МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

## Факультет информационных технологий и робототехники

Кафедра программного обеспечения вычислительной техники

и автоматизированных систем

**Отчет по лабораторной работе № 1**

по дисциплине: ”Теория информации”

на тему: ***”*Изучение криптографических систем, основанных на методе подстановки*”***

Выполнили**:** студенты группы *10701118:*

И.А.Воробей

Д.В. Алейников

Проверил доцент**:** А.А. Нессенчук

Минск 2019

Цель работы: Изучение способов шифрования информации в криптографических системах, основанных на методе подстановки.

1 Краткие теоретические сведения

Криптографические системы, основанные на методе подстановки, разделяются на четыре основных класса (monoalphabetic, homophonic, polyalphabetic, polygram).

В системах класса monoalphabetic символ исходного текста заменяется другим символом таким образом, что между ними существует однозначное соответствие. То есть каждый символ исходного текста однозначно заменяется его подстановкой. Криптографическим ключом такой системы является таблица соответствия исходного алфавита алфавиту подстановки. Например, для английского алфавита существует 26!=4\*1026 различных криптографических систем первого класса. Наиболее простые системы данного класса предполагают аналитическое описание подстановок. Так, простейший шифратор, основанный на принципе подстановки, сдвигает каждую букву английского алфавита на k позиций, где k является ключом шифра. В так называемом алгоритме Цезаря i-ая буква алфавита заменяется (i+k)-й буквой по модулю 26. Юлий Цезарь использовал подобную систему для k=3. Аналитически криптосистема Цезаря описывается выражением

Ek(i)=(i+k) mod 26

Например, в соответствии с приведенным выражением буква А исходного английского алфавита, имеющая номер i=0, заменяется буквой D, имеющей номер (i+k) mod 26=(0+3) mod 26=3, а буква Z (i=25) заменяется буквой C, имеющей номер (i+k) mod 26=(25+3) mod 26 = 2. Следующий пример иллюстрирует алогритм шифрования Цезаря:

Исходный текст: CRYPTOGRAPHYANDDATASECURITY.

Шифротекст: FUBSWRJUDSKBDQSGDWDVHFXULWB.

Алгоритм дешифрования имеет вид

Dk(i) = (i+26-k) mod 26

Существую более сложные методы подстановки. Шифраторы, основанные на умножении номера каждого символа исходного текста на значение ключа k, описываются следующим отношением:

Ek(i)=(i\*k) mod n, (1.1)

где i – номер символа исходного текста, n – количество символов в исходном алфавите (n=26 для английского алфавита и n=256 для ASCII-кодов),

k – ключ, n и k должны быть взаимно простыми.

Криптографические системы polyalphabetic основаны на использовании нескольких различных ключей. Большинство шифраторов подобного типа являются периодическими с периодом P. Исходный текст вида

X=x1x2x3x4..xpxp+1..x2p..

шифруется с помощью ключей k1,k2,…kp.

Криптосистемы третьего класса, основанные на полиалфавитной подстановке, широко использовались и используются на практике. На их основе разработано целое семейство роторных шифраторов, которые широко применялись во время второй мировой войны и в послевоенное время. Среди них можно выделить машину Хагелина М-209 (США), немецкую шифровальную машину «Энигма», японский «Пурпурный код».

Криптографические системы класса polygram характеризуются подстановкой не одного, а нескольких символов в исходном тексте. В общем случае n символов исходного текста заменяются n символами шифротекста.

## 2 Задание на выполнение работы

1) Реализовать алгоритмы и программу шифрования исходного текста (на базе произвольно выбранного алфавита) в системе класса monoalphabetic.

Алгоритм шифрования реализовать с использованием формулы см (1.1)

2) Реализовать алгоритм и программу дешифрования сформированного согласно пункту 1 шифротекста.

Алгоритм дешифрования должен быть разработан самостоятельно. Алгоритм может быть реализован, как на основе формулы, так и с использованием логико-смыслового анализа исходных данных задачи.

**3 Алгоритм решения поставленной задачи**

Ниже приведены блок-схемы работы программы. (см. Рисунки 1-3)

Блок 1: Нахождение НОК

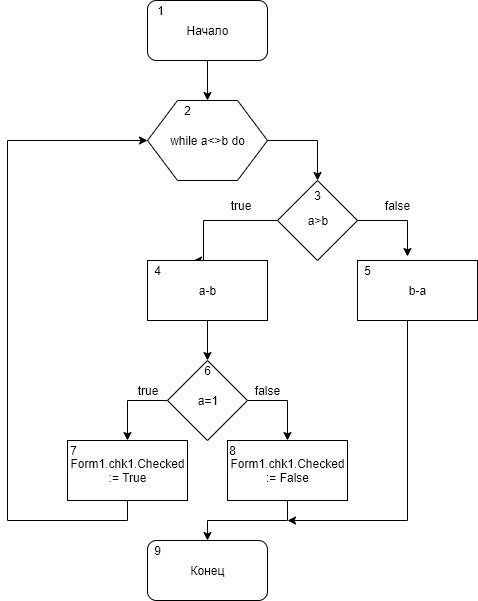


Рисунок 1 – Алгоритм нахождения НОД

**4 Пояснение к блок-схеме №1**

1)Начало программы.

