# РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

### Факультет физико-математических и естественных наук

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1 =============== ## Тема: Шифры простой замены 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 простой замены.

### Задачи

1. Реализовать алгоритм шифрования Цезаря с произвольным ключом k
2. Реализовать алгоритм шифрования Атбаш

## Теория

В основе функционирования шифров простой замены лежит следующий принцип: для получения шифртекста отдельные символы или группы символов исходного алфавита заменяются символами или группами символов шифроалфавита.

Шифр Цезаря (также он является шифром простой” замены) — это моноалфавитная подстановка, т.е. каждой букве открытого текста ставится в соответствие одна буква шифртекста. На практике при создании шифра простой замены в качестве шифроалфавита берется исходный алфавит, но с нарушенным порядком букв (алфавитная перестановка). Для запоминания нового порядка букв перемешивание алфавита осуществляется с помощью пароля. В качестве пароля могут выступать слово или несколько слов с неповторяющимися буквами. Шифровальная таблица состоит “из двух строк: в первой записывается стандартный алфавит открытого текста, во второй — начиная с некоторой позиции размещается пароль (пробелы опускаются), а далее идут в алфавитном порядке оставшиеся буквы, не вошедшие в пароль. В случае несовпадения начала пароля с началом строки процесс после ее завершения циклически продолжается с первой позиции. Ключом шифра служит пароль вместе с числом, указывающим

Шифр Атбаш является частным случаем шифра Цезаря. Дело в том, что в данный шифр выполняет сдвиг на всю длину алфавита, и ключ можно приравнять к значению данного сдвига. ## Ход работы ### 1 задание — Для решения поставленной задачи - реализации алгоритма шифрования Цезаря воспользуемся языком Julia, чтобы написать скрипт, который будет выполнять шифрование вводимого сообщения по заданным параметрам алгоритма (Листинг-1):

println("Cesar cipher's work demonstration\n--------------------------------------")  
println("[!!!]Please, follow the instructions!")  
  
alphabet::String = "абвгдежзийклмнопрстуфхцчшщъыьэюя "  
alphabet\_chars = collect(alphabet)  
println("\nUsed alphabet(use symbols ONLY from the defined alphabet): ")  
for (char\_index, symbol) in enumerate(alphabet\_chars)  
 symbolCode = convert(Int32, symbol)  
 print("$(symbol)[$(symbolCode)]{$(char\_index-1)} ")  
 if (char\_index % 10) == 0  
 println()  
 end  
end  
  
print("\nEnter key: ")  
key::Int32 = 5  
println("Entered key: $(key)")  
  
print("\nEnter password(without repeating symbols): ")  
password::String = "пароль"  
println("Entered password: $(password)")  
  
print("\nEnter open message: ")  
openMessage::String = "этаглавауспешнозавершена"  
println("Entered open message: $(openMessage)")  
  
function findMessageSymbolPosition(symbol::Char, chars::Vector{Char})  
 for char\_index in 1:length(chars)  
 #print("$(chars[char\_index])?=$(symbol)\n")  
 if chars[char\_index] == symbol  
 return char\_index  
 end  
 end  
  
 return -1  
end  
  
function encrypt()  
 println("----ENCRYPTING----")  
  
 encryptedMessage::String = ""  
 for symbol in openMessage  
 pos = findMessageSymbolPosition(symbol, alphabet\_chars)  
 print("$(symbol) - position: $(pos - 1)")  
  
 encrypted\_pos = (pos - 1 + key) % length(alphabet\_chars)  
 encrypted\_symbol = alphabet\_chars[encrypted\_pos+1]  
 encryptedMessage \*= encrypted\_symbol  
 println("------> $(encrypted\_symbol) - position: $(findMessageSymbolPosition(encrypted\_symbol, alphabet\_chars) - 1)")  
 end  
 return encryptedMessage  
end  
encryptedMessage::String = encrypt()  
println("\nEncrypted message: $(encryptedMessage)")  
  
function decrypt()  
 println("----DECRYPTING----")  
  
 decryptedMessage::String = ""  
 for symbol in encryptedMessage  
 pos = findMessageSymbolPosition(symbol, alphabet\_chars)  
 print("$(symbol) - position: $(pos - 1)")  
  
 # Шифрование  
 decrypted\_pos = (pos - 1 - key) % length(alphabet\_chars)  
 if decrypted\_pos < 0  
 decrypted\_pos = length(alphabet\_chars) + decrypted\_pos  
 end  
 decrypted\_symbol = alphabet\_chars[decrypted\_pos+1]  
 decryptedMessage \*= decrypted\_symbol  
 println("------> $(decrypted\_symbol) - position: $(findMessageSymbolPosition(decrypted\_symbol, alphabet\_chars) - 1)")  
 end  
 return decryptedMessage  
end  
  
decryptedMessage::String = decrypt()  
println("\nDecrypted message: $(decryptedMessage)")

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

Cesar cipher's work demonstration  
--------------------------------------  
[!!!]Please, follow the instructions!  
  
