Міністерство освіти і науки України

Сумський державний університет

Кафедра

Прикладної математики та моделювання складних систем

Звіт з лабораторної роботи №1

Дисципліна

Криптографія

Студентка: Пороскун Олена Олегівна

Викладач: Козлова Ірина Іванівна

Суми, Сумська область

2021

Лабораторна робота №1

**Тема**: Симетричні криптоперетворення

**Мета**: навчитися застосувати вивчені методи шифрування для зашифровування та розшифровування повідомлень, скласти алгоритм та програму розв’язування наступних задач.

***Зміст задачі:***

1. Зашифрувати повідомлення «(Своє повне ім’я і прізвище)» методом простої підстановки, алфавіт український або російський. Визначити розмірність простору ключів , якщо потужність крипто аналітичної системи у = 10 вар/с. Розшифруйте повідомлення і перевірте однозначність процедури за шифрування-розшифрування.

**Розв’язування задачі**

*Розмір простору ключів*

(де m - основа алфавіту).

*Ентропія джерела ключів*

*Безпечний час*

де ймовірність,з якою має бути здійснений криптоаналіз.

*Відстань єдності для шифру*

***Програмний код:***

#include <string.h>

#include <Windows.h>

#include <iostream>

using namespace std;

int main(){

setlocale (LC\_CTYPE, "ukr");

int size\_mes, size\_ABC;

char encrypted\_mes[20];

char decrypted\_mes[20];

char message[] = "олена\_пороскун";

cout << message << " - повiдомлення \n\n";

char ABC[] = "абвгдеєжзиiїйклмнопрстуфхцчшщьюя\_";

cout << ABC << " – вхiдний алфавiт\n";

char ABC1[] ="ьюя\_абвгдеєжзиiїйклмнопрстуфхцчшщ";

cout << ABC1 << " - вихiдний алфавiт\n\n";

size\_mes = sizeof(message);

size\_ABC = sizeof(ABC);

for (int i = 0; i < size\_mes; i++) {

for (int j = 0; j < size\_ABC; j++) {

if (ABC[j] == message[i]) {

encrypted\_mes[i] = ABC1[j];

}

}

}

cout << encrypted\_mes << " - зашифроване повiдомлення \n";

for (int i = 0; i < size\_mes; i++) {

for (int j = 0; j < size\_ABC; j++) {

if (ABC1[j] == encrypted\_mes[i]) {

decrypted\_mes[i] = ABC[j];

