**УКРАЇНИ МІНІСТЕРСТВО ОСВІТИ І НАУКИ**

**СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ**

**ЦЕНТР ЗАОЧНОЇ, ДИСТАНЦІЙНОЇ ТА ВЕЧІРНЬОЇ ФОРМ НАВЧАННЯ**

**КАФЕДРА КОМП’ЮТЕРНИХ НАУК**

**ПРАКТИЧНА РОБОТА № 2**

з дисципліни "Захист інформації"

Тема: **"Шифри cкладної заміни"**

Варіант № 13

**Перевірив Коробченко О.В.**

**Студент групи Індн – 31с Кролевецький П.А.**

Суми – 2017

**Мета:**

Виробити вміння та навички шифрування інформації за допомогою шифру Віженера

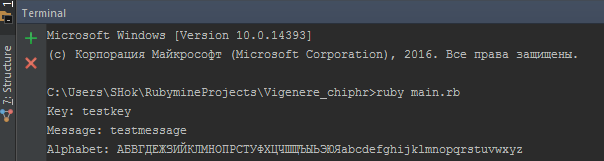
**Завдання:**

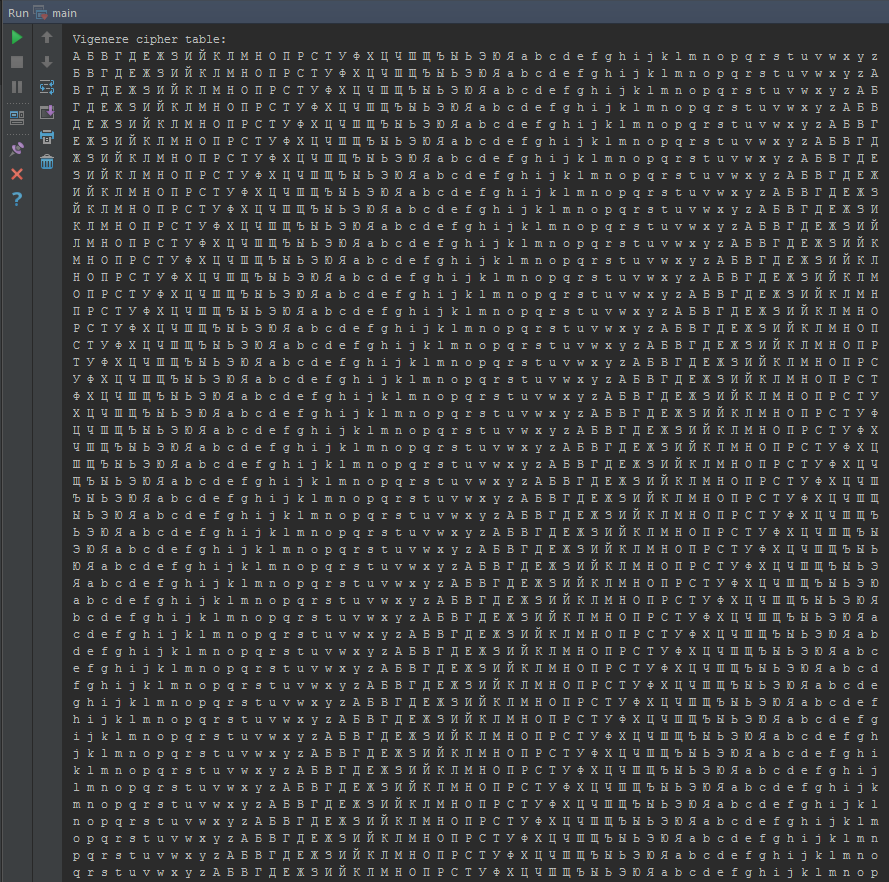
|  |  |  |
| --- | --- | --- |
| **№**  **в-та** | **А1** | **Тип шифрування** |
| 13 | Великі букви російського алфавіту і малі букви латинського алфавіту | За допомогою таблиці Віженера |

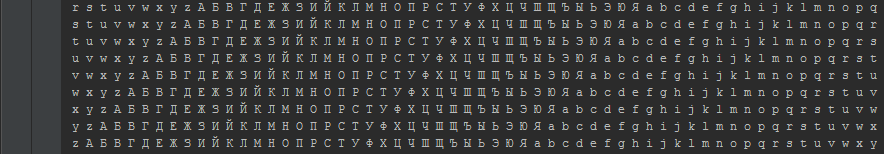
**Вихідні дані, які були застосовані для виконання поставленого завдання:**

