**МИНИСТЕРСТВО ПО РАЗВИТИЮ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И КОММУНИКАЦИЙ РЕСПУБЛИКИ УЗБЕКИСТАНА**

ТАШКЕНТСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

***Практическая работа 3***

*по предмету* ***«Криптография»***

На тему: **«**Программные реализации классических шифров**»**

Группа: 721-19

Выполнил: Хакимбеков Дониёрбек

Принял: Хамидов Шерзод

***Ташкент 2021***

**Шифр Цезаря**

#include<iostream>

#include<Windows.h>

#include<string>

using namespace std;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int i=0, j=0, k;

char ABC[]={'А','Б','В','Г','Д','Е','Ж','З','И',

'Й','К','Л','М','Н','О','П','Р','С',

'Т','У','Ф','Х','Ц','Ч','Ш','Щ','Ъ',

'Ы','Ь','Э','Ю','Я','\_','.',',','0',

'1','2','3','4','5','6','7','8','9'};

char msg[80];

cout<<"Введите сообщение, которое будет закодировано: "<<endl;

cin>>msg;

cout<<"\nВведите сдвиг:"<<endl;

cin>>k;

cout<<endl;

for (i=0; i<strlen(msg); i++)

{ if(i==' ') msg[i]=' ';

else{

for (j = 0; j < 45; j++)

{

if (msg[i] == ABC[j])

{

if (j >= 45)

msg[i] = ABC[i-45];

else

msg[i] = ABC[j+k];

break;

}

}}

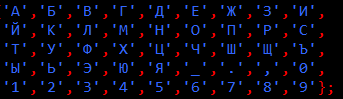
}

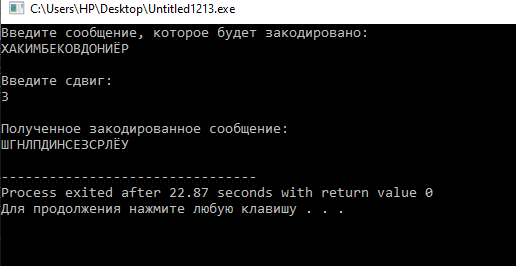
cout<<"Полученное закодированное сообщение: "<<endl;

cout<<msg<<endl;

return 0;

}





**Шифр Вернама**

#include <iostream>

#include <conio.h> //Для оформления программы

#include <windows.h>

#include <string> //Класс строки

#include <cstdlib>

#include <cmath>

void clrscr(){

system("cls");

}

using namespace std;

typedef unsigned int uni; //Вводим собственное обозначение типа данных (unsigned int = uni)

//Класс строки представленной в виде двочного кода для русского алфавита

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

class cystr

{

public:

int dem; //размерность строки

int \*\*s\_c; //шифрованная строка в двоичном коде

int \*\*open; //открытая строка в двоичном коде

static string alph; //статическая поле класса (алфавит) (string - тип данных)

cystr ();

cystr (int \*\*, int);

~cystr();

};

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Описание статического поля класса и методов

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

string cystr::alph = "АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯабвгдежзий\

клмнопрстуфхцчшщъыьэюя";

cystr::cystr(){}

cystr::cystr(int \*\*p, int n)

{

dem = n;

s\_c = new int \*[n];

open = new int \*[n];

for(int i = 0; i < n; i++)

{

s\_c[i] = p[i];

open[i] = p[i];

}

}

cystr::~cystr()

{

dem = 0;

delete [] open;

delete [] s\_c;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Класс ключа для шифровки и дешифровки

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

class key

{

int dem;

int \*\*s\_k; // Строка ключа в двоичном коде

public:

key();

key(int \*\*, int);

~key();

void enc (cystr &); //Метод шифрования строки (ничего не возвращает просто шифрует строку)

void dec (cystr &); //передаём в функцию значения по адресу

friend int \*\* encode (string);

};

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Описание методов шифровани и дешифрования

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

key::key(int \*\*p, int n)

{

dem = n;

s\_k = new int \*[n];

for(int i = 0; i < n; i++)

s\_k[i] = p[i];

}

key::~key()

{

dem = 0;

delete [] s\_k;

}

void key::enc (cystr &toc)

{

for(int i = 0; i < dem; i++)

for(int j = 0; j < 6; j++)

toc.s\_c[i][j] = (toc.open[i][j] + s\_k[i][j])%2;

}

void key::dec (cystr &tod)

{

for(int i = 0; i < dem; i++)

for(int j = 0; j < 6; j++)

tod.open[i][j] = (tod.s\_c[i][j]+s\_k[i][j])%2;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Функциии кодирования в двоичный код строки и обратно

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int \*\* encode (string s)

{

char err[] = "Ошибка кодирования. Неизвестный символ";

CharToOem(err,err);

int \*temp; // временная переменная для хранения промежуточных результатов

temp = new int [s.size()];

for(uni i = 0; i < s.size(); i++)

{

temp[i] = cystr::alph.find(s[i], 0); //temp - одномерный массив для хранения строки в виде десятичных чисел (от 0 до 63)

if(temp[i] == -1)

{

cout<<err<<endl;

system("pause");

abort();

}

}

int \*\*p;

p = new int \*[s.size()];

for(uni i = 0; i < s.size(); i++)

p[i] = new int [6];

for(uni i = 0; i < s.size(); i++)

for(int j = 0; j < 6; j++)