}

}

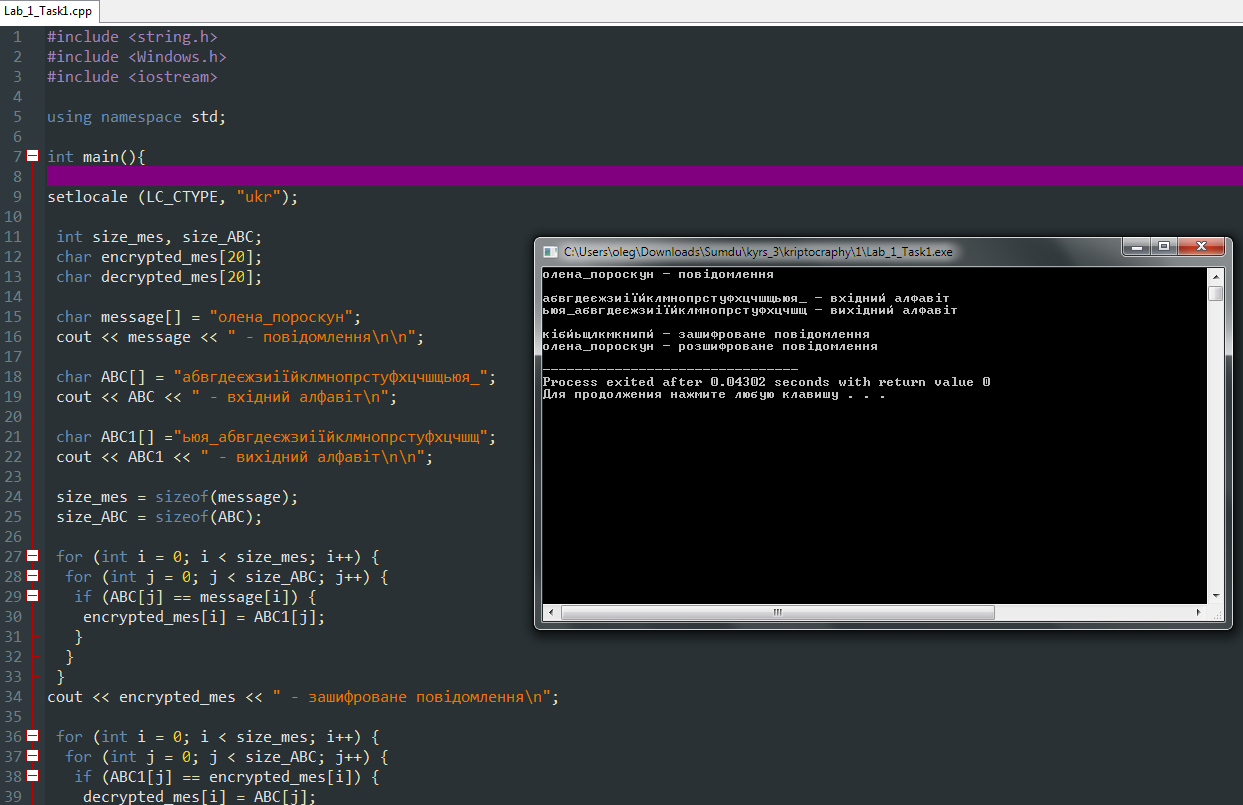
}

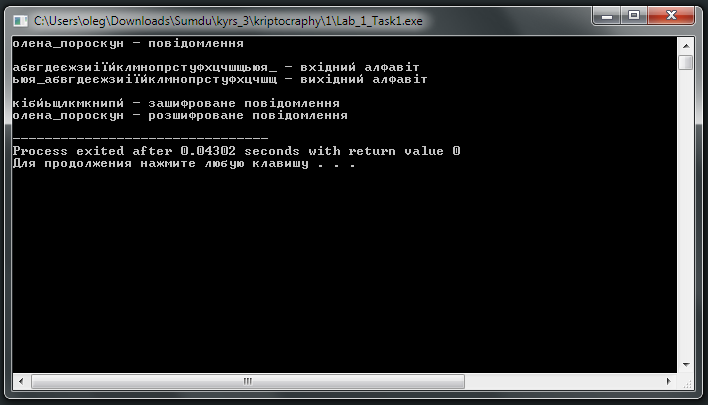
cout << decrypted\_mes << " - розшифроване повiдомлення\n";

return 0;

}

***Прінт-скрін виконання програми:***





***Зміст задачі:***

2. Зашифрувати повідомлення з попереднього завдання, використовуючи шифр Віженера. Визначити розмірність простору ключів , якщо потужність крипто аналітичної системи у = 10 вар/с. Розшифруйте повідомлення і перевірте однозначність процедури за шифрування-розшифрування.

Ключ «небосхил»,

**Розв’язування задачі**

*Розмір простору ключів*

(де m - основа алфавіту, - розмір ключа (довжина)).

*Ентропія джерела ключів*

*Безпечний час*

при = 1.