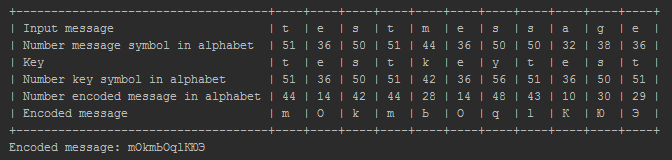
* Алфавіт що містить наступні символи АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯabcdefghijklmnopqrstuvwxyz
* Ключ – testkey
* Текст – testmessage

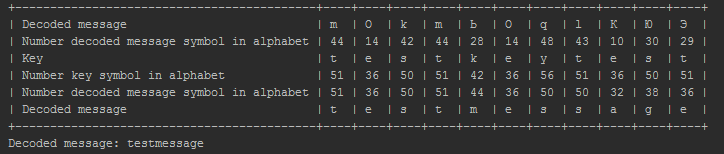
**Скриншоти роботи програми:**











**Вихідний текст програми:**

*#!/bin/env ruby  
  
# Библиотека для отрисовки текстовых таблиц*require 'terminal-table'  
  
**class** *Matrix* attr\_accessor :alphabet  
 attr\_accessor :matrix  
  
 **def** initialize(*alphabet* = [\*('А'..'Я'), \*('a'..'z')])  
 @alphabet = *alphabet* generate\_matrix(@alphabet)  
 **end  
  
 def** generate\_matrix(*alphabet*)  
 @matrix = []  
  
 **for** *matrix\_index* **in** 0..*alphabet*.size-1  
 *# alphabet.each\_index do |matrix\_index|* tmp\_alphabet = [].replace(*alphabet*) *# хак для копирования массива* **for** *iteration* **in** 0..*matrix\_index*-1  
 tmp\_alphabet.push(tmp\_alphabet.shift) *# переносим одну букву в конец массива* **end** @matrix[*matrix\_index*] = [].replace(tmp\_alphabet) *# хак для копирования массива* **end  
 end  
  
 def** print\_matrix  
 @matrix.each\_index **do** |*index*|  
 puts @matrix[*index*].join(' ')  
 **end  
 end  
  
 def** encode\_find\_symbol(*key\_symbol*, *text\_symbol*)  
 first\_symbol\_index = @alphabet.find\_index(*key\_symbol*)  
 second\_symbol\_index = @alphabet.find\_index(*text\_symbol*)  
  
 **return** @matrix[first\_symbol\_index][second\_symbol\_index]  
 **end  
  
  
 def** decode\_find\_symbol(*key\_symbol*, *text\_symbol*)  
 first\_symbol\_index = @alphabet.find\_index(*key\_symbol*)  
 second\_symbol\_index = @matrix[first\_symbol\_index].find\_index(*text\_symbol*)  
  
 **return** @alphabet[second\_symbol\_index]  
 **end  
  
 def** encode\_string(*key*, *str*)  
 information\_table\_rows = []  
 key\_long = *key  
  
 # Добавляем в таблицу перую строку* information\_table\_rows << ['Input message', \**str*.split('')]  
  
 *# Добавляем в таблицу вторую строку* input\_message\_index\_list = []  
 *str*.split('').each { |*item*| input\_message\_index\_list.push(@alphabet.find\_index(*item*)) }  
 information\_table\_rows << ['Number message symbol in alphabet', \*input\_message\_index\_list]  
  
 *# Увеличиваем ключ до длинны сообщения* itr = 0  
 **while** key\_long.length < *str*.length  
 key\_long += *key*[itr].to\_s  
  
 **if** itr > *key*.length-1  
 itr = 0  
 **else** itr += 1  
 **end  
 end** *# Добавляем в таблицу третью строку* information\_table\_rows << ['Key', \*key\_long.split('')]  
  
