpi.open(tsrs.c\_str(),ios::binary); if(!tfpi.is\_open()) {cout<<"\nerr: gff no input file"; return 0;}

fpo.open(dst.c\_str(),ios::binary); if(!fpo.is\_open()) {cout<<"\nerr: gff no input file"; return 0;}

unsigned char ch='\0',tch='\0'; unsigned int cnt=0; otgd ofst; bool isst=0; usi tcnt=0;

while(1){

if(cnt>=otgo&&cnt<(otgo+4)){

ofst.setuch((cnt-otgo),ch);

cout<<dec<<"\nofst.ucharr["<<(usi)((cnt-otgo))<<"]="<<hex<<(usi)ofst.ucharr[(cnt-otgo)];

otd=ofst.uin;

if((cnt-otgo)==sizeof(unsigned int)-1) isst=1;

}

if(fpi.eof()) break;

if(!isst||cnt<=otd){

fpi>>noskipws>>ch; cnt++;

fpo<<ch;

}

else{

fpi>>noskipws>>ch; cnt++;

if(tcnt==0) {

tfpi>>noskipws>>tch; tcnt=8;

}

if(fpi.eof()){

ml=cnt; break;

}

if(tfpi.eof()){

ml=cnt;

while(1){

fpi>>noskipws>>ch; cnt++;

fpo<<ch;

if(fpi.eof()) break;

}

break;

}

//cout<<'\n'<<dec<<cnt<<" chb="<<hex<<(usi)ch<<" tchb="<<hex<<(usi)tch<<'\n';

ch=(ch&0xfe)|(0x01&tch); tch>>=1; tcnt--;

fpo<<ch;

/\*cout<<dec<<cnt<<" cha="<<hex<<(usi)ch<<" tcha="<<hex<<(usi)tch;

cout<<"\ndec="<<dec<<otd<<'\n';\*/

//system("pause");

}

//cout<<"\ndec="<<dec<<otd;

//cout<<dec<<cnt<<' '<<hex<<(usi)ch<<'\n';

}

fpi.close(); tfpi.close(); fpo.close();

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

fpi.open(dst.c\_str(),ios::binary); if(!fpi.is\_open()) {cout<<"\nerr: gff no input file"; return 0;}

fpo.open(ddst.c\_str(),ios::binary); if(!fpo.is\_open()) {cout<<"\nerr: gff no input file"; return 0;}

ch='\0'; tch='\0'; cnt=0; isst=0; tcnt=0;

while(1){

if(cnt>=otgo&&cnt<(otgo+4)){

ofst.setuch((cnt-otgo),ch);

cout<<dec<<"\nofst.ucharr["<<(usi)((cnt-otgo))<<"]="<<hex<<(usi)ofst.ucharr[(cnt-otgo)];

otd=ofst.uin;

if((cnt-otgo)==sizeof(unsigned int)-1) isst=1;

}

if(fpi.eof()) break;

if(cnt==ml) break;

if(!isst||cnt<=otd){

fpi>>noskipws>>ch; cnt++;

}

else{

fpi>>noskipws>>ch; cnt++;

if(tcnt==8) {

fpo<<tch; tcnt=0;//TODO;

tch='\0';

}

if(fpi.eof()) break;

//cout<<'\n'<<dec<<cnt<<" chb="<<hex<<(usi)ch<<" tchb="<<hex<<(usi)tch<<'\n';

if(0x01&ch){

tch|=1<<(tcnt);

}

tcnt++;

/\*cout<<dec<<cnt<<" cha="<<hex<<(usi)ch<<" tcha="<<hex<<(usi)tch;

cout<<"\ndec="<<dec<<otd<<'\n';\*/

//system("pause");

}

//cout<<"\ndec="<<dec<<otd;

//cout<<dec<<cnt<<' '<<hex<<(usi)ch<<'\n';