Used alphabet(use symbols ONLY from the defined alphabet):   
а[1072]{0} б[1073]{1} в[1074]{2} г[1075]{3} д[1076]{4} е[1077]{5} ж[1078]{6} з[1079]{7} и[1080]{8} й[1081]{9}   
к[1082]{10} л[1083]{11} м[1084]{12} н[1085]{13} о[1086]{14} п[1087]{15} р[1088]{16} с[1089]{17} т[1090]{18} у[1091]{19}   
ф[1092]{20} х[1093]{21} ц[1094]{22} ч[1095]{23} ш[1096]{24} щ[1097]{25} ъ[1098]{26} ы[1099]{27} ь[1100]{28} э[1101]{29}   
ю[1102]{30} я[1103]{31} [32]{32}   
Enter key: Entered key: 5  
  
Enter password(without repeating symbols): Entered password: пароль  
  
Enter open message: Entered open message: этаглавауспешнозавершена  
----ENCRYPTING----  
э - position: 29------> б - position: 1  
т - position: 18------> ч - position: 23  
а - position: 0------> е - position: 5  
г - position: 3------> и - position: 8  
л - position: 11------> р - position: 16  
а - position: 0------> е - position: 5  
в - position: 2------> з - position: 7  
а - position: 0------> е - position: 5  
у - position: 19------> ш - position: 24  
с - position: 17------> ц - position: 22  
п - position: 15------> ф - position: 20  
е - position: 5------> к - position: 10  
ш - position: 24------> э - position: 29  
н - position: 13------> т - position: 18  
о - position: 14------> у - position: 19  
з - position: 7------> м - position: 12  
а - position: 0------> е - position: 5  
в - position: 2------> з - position: 7  
е - position: 5------> к - position: 10  
р - position: 16------> х - position: 21  
ш - position: 24------> э - position: 29  
е - position: 5------> к - position: 10  
н - position: 13------> т - position: 18  
а - position: 0------> е - position: 5  
  
Encrypted message: бчеирезешцфкэтумезкхэкте  
----DECRYPTING----  
б - position: 1------> э - position: 29  
ч - position: 23------> т - position: 18  
е - position: 5------> а - position: 0  
и - position: 8------> г - position: 3  
р - position: 16------> л - position: 11  
е - position: 5------> а - position: 0  
з - position: 7------> в - position: 2  
е - position: 5------> а - position: 0  
ш - position: 24------> у - position: 19  
ц - position: 22------> с - position: 17  
ф - position: 20------> п - position: 15  
к - position: 10------> е - position: 5  
э - position: 29------> ш - position: 24  
т - position: 18------> н - position: 13  
у - position: 19------> о - position: 14  
м - position: 12------> з - position: 7  
е - position: 5------> а - position: 0  
з - position: 7------> в - position: 2  
к - position: 10------> е - position: 5  
х - position: 21------> р - position: 16  
э - position: 29------> ш - position: 24  
к - position: 10------> е - position: 5  
т - position: 18------> н - position: 13  
е - position: 5------> а - position: 0  
  
Decrypted message: этаглавауспешнозавершена

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

Исходя из полученных результатов (Листинг-2), можно судить о том, что данный алгоритм производит успешные шифрование и дешифрование вводимого текста и зашифрованного текста соответственно, выполняя сдвиг согласно заданному значению ключа.

## ### 2 задание

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

println("Atbash cipher's work demonstration\n--------------------------------------")  
println("[!!!]Please, follow the instructions!")  
  
alphabet::String = "абвгдежзийклмнопрстуфхцчшщъыьэюя "  
alphabet\_chars = collect(alphabet)  
println("\nUsed alphabet(use symbols ONLY from the defined alphabet): ")  
for (char\_index, symbol) in enumerate(alphabet\_chars)  
 symbolCode = convert(Int32, symbol)  
 print("$(symbol)[$(symbolCode)]{$(char\_index-1)} ")  
 if (char\_index % 10) == 0  
 println()  
 end  
end  
  
  
print("\nEnter key: ")  
key::Int32 = 32#parse(Int32, readline())  
println("Entered key: $(key)")  
  
  
print("\nEnter open message: ")  
openMessage::String = "этаглавауспешнозавершена"#convert(String, readline())  
println("Entered open message: $(openMessage)")  
  
  
function findMessageSymbolPosition(symbol::Char, alphabet\_chars::Vector{Char})  
 for char\_index in 1:length(alphabet\_chars)  
 if alphabet\_chars[char\_index] == symbol  
 return char\_index  
 end  
 end  
end  
  
  
function encrypt()  
 println("----ENCRYPTING----")  
  