 *# Добавляем в таблицу четвёртую строку* key\_index\_list = []  
 key\_long.split('').each { |*item*| key\_index\_list.push(@alphabet.find\_index(*item*)) }  
 information\_table\_rows << ['Number key symbol in alphabet', \*key\_index\_list]  
  
 encoded\_string = ''  
 **for** *index* **in** 0..*str*.length-1  
 encoded\_string += encode\_find\_symbol(key\_long[*index*], *str*[*index*]).to\_s  
 **end** *# Добавляем в таблицу пятую строку* encoded\_message\_index\_list = []  
 encoded\_string.split('').each { |*item*| encoded\_message\_index\_list.push(@alphabet.find\_index(*item*)) }  
 information\_table\_rows << ['Number encoded message in alphabet', \*encoded\_message\_index\_list]  
  
 *# Добавляем в таблицу шестую строку* information\_table\_rows << ['Encoded message', \*encoded\_string.split('')]  
  
 information\_table\_object = Terminal::Table.new :rows *=>* information\_table\_rows  
  
 **return** encoded\_string, information\_table\_object  
 **end  
  
 def** decode\_string(*key*, *str*)  
 information\_table\_rows = []  
 key\_long = *key  
  
 # Добавляем в таблицу перую строку* information\_table\_rows << ['Decoded message', \**str*.split('')]  
  
 *# Добавляем в таблицу вторую строку* input\_message\_index\_list = []  
 *str*.split('').each { |*item*| input\_message\_index\_list.push(@alphabet.find\_index(*item*)) }  
 information\_table\_rows << ['Number decoded message symbol in alphabet', \*input\_message\_index\_list]  
  
 *# Увеличиваем ключ до длинны сообщения* itr = 0  
 **while** key\_long.length < *str*.length  
 key\_long += *key*[itr].to\_s  
  
 **if** itr > *key*.length-1  
 itr = 0  
 **else** itr += 1  
 **end  
 end** *# Добавляем в таблицу третью строку* information\_table\_rows << ['Key', \*key\_long.split('')]  
  
 *# Добавляем в таблицу четвёртую строку* key\_index\_list = []  
 key\_long.split('').each { |*item*| key\_index\_list.push(@alphabet.find\_index(*item*)) }  
 information\_table\_rows << ['Number key symbol in alphabet', \*key\_index\_list]  
  
 decoded\_string = ''  
 **for** *index* **in** 0..*str*.length-1  
 decoded\_string += decode\_find\_symbol(key\_long[*index*], *str*[*index*]).to\_s  
 **end** *# Добавляем в таблицу пятую строку* decoded\_message\_index\_list = []  
 decoded\_string.split('').each { |*item*| decoded\_message\_index\_list.push(@alphabet.find\_index(*item*)) }  
 information\_table\_rows << ['Number decoded message symbol in alphabet', \*decoded\_message\_index\_list]  
  
 *# Добавляем в таблицу шестую строку* information\_table\_rows << ['Decoded message', \*decoded\_string.split('')]  
  
 information\_table\_object = Terminal::Table.new :rows *=>* information\_table\_rows  
 **return** decoded\_string, information\_table\_object  
 **end  
  
end**key = 'testkey'  
message = 'testmessage'  
alphabet = [\*('А'..'Я'), \*('a'..'z')]  
  
puts 'Key: '+key  
puts 'Message: '+message  
puts 'Alphabet: '+alphabet.join+**"**\n\n**"**matrix = Matrix.new(alphabet)  
puts 'Vigenere cipher table:'  
matrix.print\_matrix  
puts **"**\n\n**"**encoded\_str, encode\_inform\_table = matrix.encode\_string(key, message)  
puts encode\_inform\_table  
puts 'Encoded message: '+encoded\_str+**"**\n\n**"**decoded\_str, decode\_inform\_table = matrix.decode\_string(key, encoded\_str)  
puts decode\_inform\_table  
puts 'Decoded message: '+decoded\_str

**Результати роботи програми:**

Key: testkey

Message: testmessage

Alphabet: АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯabcdefghijklmnopqrstuvwxyz

Vigenere cipher table:

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 a b c d e f g h i j k l m n o p q r s t u v w x y z

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 a b c d e f g h i j k l m n o p q r s t u v w x y z А

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 a b c d e f g h i j k l m n o p q r s t u v w x y z А Б

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 a b c d e f g h i j k l m n o p q r s t u v w x y z А Б В

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 a b c d e f g h i j k l m n o p q r s t u v w x y z А Б В Г

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 a b c d e f g h i j k l m n o p q r s t u v w x y z А Б В Г Д

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 a b c d e f g h i j k l m n o p q r s t u v w x y z А Б В Г Д Е