*Відстань єдності для шифру*

***Програмний код:***

#include <string.h>

#include <Windows.h>

#include <iostream>

using namespace std;

int main(){

setlocale (LC\_CTYPE, "ukr");

int size\_mes, size\_ABC, size\_key1;

char encrypted\_mes[20], decrypted\_mes[20];

char message[] = "олена\_пороскун";

cout << message << " - повiдомлення\n";

char key[] = "небосхил";

cout << "\n" << key << " - ключ \n";

char ABC[] = "абвгдеєжзиiїйклмнопрстуфхцчшщьюя\_";

cout << "\n" << ABC << " - вхiдний алфавiт\n\n";

size\_mes = sizeof(message)-1;

size\_ABC = sizeof(ABC)-1;

// ------------------------------------------------------------

// символи масиву М1

int M1[size\_mes];

for (int i = 0; i < size\_mes; i++) {

for (int j = 0; j < size\_ABC; j++) {

if (ABC[j] == message[i]) {

M1[i] = j;

}

}

}

for (int i = 0; i < size\_mes; i++) {

printf("%d ", M1[i]);

}

printf(" - повiдомлення(M1)\n");

// ------------------------------------------------------------

// символи масиву ключа Г1

char key1[] = "небосхилнебосх";

cout << "\n" << key1 << " - ключ з повторюваннями \n\n";

size\_key1 = sizeof(key1)-1;

int G1[size\_key1];

for (int i = 0; i < size\_key1; i++) {

for (int j = 0; j < size\_ABC; j++) {

if (ABC[j] == key1[i]) {

G1[i] = j;

}

}

}

for (int i = 0; i < size\_mes; i++) {

printf("%d ", G1[i]);

}

printf(" - ключ з повторюваннями(G1) \n\n");

// ------------------------------------------------------------

// символи криптограми С1

int C1[size\_key1];

for (int i = 0; i < size\_key1; i++) {

C1[i] = (M1[i] + G1[i]) % size\_ABC;

}

for (int i = 0; i < size\_mes; i++) {

printf("%d ", C1[i]);

}

printf(" - зашифроване повiдомлення (C1) \n\n");

// зашифроване повiдомлення

for (int i = 0; i < size\_mes; i++) {

for (int j = 0; j < size\_ABC; j++) {

if (C1[i] == j) {

encrypted\_mes[i] = ABC[j];

}

}

}

for (int i = 0; i < size\_mes; i++) {

printf("%c", encrypted\_mes[i]);

}

printf(" - зашифроване повiдомлення \n\n\n");

// ------------------------------------------------------------

// розшифроване повiдомлення M2

int M2[size\_key1];

for (int i = 0; i < size\_key1; i++) {

M2[i] = (C1[i] - G1[i]) % size\_ABC;

while (M2[i] < 0){

M2[i] = M2[i] + size\_ABC;

}

}

for (int i = 0; i < size\_mes; i++) {

printf("%d ", M2[i]);

}

printf(" - розшифроване повiдомлення(M2) \n\n");

for (int i = 0; i < size\_mes; i++) {

for (int j = 0; j < size\_ABC; j++) {

if (M2[i] == j) {

decrypted\_mes[i] = ABC[j];

}

}

}

for (int i = 0; i < size\_mes; i++) {

printf("%c", decrypted\_mes[i]);