2)Цикл для анализа данных.

3)Условие через которое будет выполняться нахождения нужных элементов.

4)Выполнение при правильных данных.

5)Выполнение при ошибочном значении.

6)Цикл для определения значения кнопки.

7)Кнопка нажата.

8)Кнопка не нажата.

9)Конец программы

Блок 2: Зашифровка исходных данных

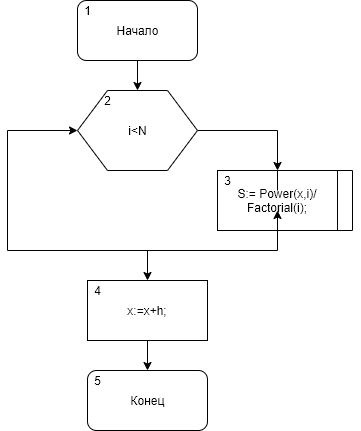


Рисунок 2 - Алгоритм зашифровка

**5 Пояснение к блок-схеме №2**

1)Начало программы.

2)Цикл проверяет условие.

3)Шифрование текста.

4)Продолжение шифрования.

5)Конец программы.

Блок 3: Дешифровка исходных данных

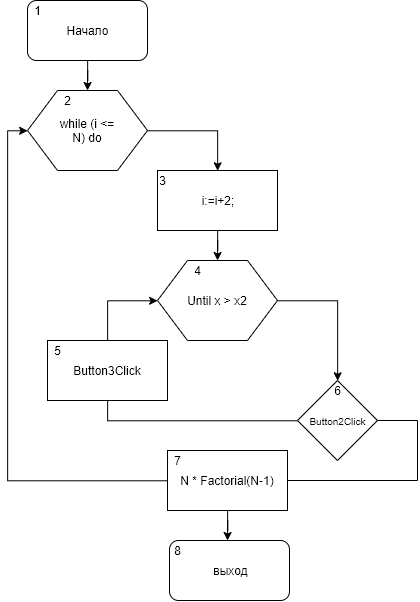


Рисунок 3 – Алгоритм шифрования

**6 Пояснение к блок-схеме №3**

1)Начало программы.

2)Цикл который содержит в себе пустой список и цикл дешифрования.

3)Создание пустого списка.

4)Цикл для дешифрования текста.

5)Условие для дешифровки.

6)Дешифровка.

7)Выход из цикла.

8)Выход из программы.

**7 Ниже приведен текст программы**

procedure NOK(a:Integer;b:Integer);

begin

while a<>b do

if a > b then a := a - b

else b := b - a;

if a = 1 then Form1.chk1.Checked := True

else Form1.chk1.Checked := False;

end;

procedure TForm1.btn1Click(Sender: TObject);

begin

k := StrToInt(edt5.Text);

f := 0;

NOK(k,26);

dt := '';

t := edt1.Text;

t := AnsiUpperCase(t);

if (chk1.Checked = True) or (chk2.Checked = True) then

begin

for i := 0 to Length(t) do

begin

for j := 0 to 25 do

begin

if alphabet[j] = t[i] then

begin

x := (j \* k + f) mod m;

dt := dt + alphabet[x];

end;

end;

if t[i] = ' ' then dt := dt + ' ';

end;

edt2.Text := dt;

chk2.Checked := False;

end

else form2.Show;

end;

procedure TForm1.FormCreate(Sender: TObject);

begin

m := 26;

//lbl7.Caption := 'm = ' + IntToStr(m);

lbl8.Caption := 'Ek(i) = (i \* ko) mod n; n =26'

end;

procedure TForm1.btn2Click(Sender: TObject);

begin

k := StrToInt(edt5.Text);

f := 0;

raz := '';

t := edt3.Text;

mistakes := '';

for i := 0 to Length(t) do

begin

mis := False;

for z:= 0 to 25 do

begin

y := (z \* k + f) mod m;

if alphabet[y] = t[i] then

begin

if mis then

begin

mistakes := mistakes + IntToStr(i) + ' ';

Break;

end

else

begin

raz := raz + alphabet[z];

mis := True;

end;

end;

end;

end;

if t[i] = ' ' then raz := raz + ' ';

if mistakes <> '' then

begin

form3.lbl1.Caption := 'Âîçìîæíû îøèáêè ðàñøèôðîâêè áóêâ ïîä íîìåðîì ' + #13 + #10 + mistakes;

form3.Show;

end;

edt4.Text := raz;

end;

procedure TForm1.btn3Click(Sender: TObject);

begin

Form2.show();

end;

end.

**8 Описание входных /выходных данных**

Вводим слово на английском языке в малом регистре (см Рисунок2). Верхнее окно выводим после введения нажимаем Enter и в следующей строке появляется закодированное слово.

В 3 окно вводим закодированное слово, после нажатия Deincription в следующей строке выводиться шифрование.