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 a b c d e f g h i j k l m n o p q r s t u v w x y z А Б В Г Д Е Ж

И Й К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ъ Ы Ь Э Ю Я a b c d e f g h i j k l m n o p q r s t u v w x y z А Б В Г Д Е Ж З

Й К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ъ Ы Ь Э Ю Я a b c d e f g h i j k l m n o p q r s t u v w x y z А Б В Г Д Е Ж З И

К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ъ Ы Ь Э Ю Я a b c d e f g h i j k l m n o p q r s t u v w x y z А Б В Г Д Е Ж З И Й

Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ъ Ы Ь Э Ю Я a b c d e f g h i j k l m n o p q r s t u v w x y z А Б В Г Д Е Ж З И Й К

М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ъ Ы Ь Э Ю Я a b c d e f g h i j k l m n o p q r s t u v w x y z А Б В Г Д Е Ж З И Й К Л

Н О П Р С Т У Ф Х Ц Ч Ш Щ Ъ Ы Ь Э Ю Я a b c d e f g h i j k l m n o p q r s t u v w x y z А Б В Г Д Е Ж З И Й К Л М

О П Р С Т У Ф Х Ц Ч Ш Щ Ъ Ы Ь Э Ю Я a b c d e f g h i j k l m n o p q r s t u v w x y z А Б В Г Д Е Ж З И Й К Л М Н

П Р С Т У Ф Х Ц Ч Ш Щ Ъ Ы Ь Э Ю Я a b c d e f g h i j k l m n o p q r s t u v w x y z А Б В Г Д Е Ж З И Й К Л М Н О

Р С Т У Ф Х Ц Ч Ш Щ Ъ Ы Ь Э Ю Я a b c d e f g h i j k l m n o p q r s t u v w x y z А Б В Г Д Е Ж З И Й К Л М Н О П

С Т У Ф Х Ц Ч Ш Щ Ъ Ы Ь Э Ю Я a b c d e f g h i j k l m n o p q r s t u v w x y z А Б В Г Д Е Ж З И Й К Л М Н О П Р

Т У Ф Х Ц Ч Ш Щ Ъ Ы Ь Э Ю Я a b c d e f g h i j k l m n o p q r s t u v w x y z А Б В Г Д Е Ж З И Й К Л М Н О П Р С

У Ф Х Ц Ч Ш Щ Ъ Ы Ь Э Ю Я a b c d e f g h i j k l m n o p q r s t u v w x y z А Б В Г Д Е Ж З И Й К Л М Н О П Р С Т

Ф Х Ц Ч Ш Щ Ъ Ы Ь Э Ю Я a b c d e f g h i j k l m n o p q r s t u v w x y z А Б В Г Д Е Ж З И Й К Л М Н О П Р С Т У

Х Ц Ч Ш Щ Ъ Ы Ь Э Ю Я a b c d e f g h i j k l m n o p q r s t u v w x y z А Б В Г Д Е Ж З И Й К Л М Н О П Р С Т У Ф

Ц Ч Ш Щ Ъ Ы Ь Э Ю Я a b c d e f g h i j k l m n o p q r s t u v w x y z А Б В Г Д Е Ж З И Й К Л М Н О П Р С Т У Ф Х

Ч Ш Щ Ъ Ы Ь Э Ю Я a b c d e f g h i j k l m n o p q r s t u v w x y z А Б В Г Д Е Ж З И Й К Л М Н О П Р С Т У Ф Х Ц

Ш Щ Ъ Ы Ь Э Ю Я a b c d e f g h i j k l m n o p q r s t u v w x y z А Б В Г Д Е Ж З И Й К Л М Н О П Р С Т У Ф Х Ц Ч

Щ Ъ Ы Ь Э Ю Я a b c d e f g h i j k l m n o p q r s t u v w x y z 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

Ъ Ы Ь Э Ю Я a b c d e f g h i j k l m n o p q r s t u v w x y z 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

Ы Ь Э Ю Я a b c d e f g h i j k l m n o p q r s t u v w x y z 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

Ь Э Ю Я a b c d e f g h i j k l m n o p q r s t u v w x y z 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

Э Ю Я a b c d e f g h i j k l m n o p q r s t u v w x y z 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

Ю Я a b c d e f g h i j k l m n o p q r s t u v w x y z 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