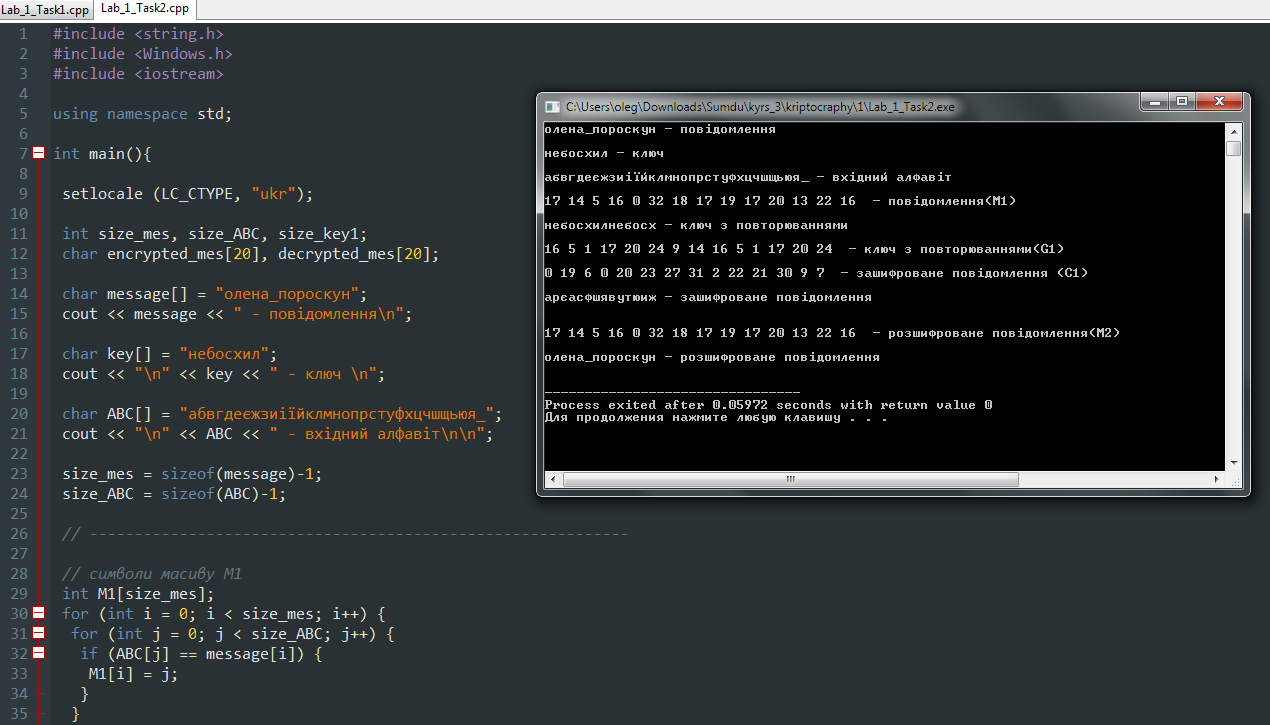
}

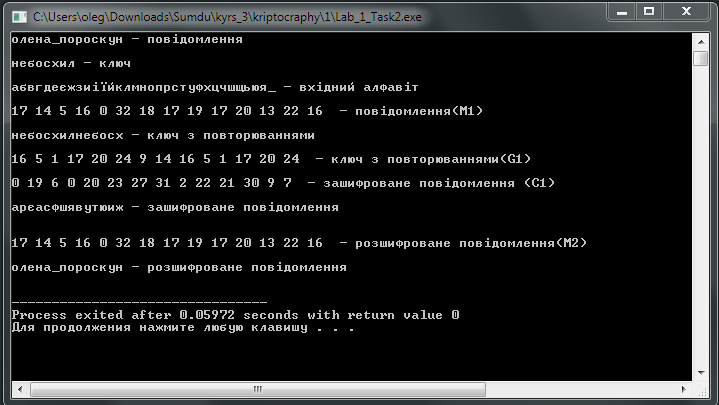
printf(" - розшифроване повiдомлення \n\n");

return 0;

}

***Прінт-скрін програми:***





***Зміст задачі:***

3. Зашифруйте повідомлення «2, 7, 9, D, A, 3, 8, D, C, 1, A», яке подано в шістнадцятковій системі числення, потоковим методом, використовуючи ключ

{0110, 1110, 1010, 1000, 1011, 1001, 1111, 0011, 1100, 1000, 1101}.

Знайдіть повну множину ключів і безпечний час такої криптосистеми, якщо символи ключа з’являються рівноймірно і незалежно, у = 10 вар/с. Розшифруйте повідомлення.

**Розв’язування задачі**

*Розмір простору ключів*

(де m - основа алфавіту, - розмір ключа (довжина)).

*Ентропія мови повідомлення*

*де - ентропія мови, де всі символи рівноймовірні і незалені.*

*Безпечний час*

при = 1.