{

p[i][5-j] = temp[i]%2;

temp[i] = int(temp[i]/2);

}

return p;

}

char \* decode (int \*\*p, int n)

{

int \*temp;

temp = new int [n];

for(int i = 0; i < n; i++)

{

temp[i] = 0;

for(int j = 0; j < 6; j++)

temp[i] += p[i][j]\*pow(2.0,(5-j));

}

char \* s;

s = new char [n+1];

for(int i = 0; i < n; i++)

s[i] = cystr::alph[temp[i]];

s[n] = '\0';

return s;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Функция управления шифрованием (интерфейс)

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void action1 ()

{

char \*p;

uni q;

char buf [256];

string s;

char instr[] = "Введите строку без пробелов";

char inkey1[] = "Введите ключ шифрования такой же длинны";

char encyp\_suc[] = "Строка зашифрована";

char iner[] = "Ошибка при вводе. Привышена размерность";

CharToOem(iner,iner);

CharToOem(encyp\_suc,encyp\_suc);

CharToOem(inkey1,inkey1);

CharToOem(instr,instr);

cout<<instr<<endl;

cin>>buf;

OemToChar(buf,buf);

s = buf;

q = s.size();

cystr sc1(encode(s),s.size()); //sc1 - объект класса cystr

cout<<inkey1<<endl;

cin>>buf;

OemToChar(buf,buf);

s = buf;

if(q != s.size())

{

cout<<iner<<endl;

system("pause");

abort();

}

key o(encode(s),s.size()); //o - объект класа key

o.enc(sc1);

cout<<encyp\_suc<<endl;

p = decode(sc1.s\_c, sc1.dem);

CharToOem(p,p);

cout<<p<<endl;

system("pause");

clrscr(); //очистка экрана

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Функция управления дешифрованием (интерфейс)

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void action2 ()

{

uni q;

char \*p;

char buf [256];

string s;

char decyp\_suc[] = "Строка дешифрована";

char instr[] = "Введите строку без пробелов";

char inkey2[] = "Введите ключ дешифрования такой же длинны";

char iner[] = "Ошибка при вводе. Привышена размерность";

CharToOem(iner,iner);

CharToOem(inkey2,inkey2);

CharToOem(instr,instr);

CharToOem(decyp\_suc,decyp\_suc);

cout<<instr<<endl;

cin>>buf;

OemToChar(buf,buf);

s = buf;

q = s.size();

cystr sc1(encode(s),s.size());

cout<<inkey2<<endl;

cin>>buf;

OemToChar(buf,buf);

s = buf;

if(q != s.size())

{

cout<<iner<<endl;

system("pause");

abort();

}

key o(encode(s),s.size());

o.dec(sc1);

cout<<decyp\_suc<<endl;

p = decode(sc1.open, sc1.dem);

CharToOem(p,p);

cout<<p<<endl;

system("pause");

clrscr();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Главная функция

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int main()

{

int q;

char buf [256];

char welcome[] = "Вас приветствует программа КРИПТО 0.0.1 (Шифр Вернама)";

char qes[] = "Выбирите действие\n1 - Шифрование\n2 - Дешифрование\n3 - Выход";

char er[] = "Неверно";

CharToOem(welcome,welcome);

CharToOem(qes,qes);

CharToOem(er,er);

cout<<welcome<<endl<<endl;

while(q != '3')

{

cout<<qes<<endl;

cin>>q;

getchar();

switch(q)

{

case 1: action1(); break;

case 2: action2(); break;

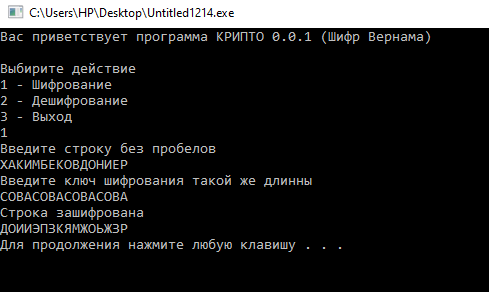
case 3: return 0;

default: cout<<er<<endl; system("pause"); clrscr();

}

}

}



**Шифр Виженера**

#include <iostream> // For Input and Output

#include <fstream> // For file input and output

#include <string.h>

using namespace std;

const int wordLength = 81;

void encodeText(char text[wordLength], char key[wordLength]){

int textLen = strlen(text);

int keyLen = strlen(key);

int i, j;

char newKey[textLen];

char encryptedText[textLen];

//новый ключ

for(i = 0, j = 0; i < textLen; ++i, ++j){

if(j == keyLen){

j = 0;

}

newKey[i] = key[j];

}

newKey[i] = '\0';

//шифровка

for(i = 0; i < textLen; i++){

if (text[i] != ' '){

encryptedText[i] = ((text[i] + newKey[i]) % 26);

encryptedText[i] += 'a';

}

else{

encryptedText[i] = ' ';

}

}

encryptedText[i] = '\0';

cout << newKey << endl;

cout << text << endl;

cout << encryptedText << endl;

}

int main(){

cout << "Vvedite tekst dlya shifrovaniya: ";

char textToEncode[wordLength];

cin.getline(textToEncode, wordLength);

cout << "Vvedi klyuch: ";

char keyword[wordLength];

cin.getline(keyword,wordLength );

cout << "Klyuch,tekst i shifr : \n";

encodeText(textToEncode, keyword);

return 0;

}