Я a b c d e f g h i j k l m n o p q r s t u v w x y z 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

a b c d e f g h i j k l m n o p q r s t u v w x y z 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

b c d e f g h i j k l m n o p q r s t u v w x y z 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 a

c d e f g h i j k l m n o p q r s t u v w x y z 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 a b

d e f g h i j k l m n o p q r s t u v w x y z 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 a b c

e f g h i j k l m n o p q r s t u v w x y z 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 a b c d

f g h i j k l m n o p q r s t u v w x y z 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 a b c d e

g h i j k l m n o p q r s t u v w x y z 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 a b c d e f

h i j k l m n o p q r s t u v w x y z 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 a b c d e f g

i j k l m n o p q r s t u v w x y z 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 a b c d e f g h

j k l m n o p q r s t u v w x y z 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 a b c d e f g h i

k l m n o p q r s t u v w x y z 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 a b c d e f g h i j

l m n o p q r s t u v w x y z 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 a b c d e f g h i j k

m n o p q r s t u v w x y z 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 a b c d e f g h i j k l

n o p q r s t u v w x y z 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 a b c d e f g h i j k l m

o p q r s t u v w x y z 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 a b c d e f g h i j k l m n

p q r s t u v w x y z 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 a b c d e f g h i j k l m n o

q r s t u v w x y z 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 a b c d e f g h i j k l m n o p

r s t u v w x y z 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 a b c d e f g h i j k l m n o p q

s t u v w x y z 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 a b c d e f g h i j k l m n o p q r

t u v w x y z 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 a b c d e f g h i j k l m n o p q r s

u v w x y z 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 a b c d e f g h i j k l m n o p q r s t

v w x y z 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 a b c d e f g h i j k l m n o p q r s t u

w x y z 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 a b c d e f g h i j k l m n o p q r s t u v

x y z 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 a b c d e f g h i j k l m n o p q r s t u v w

y z 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 a b c d e f g h i j k l m n o p q r s t u v w x

z 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 a b c d e f g h i j k l m n o p q r s t u v w x y

+------------------------------------+----+----+----+----+----+----+----+----+----+----+----+

| Input message | t | e | s | t | m | e | s | s | a | g | e |

| Number message symbol in alphabet | 51 | 36 | 50 | 51 | 44 | 36 | 50 | 50 | 32 | 38 | 36 |

| Key | t | e | s | t | k | e | y | t | e | s | t |

| Number key symbol in alphabet | 51 | 36 | 50 | 51 | 42 | 36 | 56 | 51 | 36 | 50 | 51 |

| Number encoded message in alphabet | 44 | 14 | 42 | 44 | 28 | 14 | 48 | 43 | 10 | 30 | 29 |

| Encoded message | m | О | k | m | Ь | О | q | l | К | Ю | Э |

+------------------------------------+----+----+----+----+----+----+----+----+----+----+----+

Encoded message: mОkmЬОqlКЮЭ

+-------------------------------------------+----+----+----+----+----+----+----+----+----+----+----+

| Decoded message | m | О | k | m | Ь | О | q | l | К | Ю | Э |

| Number decoded message symbol in alphabet | 44 | 14 | 42 | 44 | 28 | 14 | 48 | 43 | 10 | 30 | 29 |

| Key | t | e | s | t | k | e | y | t | e | s | t |

| Number key symbol in alphabet | 51 | 36 | 50 | 51 | 42 | 36 | 56 | 51 | 36 | 50 | 51 |

| Number decoded message symbol in alphabet | 51 | 36 | 50 | 51 | 44 | 36 | 50 | 50 | 32 | 38 | 36 |

| Decoded message | t | e | s | t | m | e | s | s | a | g | e |

+-------------------------------------------+----+----+----+----+----+----+----+----+----+----+----+

Decoded message: testmessage

**Висновки:**

При виконанні роботи побудовано таблицю Віженера для алфавіту: АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯabcdefghijklmnopqrstuvwxyz, зашифровано вихідний текст: testmessage за допомогою ключа testkey. Отримано зашифрований текст: mОkmЬОqlКЮЭ. Проведено дешифрування зашифрованого тексту за допомогою ключа testkey. При дешифруванні отримано вихідний текст: testmessage. В ході виконання роботи отримані навички шифрування інформації за допомогою шифру Віженера.