***Програмний код:***

#include <string.h>

#include <Windows.h>

#include <iostream>

using namespace std;

int main(){

setlocale (LC\_CTYPE, "ukr");

int size\_message, size\_ABC, size\_key;

char encrypted\_mes[20], decrypted\_mes[20];

// повiдомлення "2, 7, 9, D, A, 3, 8, D, C, 1, A",

char message[] = "279DA38DC1A";

cout << message << " - повідомлення\n";

size\_message = sizeof(message) - 1;

//cout << size\_message << " - size\_message \n";

// ключ - {0110, 1110, 1010, 1000, 1011, 1001, 1111, 0011, 1100, 1000, 1101}.

char key[11][5] = {"0110", "1110", "1010", "1000", "1011", "1001", "1111", "0011", "1100", "1000", "1101"} ;

size\_key = sizeof( key ) / sizeof( key[0] );

//cout << size\_key << " - size\_key \n";

for (int i = 0; i < size\_key; i++) {

printf("%s ", key[i]);

}

printf(" - ключ(key) \n\n\n");

char ABC1[16][5] = {"0000", "0001", "0010", "0011", "0100", "0101", "0110", "0111",

"1000", "1001", "1010", "1011", "1100", "1101", "1110", "1111"};

char ABC2[] = "0123456789ABCDEF";

size\_ABC = sizeof( ABC1 ) / sizeof( ABC1[0] );

//cout << size\_ABC << " - size\_ABC \n";

for (int i = 0; i < size\_ABC; i++) {

printf("%s ", ABC1[i]);

}

printf(" - ABC1 \n\n");

for (int i = 0; i < size\_ABC; i++) {

printf("%c ", ABC2[i]);

}

printf(" - ABC2 \n\n\n");

// ------------------------------------------------------

// ключ в 16-вiй системi

char key1[11];

for (int i = 0; i < size\_key; i++){

for (int j = 0; j < size\_ABC; j++){

if (ABC1[j][0] == key[i][0] && ABC1[j][1] == key[i][1] && ABC1[j][2] == key[i][2] && ABC1[j][3] == key[i][3]) {

key1[i] = ABC2[j];

}

}

}

for (int i = 0; i < size\_key; i++) {

printf("%c ", key1[i]);

}

printf(" - ключ в 16-вiй системi (key1) \n\n");

printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n\n");

// ------------------------------------------------------------

// символи масиву M1

int M1[size\_message];

for (int i = 0; i < size\_message; i++) {

for (int j = 0; j < size\_ABC; j++) {

if (ABC2[j] == message[i]) {

M1[i] = j;

}

}

}

for (int i = 0; i < size\_message; i++) {

printf("%d ", M1[i]);

}

printf(" - повiдомлення(M1)\n\n");

// ------------------------------------------------------------

// символи масиву ключа Г1

int G1[size\_key];

for (int i = 0; i < size\_message; i++) {

for (int j = 0; j < size\_ABC; j++) {

if (ABC2[j] == key1[i]) {

G1[i] = j;

}

}

}

for (int i = 0; i < size\_message; i++) {

printf("%d ", G1[i]);

}

printf(" - ключ(G1) \n");

printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n\n\n");

// ------------------------------------------------------------

// символи криптограми С1

int C1[size\_key];

for (int i = 0; i < size\_key; i++) {

C1[i] = (M1[i] + G1[i]) % size\_ABC;

}

for (int i = 0; i < size\_message; i++) {

printf("%d ", C1[i]);

}

printf(" - зашифроване повiдомлення (C1) \n\n");

// зашифроване повiдомлення

for (int i = 0; i < size\_message; i++) {

for (int j = 0; j < size\_ABC; j++) {

if (C1[i] == j) {

encrypted\_mes[i] = ABC2[j];

}

}

}

for (int i = 0; i < size\_message; i++) {

printf("%c ", encrypted\_mes[i]);

