# РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

### Факультет физико-математических и естественных наук

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2 =============== ## Тема: Шифры перестановки дисциплина: Математические основы защиты информации и информационной безопасности

Студент: Койфман Кирилл Дмитриевич

Группа: НФИмд-01-25

## Введение

### Цель работы

Получение практических навыков реализации алгоритмов, применяющих метод перестановки.

### Задачи

1. Реализовать алгоритм маршрутного шифрования
2. Реализовать алгоритм шифрования Виженера

## Теория

Данный способ шифрования разработал французский математик Франсуа Виет. Открытый текст записывают в некоторую геометрическую фигуру (обычно прямоугольник) по некоторому пути, а затем, выписывая символы по другому пути, получают шифртекст. Пусть m и n — целые положительные числа, большие 1. Открытый текст разбивается на блоки равной длины, состоящие из числа символов, равному произведению mn. Если последний блок получится меньше остальных, то в него следует дописать требуемое количество произвольных символов. Составляется таблица размерности mn. Блоки вписывается построчно в таблицу. Криптограмма получается выписыванием букв из таблицы в соответствии с некоторым маршрутом. Ключом такой криптограммы является маршрут и числа m и n. Обычно буквы выписывают по столбцам, которые упорядочивают согласно паролю: внизу таблицы приписывается слово из n неповторяющихся букв и столбцы нумеруются по алфавитному порядку букв пароля.

При работе шифра Виженера открытый текст разбивается на блоки длины n. Ключ представляет собой последовательность из n натуральных чисел: а\_1, а\_2, …, а\_n. Далее в каждом блоке первая буква циклически сдвигается вправо по алфавиту на а\_1 позиций, вторая буква — на а\_2 позиций, последняя — на а\_n позиций. Для лучшего запоминания в качестве ключа можно взять осмысленное слово, а алфавитные номера входящих в него букв использовать для осуществления сдвигов.

## Ход работы

## ### 1 задание

Для решения поставленной задачи - реализации алгоритма маршрутного шифрования воспользуемся языком Julia, чтобы написать скрипт, который будет выполнять шифрование вводимого сообщения по заданным параметрам алгоритма (Листинг-1):

println("Route cipher's work demonstration\n--------------------------------------")  
println("[!!!]Please, follow the instructions!")  
  
print("\nEnter open message: ")  
openMessage::String = "нельзянедооцениватьпротивника"  
println("Entered open message: $(openMessage)")  
openMessage\_chars = collect(openMessage)  
  
print("\nEnter parameter n: ")  
parm\_n::Int32 = 6  
println("Entered parameter n value: $(parm\_n)")  
  
if length(openMessage\_chars) % parm\_n != 0  
 padding\_needed = parm\_n - (length(openMessage\_chars) % parm\_n)  
 openMessageLastSymbol = openMessage\_chars[length(openMessage\_chars)]  
 openMessage \*= string(openMessageLastSymbol)^padding\_needed  
 openMessage\_chars = collect(openMessage)  
 println("The source message was added extra symbols: $openMessage")  
end  
  
parm\_m::Int32 = length(openMessage\_chars) ÷ parm\_n  
println("Parameter m value: $(parm\_m)")  
  
  
print("\nEnter password(without repeating symbols & must equals parm-n): ")  
password::String = "пароль"  
println("Entered password: $(password)")  
password\_chars = collect(password)  
  
  
password\_order = sortperm(password\_chars)  
println("Columns order: ", password\_order)  
  
function encrypt()  
 encrypted\_message = ""  
  
 #access password symbol culumn  
 for col\_idx in password\_order  
 #collect each symbol from the password symbol column   
 for row in 0:(parm\_m-1)  
 index = row \* parm\_n + col\_idx  
 if index <= length(openMessage\_chars)  
 encrypted\_message \*= openMessage\_chars[index]  
 end  
 end  
 end  
  
 return encrypted\_message  
end  
  
result = encrypt()  
println("Encrypted message: ", result)

*Листинг-1(фрагмент алгоритма, реализующего метод маршрутного шифрования)*

Route cipher's work demonstration  
--------------------------------------  
[!!!]Please, follow the instructions!  
  
Enter open message: Entered open message: нельзянедооцениватьпротивника  
  
Enter parameter n: Entered parameter n value: 6  
The source message was added extra symbols: нельзянедооцениватьпротивникаа  
Parameter m value: 5  
  
Enter password(without repeating symbols & must equals parm-n): Entered password: пароль  
Columns order: [2, 5, 4, 1, 3, 6]  
Encrypted message: еенпнзоатаьовокннеьвлдирияцтиа

*Листинг-2(результаты работы алгоритма маршрутного шифрования)*

Исходя из полученных результатов (Листинг-2), можно судить о том, что данный алгоритм успешно провёл шифрование вводимого текста.