 encryptedMessage::String = ""  
 for symbol in openMessage  
 pos = findMessageSymbolPosition(symbol, alphabet\_chars)  
 print("$(symbol) - position: $(pos - 1)")  
  
 encrypted\_pos = (pos - 1 + key) % length(alphabet\_chars)  
 encrypted\_symbol = alphabet\_chars[encrypted\_pos + 1]  
 encryptedMessage \*= encrypted\_symbol  
 println("------> $(encrypted\_symbol) - position: $(findMessageSymbolPosition(encrypted\_symbol, alphabet\_chars) - 1)")  
 end  
 return encryptedMessage  
end  
  
  
encryptedMessage::String = encrypt()  
println("\nEncrypted message: $(encryptedMessage)")  
  
function decrypt()  
 println("----DECRYPTING----")  
  
 decryptedMessage::String = ""  
 for symbol in encryptedMessage  
 pos = findMessageSymbolPosition(symbol, alphabet\_chars)  
 print("$(symbol) - position: $(pos - 1)")  
  
 decrypted\_pos = (pos - 1 - key) % length(alphabet\_chars)  
 if decrypted\_pos < 0   
 decrypted\_pos = length(alphabet\_chars) + decrypted\_pos  
 end  
 decrypted\_symbol = alphabet\_chars[decrypted\_pos + 1]  
 decryptedMessage \*= decrypted\_symbol  
 println("------> $(decrypted\_symbol) - position: $(findMessageSymbolPosition(decrypted\_symbol, alphabet\_chars) - 1)")  
 end  
 return decryptedMessage  
end  
  
  
decryptedMessage::String = decrypt()  
println("\nDecrypted message: $(decryptedMessage)")

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

Atbash cipher's work demonstration  
--------------------------------------  
[!!!]Please, follow the instructions!  
  
Used alphabet(use symbols ONLY from the defined alphabet):   
а[1072]{0} б[1073]{1} в[1074]{2} г[1075]{3} д[1076]{4} е[1077]{5} ж[1078]{6} з[1079]{7} и[1080]{8} й[1081]{9}   
к[1082]{10} л[1083]{11} м[1084]{12} н[1085]{13} о[1086]{14} п[1087]{15} р[1088]{16} с[1089]{17} т[1090]{18} у[1091]{19}   
ф[1092]{20} х[1093]{21} ц[1094]{22} ч[1095]{23} ш[1096]{24} щ[1097]{25} ъ[1098]{26} ы[1099]{27} ь[1100]{28} э[1101]{29}   
ю[1102]{30} я[1103]{31} [32]{32}   
Enter key: Entered key: 32  
  
Enter open message: Entered open message: этаглавауспешнозавершена  
----ENCRYPTING----  
э - position: 29------> ь - position: 28  
т - position: 18------> с - position: 17  
а - position: 0------> - position: 32  
г - position: 3------> в - position: 2  
л - position: 11------> к - position: 10  
а - position: 0------> - position: 32  
в - position: 2------> б - position: 1  
а - position: 0------> - position: 32  
у - position: 19------> т - position: 18  
с - position: 17------> р - position: 16  
п - position: 15------> о - position: 14  
е - position: 5------> д - position: 4  
ш - position: 24------> ч - position: 23  
н - position: 13------> м - position: 12  
о - position: 14------> н - position: 13  
з - position: 7------> ж - position: 6  
а - position: 0------> - position: 32  
в - position: 2------> б - position: 1  
е - position: 5------> д - position: 4  
р - position: 16------> п - position: 15  
ш - position: 24------> ч - position: 23  
е - position: 5------> д - position: 4  
н - position: 13------> м - position: 12  
а - position: 0------> - position: 32  
  
Encrypted message: ьс вк б тродчмнж бдпчдм   
----DECRYPTING----  
ь - position: 28------> э - position: 29  
с - position: 17------> т - position: 18  
 - position: 32------> а - position: 0  
в - position: 2------> г - position: 3  
к - position: 10------> л - position: 11  
 - position: 32------> а - position: 0  
б - position: 1------> в - position: 2  
 - position: 32------> а - position: 0  
т - position: 18------> у - position: 19  
р - position: 16------> с - position: 17  
о - position: 14------> п - position: 15  
д - position: 4------> е - position: 5  
ч - position: 23------> ш - position: 24  
м - position: 12------> н - position: 13  
н - position: 13------> о - position: 14  
ж - position: 6------> з - position: 7  
 - position: 32------> а - position: 0  
б - position: 1------> в - position: 2  
д - position: 4------> е - position: 5  
п - position: 15------> р - position: 16  
ч - position: 23------> ш - position: 24  
д - position: 4------> е - position: 5  
м - position: 12------> н - position: 13  
 - position: 32------> а - position: 0  
  
Decrypted message: этаглавауспешнозавершена

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

## Заключение

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