}

printf(" - зашифроване повiдомлення в 16-вiй системi \n\n");

printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n\n\n");

// ------------------------------------------------------------

// розшифроване повiдомлення M2

int M2[size\_key];

for (int i = 0; i < size\_key; i++) {

M2[i] = (C1[i] - G1[i]) % size\_ABC;

while (M2[i] < 0){

M2[i] = M2[i] + size\_ABC;

}

}

for (int i = 0; i < size\_message; i++) {

printf("%d ", M2[i]);

}

printf(" - розшифроване повiдомлення(M2) \n\n");

// розшифроване повiдомлення

for (int i = 0; i < size\_message; i++) {

for (int j = 0; j < size\_ABC; j++) {

if (M2[i] == j) {

decrypted\_mes[i] = ABC2[j];

}

}

}

for (int i = 0; i < size\_message; i++) {

printf("%c ", decrypted\_mes[i]);

}

printf(" - розшифроване повiдомлення в 16-вiй системi \n\n");

// 16 система

/\*

0000 0

0001 1

0010 2

0011 3

0100 4

0101 5

0110 6

0111 7

1000 8

1001 9

1010 A

1011 B

1100 C

1101 D

1110 E

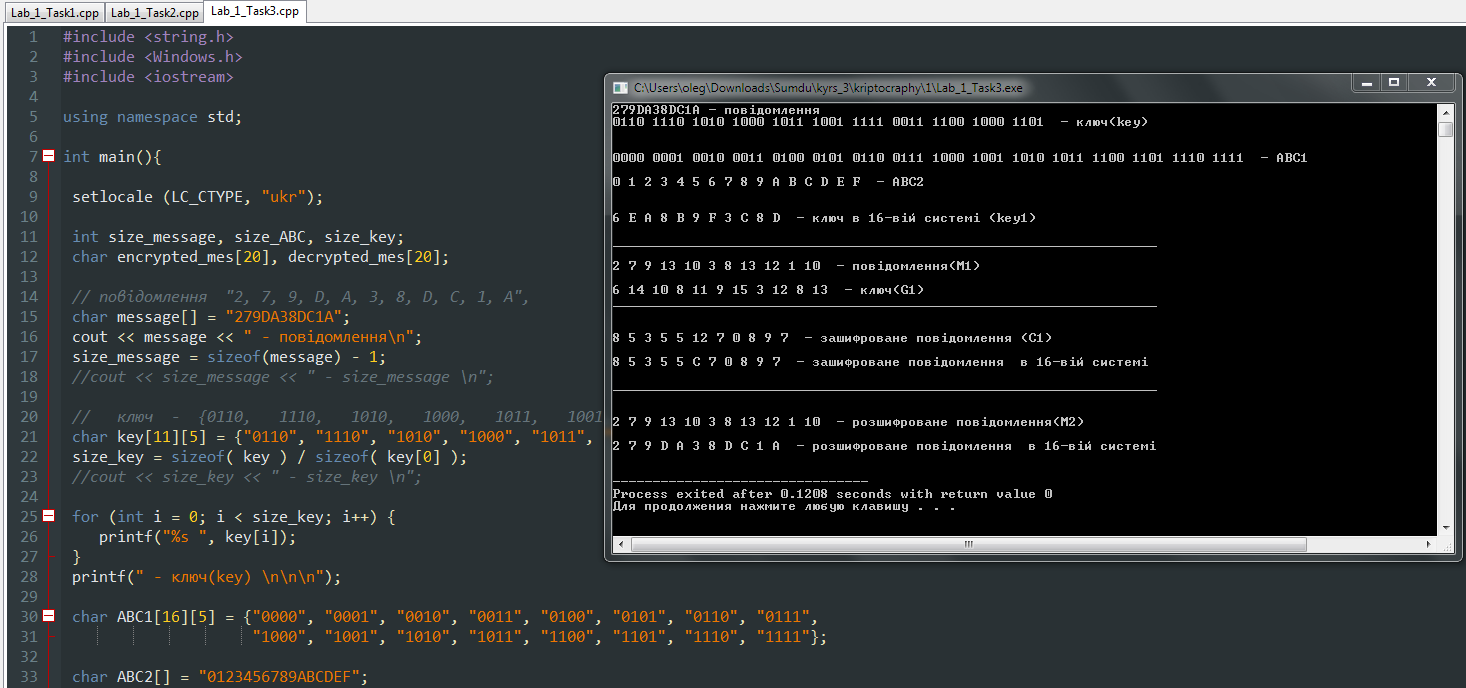
1111 F

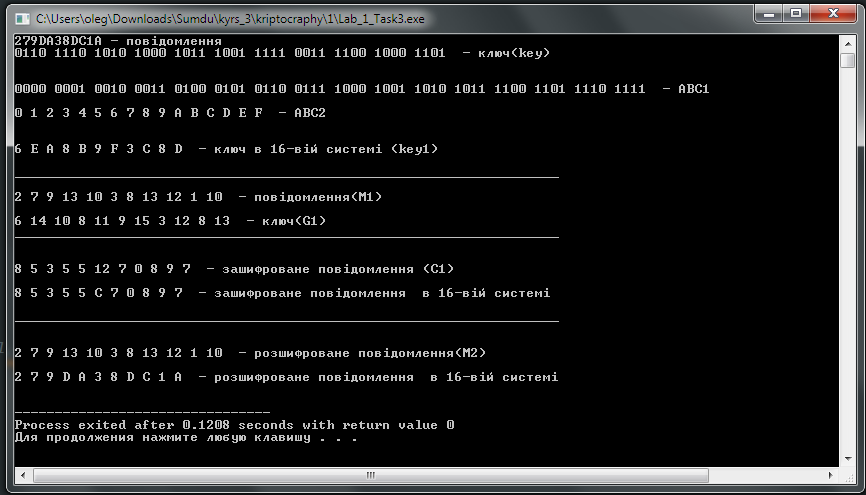
\*/

return 0;

}

***Прінт-скрін програми:***

******

******

***Контрольне питання***

9. Дати визначення блокового шифру.

Блочними називаються шифри, в яких логічною одиницею шифрування є деякий блок відкритого тексту, після перетворення якого виходить блок шифрованого тексту такої ж довжини. Зазвичай використовуються блоки довжиною 64 біта (128).