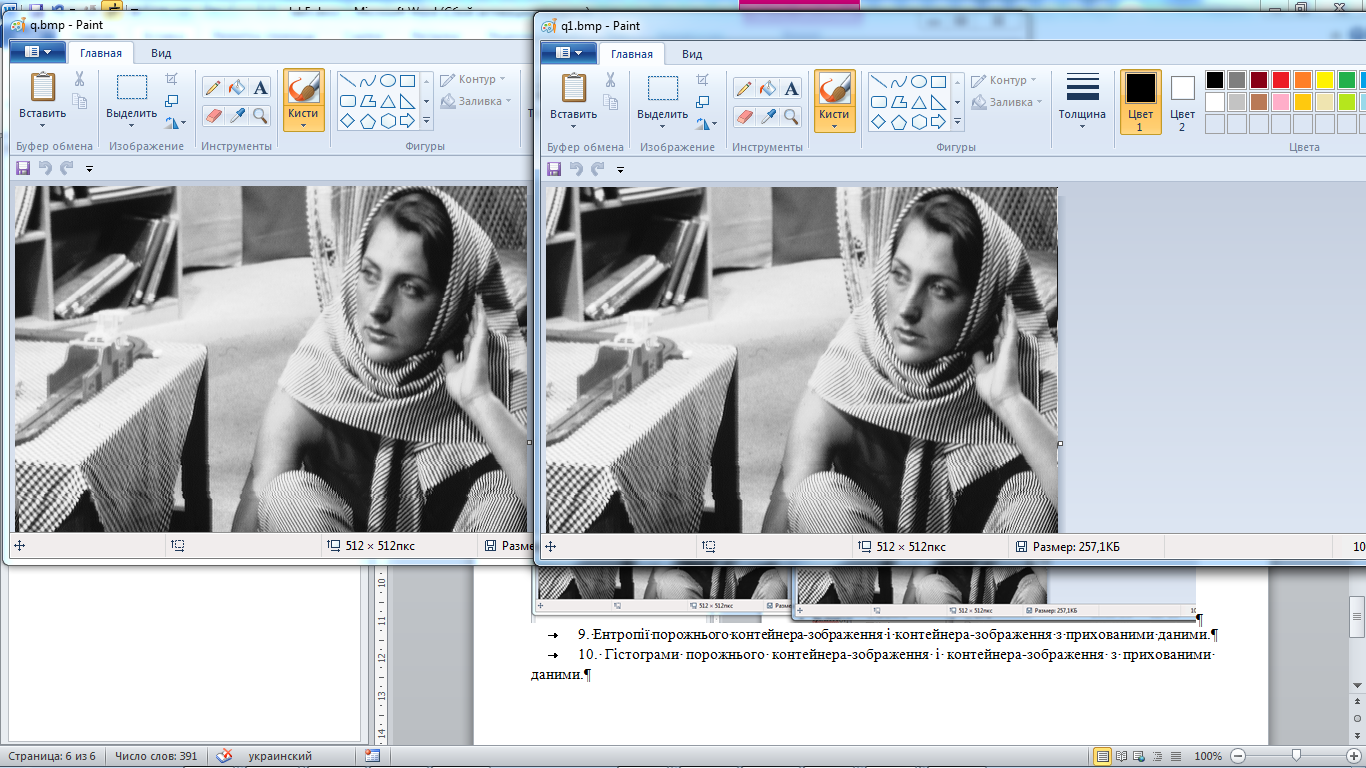
}

fpi.close(); fpo.close();