**9 Контрольный пример**

Ниже приведены скрины работы программы см.(Рисунок 4-6)

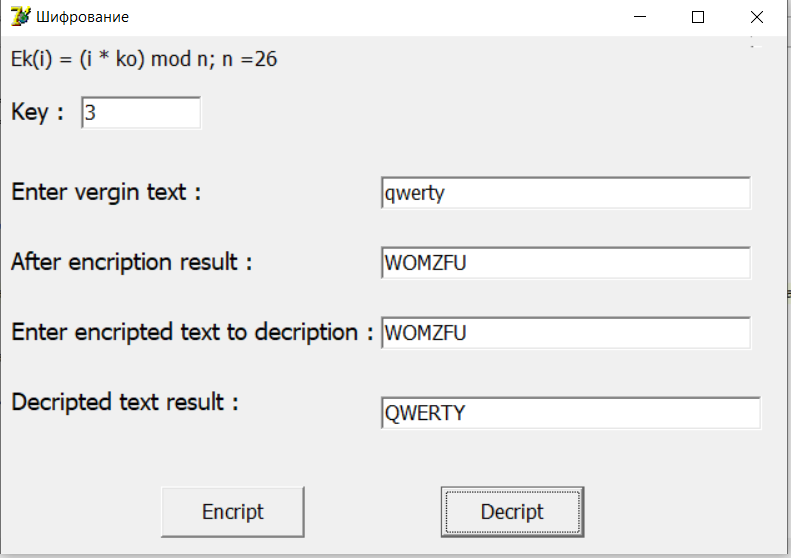


Рисунок 4 – Выполнение программы

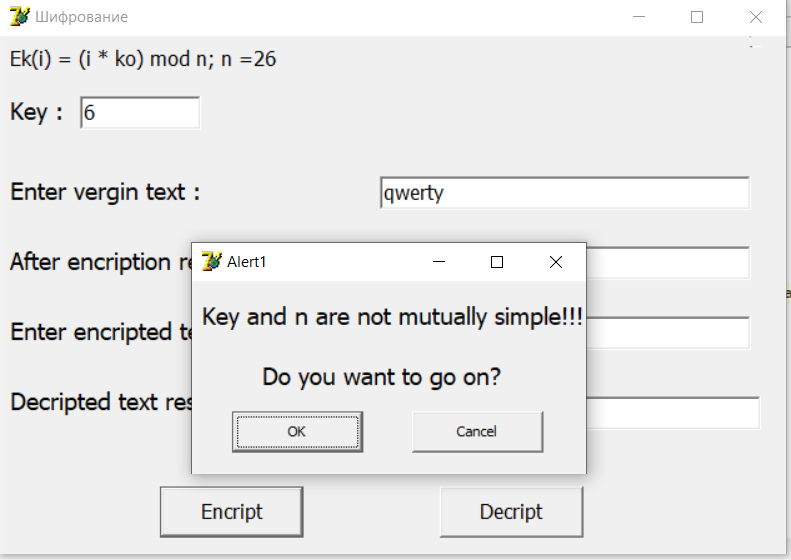
****

Рисунок 5 – Дополнительное окно

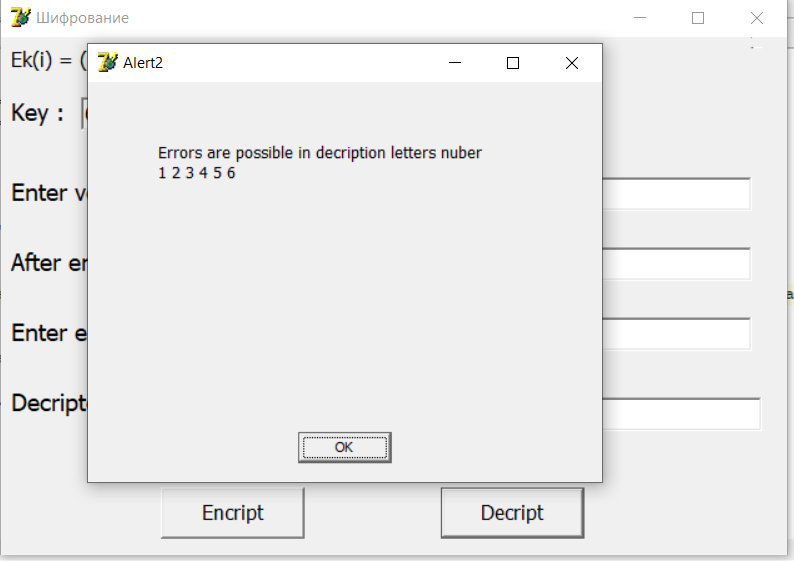


Рисунок 6 – Указание на ошибки

**10 Вывод**

1)Изучили способы шифрования информации в криптографических системах, основанных на методе подстановки.

2)Разработали программу для шифрования и дешифрования предложения

**11 Список литературы**

1)Шеннон К.''Работы по теории информации и кибернетике’’.

# 2)Романец Ю.В., Тимофеев П.А., Шаньгин В.Ф. “Защита информации в компьютерных системах и сетях”.

3)Чмора А.Л.” Современная прикладная криптография”.

4)Шнайер Б.”Прикладная криптография”.