Нехай блочний шифр оперує з блоками довжини n. Всього таких блоків 2\*\*n. Оборотних перетворень блоків довжини n у блоки такого ж розміру всього (2\*\*n)!. Якщо n невелике, то такий шифр еквівалентний шифру підстановки на відповідному алфавіті і нестійкий до частотного аналізу. Чем більше n, тим менш ефективна статистична атака. Виписати таблицю перетворення блоків довжини 64 біта не є можливим.

Блоковий шифр зазвичай складається з простих перетворень над відкритим текстом, що виконуються в певній послідовності деяку кількість разів. Ці перетворення дають змогу усунути або істотно зменшити статистичну складову інформації та залежності відкритого тексту, тобто підвищити його ентропію до такого значення, коли між тим, що існує на вході криптографічного алгоритму, та тим, що отримано на виході, не спостерігається зв’язку. У більшості випадків використані операції та перетворення повинні мати обернені до себе

*.*

У такому випадку виконання ряду операцій та перетворень, що мають обернені, над відкритим текстом теж матимуть зворотну операцію, що буде набором обернених перетворень, застосованих у зворотному порядку

***Висновок***

Виконуючи лабораторну роботу, навчилися застосовувати метод простої підстановки, шифр Віженера та потоковий метод в 16-вій системі числення на практиці, складати алгоритм та програму розв’язування цих задач, а також обчислюватирозмір простору ключів,ентропію джерела ключів/мови повідомлення**,** безпечний час**,** відстань єдності для шифру криптосистеми.

***Додаток***