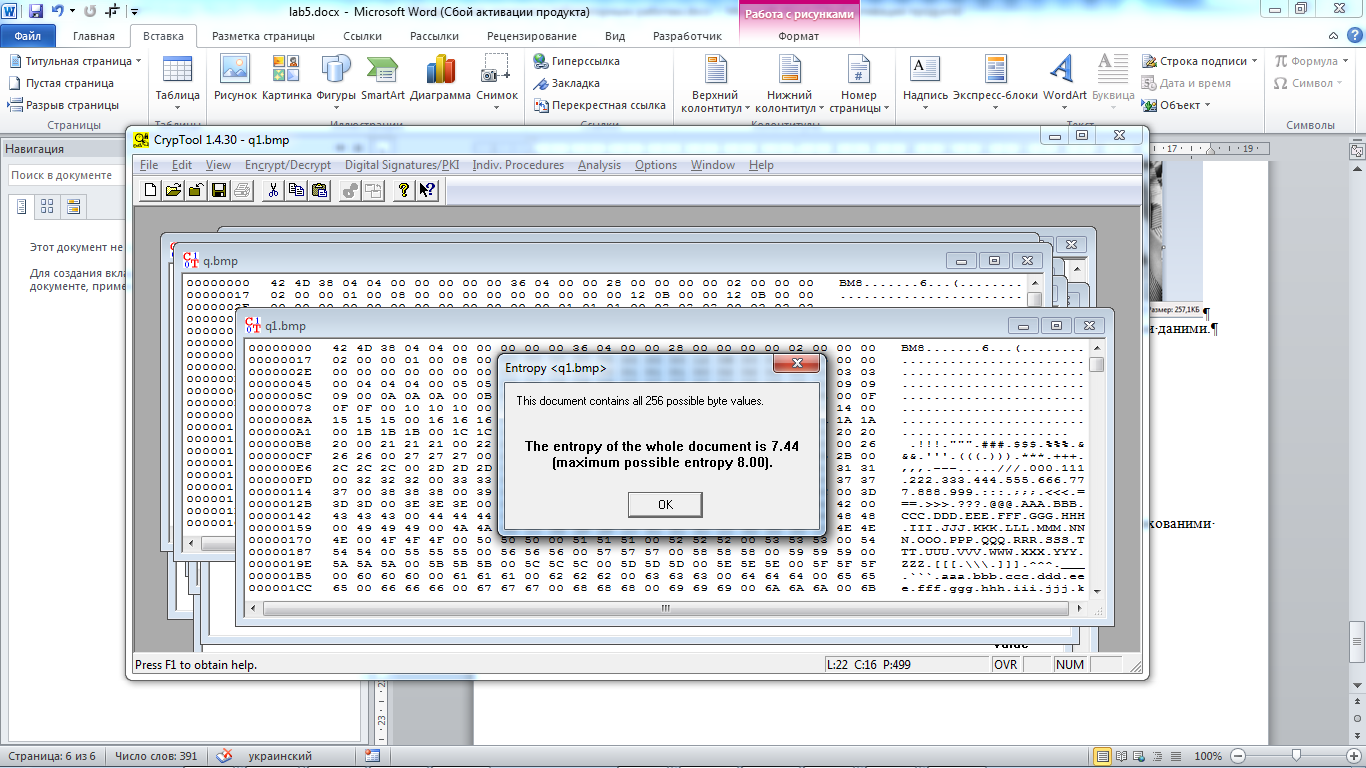
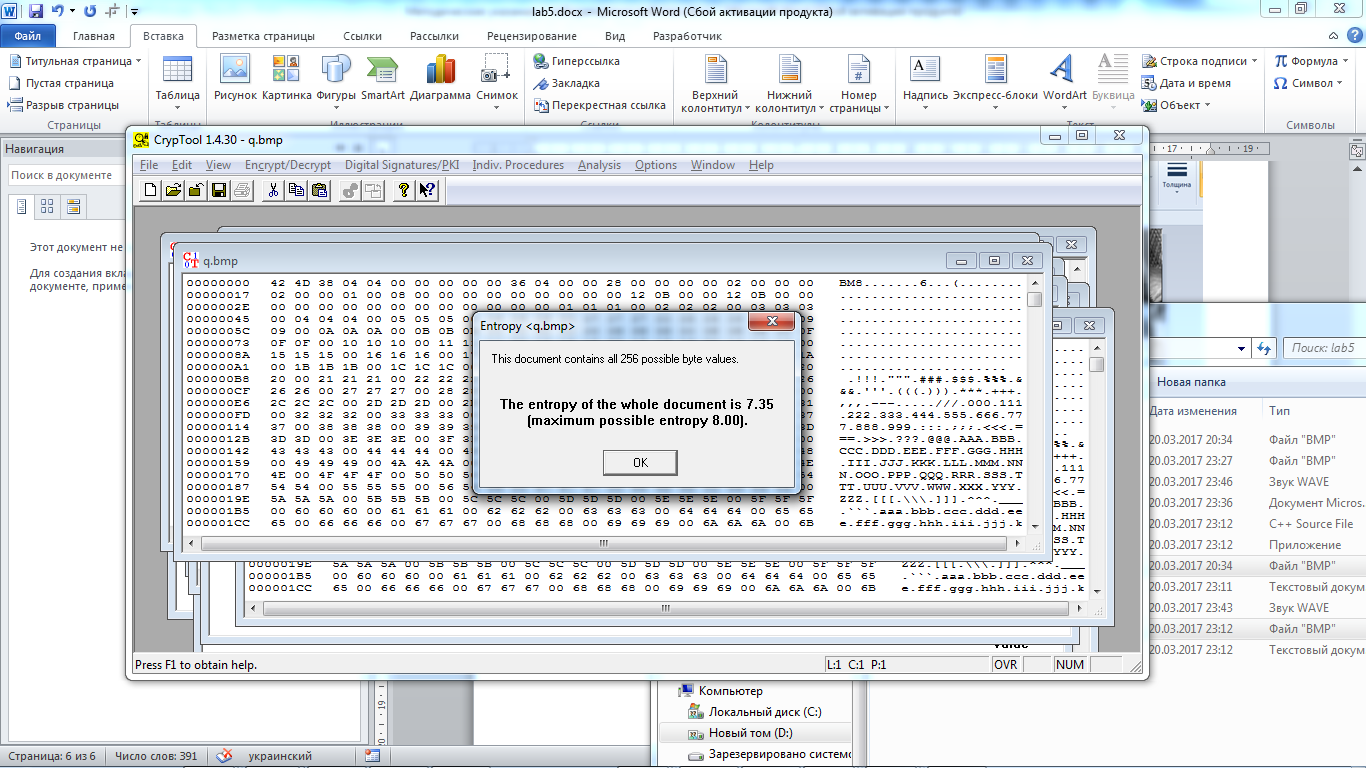
return 0;

}

8. Результат роботи розробленої програми (контейнер і контейнер з прихованими даними.



9. Ентропії порожнього контейнера-зображення і контейнера-зображення з прихованими даними.



10. Гістограми порожнього контейнера-зображення і контейнера-зображення з прихованими даними.