## ### 2 задание

Для реализации 2-го метода шифрования - метода Виженера, напишем алгоритм на языке Julia и проверим его работоспособность (Листинг-3, Листинг-4):

println("Vizgener cipher's work demonstration\n--------------------------------------")  
println("[!!!]Please, follow the instructions!")  
  
alphabet::String = "абвгдежзийклмнопрстуфхцчшщьыэюя"  
alphabet\_chars = collect(alphabet)  
function create\_shifted\_matrix\_detailed()  
 n = length(alphabet\_chars)  
 matrix = Array{Char}(undef, n, n)  
  
 println("Shifted symbols matrix:")  
 for i in 1:n  
 shift = i - 1  
 #print("Строка $i (сдвиг $shift):\t")  
  
 for j in 1:n  
 original\_index = j  
 shifted\_index = ((j - 1) + shift) % n + 1  
 matrix[i, j] = alphabet\_chars[shifted\_index]  
  
 #print("$(alphabet\_chars[shifted\_index]) ")  
 end  
 #println()  
 end  
  
 return matrix  
end  
  
print("\nEnter password(without repeating symbols): ")  
password::String = "математика"  
println("Entered password: $(password)")  
password\_chars = collect(password)  
  
print("\nEnter open message: ")  
openMessage::String = "криптографиясерьезнаянаука"  
println("Entered open message: $(openMessage)")  
openMessage\_chars = collect(openMessage)  
  
shifted\_matrix = create\_shifted\_matrix\_detailed()  
  
strings = 2  
columns = length(openMessage\_chars)  
password\_message\_char\_matrix = Array{Char}(undef, strings, columns)  
function encrypt()  
 password\_message\_char\_matrix = Array{Char}(undef, strings, columns)  
  
 for i in 1:strings  
 for j in 1:columns  
 if i == 1  
 index = ((j - 1) % length(password\_chars)) + 1  
 password\_message\_char\_matrix[i, j] = password\_chars[index]  
 #print("$(password\_message\_char\_matrix[i, j]) ")  
 elseif i == 2  
 password\_message\_char\_matrix[i, j] = openMessage\_chars[j]  
 #print("$(password\_message\_char\_matrix[i, j]) ")  
 end  
 end  
 #println()  
 end  
  
 encrypted\_message::String = ""  
 for index in 1:length(openMessage\_chars)  
 horizontal\_alphabet\_symbol = password\_message\_char\_matrix[1, index]  
 encrypting\_string\_symbol\_index = 1  
 for i in 1:length(alphabet\_chars)  
 if shifted\_matrix[1, i] == horizontal\_alphabet\_symbol  
 encrypting\_string\_symbol\_index = i  
 end  
 end  
  
 vertical\_alphabet\_symbol = password\_message\_char\_matrix[2, index]  
 encrypting\_column\_symbol\_index = 1  
 for i in 1:length(alphabet\_chars)  
 if shifted\_matrix[1, i] == vertical\_alphabet\_symbol  
 encrypting\_column\_symbol\_index = i  
 end  
 end  
 encrypted\_message \*= shifted\_matrix[encrypting\_string\_symbol\_index, encrypting\_column\_symbol\_index]  
 #print("$(shifted\_matrix[encrypting\_string\_symbol\_index, encrypting\_column\_symbol\_index]) ")  
 end  
 encrypted\_message\_chars = collect(encrypted\_message)  
  
 return encrypted\_message\_chars  
end  
  
function decrypt(encryptedMessage)  
 decrypted\_message::String = ""  
 for index in 1:length(encryptedMessage)  
   
 decrypted\_symbol\_vertical\_index = 1  
 for i in 1:length(alphabet\_chars)  
 if shifted\_matrix[1, i] == password\_chars[(index - 1) % length(password\_chars) + 1]  
 decrypted\_symbol\_vertical\_index = i  
 break  
 end  
 end  
  
 decrypted\_symbol\_horizontal\_index = 1  
 for i in 1:length(alphabet\_chars)  
 if shifted\_matrix[decrypted\_symbol\_vertical\_index, i] == encryptedMessage[index]  
 decrypted\_symbol\_horizontal\_index = i  
 break  
 end  
 end  
  
 #print("$(shifted\_matrix[1, decrypted\_symbol\_horizontal\_index]) ")  
 decrypted\_message \*= shifted\_matrix[1, decrypted\_symbol\_horizontal\_index]  
 end  
 decrypted\_message\_chars = collect(decrypted\_message)  
  
  
 return decrypted\_message\_chars  
end  
  
encryptedMessage = encrypt()  
println("\nEncrypted message: $(encryptedMessage)")  
  
decrypted\_message = decrypt(encryptedMessage)  
println("\nDecrypted message: $(decryptedMessage)")

*Листинг-3(фрагмент алгоритма, реализующего метод шифрования Виженера)*

Vizgener cipher's work demonstration  
--------------------------------------  
[!!!]Please, follow the instructions!  
  
Enter password(without repeating symbols): Entered password: математика  
  
Enter open message: Entered open message: криптографиясерьезнаянаука  
Shifted symbols matrix:  
  
Encrypted message: ['ц', 'р', 'ь', 'ф', 'я', 'о', 'х', 'ш', 'к', 'ф', 'ф', 'я', 'д', 'к', 'э', 'ь', 'ч', 'п', 'ч', 'а', 'л', 'н', 'т', 'ш', 'ц', 'а']  
  
Decrypted message: ['к', 'р', 'и', 'п', 'т', 'о', 'г', 'р', 'а', 'ф', 'и', 'я', 'с', 'е', 'р', 'ь', 'е', 'з', 'н', 'а', 'я', 'н', 'а', 'у', 'к', 'а']

*Листинг-4(результаты работы алгоритма шифрования Виженера)*

Исходя из полученных данных (Листинг-4), можно утверждать о том, что данный алгоритм успешно провёл шифрование вводимого текста, а позднее и дешифрование произведённого шифртекста.

## Заключение

В ходе проделанной лабораторной работы мной были усвоены знания по принципам работы с шифрами перестановки, а также получены навыки по их реализации.