Министерство науки и высшего образования Российской Федерации

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

Кафедра информационных систем и программной инженерии

**Лабораторная работа №4**

**по дисциплине**

**«АДМИНИСТРИРОВАНИЕ И БЕЗОПАСНОСТЬ ИНФОРМАЦИОННЫХ СИСТЕМ»**

**АЛГОРИТМ ВИЖЕНЕРА**

**Выполнил**:

ст. гр. ПРИ-120

Д. А. Грачев

**Принял**:

Доцент кафедры ИСПИ

Курочкин С. В.

Владимир, 2023

ЦЕЛЬ РАБОТЫ

Реализация криптографического алгоритма шифрования с использованием открытого текста в качестве ключа по таблице Виженера.

ВЫПОЛНЕНИЕ РАБОТЫ

1. Листинг реализации функции для формирования смещенного алфавита

function alphabetsFromChar(char) {  
 const charIndex = alphabet.indexOf(char);  
 return alphabet.slice(charIndex) + alphabet.slice(0, charIndex)  
}

1. Листинг реализации функции шифрования

function vizhenerEnc(text, key) {  
 const encription = {  
 keyLength: key.length,  
 alphabets: key.toLowerCase().split("").map((item) => alphabetsFromChar(item)),  
 nIndex: [],  
 spaceIndex: []  
 }  
  
 text = text  
 .toLowerCase()  
 .split("")  
 .filter((item) => forDel.includes(item))  
 .join("")  
  
 text.split("").forEach((item, idx) => {  
 if (item === "\n") encription.nIndex.push(idx - encription.nIndex.length - encription.spaceIndex.length)  
 else if (item === " ") encription.spaceIndex.push(idx - encription.nIndex.length - encription.spaceIndex.length)  
 })  
  
 return text  
 .replaceAll("\n", " ")  
 .replaceAll(" ", "")  
 .split("")  
 .map((char, idx) => Number.isInteger((idx + 1) / encription.keyLength) ? char + " " : char)  
 .join("")  
 .split(" ")  
 .map((item) => item.split("").map((item, idx) => encription.alphabets[idx][alphabet.indexOf(item)]).join(""))  
 .join("")  
 .split("")  
 .map((char, idx) => {  
 if (encription.nIndex.includes(idx+1)) return char += "\n"  
 else if (encription.spaceIndex.includes(idx+1)) return char += " "  
 else return char  
 })  
 .join("")  
}

1. Листинг реализации функции дешифрования

function vizhenerDec(text, key) {  
 const decription = {  
 keyLength: key.length,  
 alphabets: key.toLowerCase().split("").map((item) => alphabetsFromChar(item)),  
 nIndex: [],  
 spaceIndex: []  
 }  
  
 text = text  
 .toLowerCase()  
 .split("")  
 .filter((item) => forDel.includes(item))  
 .join("")  
  
 text.split("").forEach((item, idx) => {  
 if (item === "\n") decription.nIndex.push(idx - decription.nIndex.length - decription.spaceIndex.length)  
 else if (item === " ") decription.spaceIndex.push(idx - decription.nIndex.length - decription.spaceIndex.length)  
 })  
  
  
 return text  
 .replaceAll("\n", " ")  
 .replaceAll(" ", "")  
 .split("")  
 .map((char, idx) => Number.isInteger((idx + 1) / decription.keyLength) ? char + " " : char)  
 .join("")  
 .split(" ")  
 .map((item) => item.split("").map((item, idx) => alphabet[decription.alphabets[idx].indexOf(item)]).join(""))  
 .join("")  
 .split("")  
 .map((char, idx) => {  
 if (decription.nIndex.includes(idx+1)) return char += "\n"  
 else if (decription.spaceIndex.includes(idx+1)) return char += " "  
 else return char  
 })  
 .join("")  
}

1. Задание 1: Зашифруйте с помощью алгоритма Виженера следующую последовательность. Для шифрования используйте ключ «идеал»

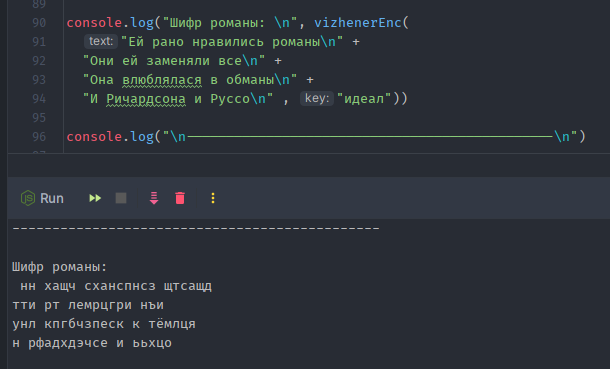


Рисунок . Задание 1

1. Задание 2: Расшифруйте помощью алгоритма Виженера следующую последовательность. Для дешифрования используйте ключ «проза».

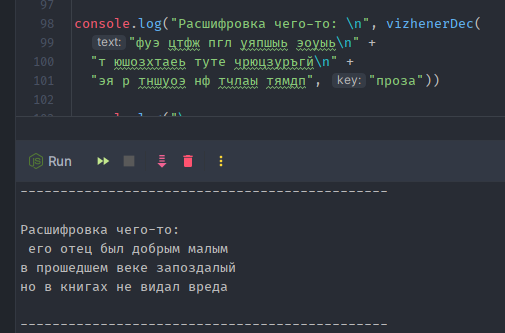


Рисунок . Задание 2

1. Напишите алгоритм и программу для шифрования/дешифрования строк алгоритмом Виежера. Для шифрования используйте ключ «даниил»
   1. Вариант 1

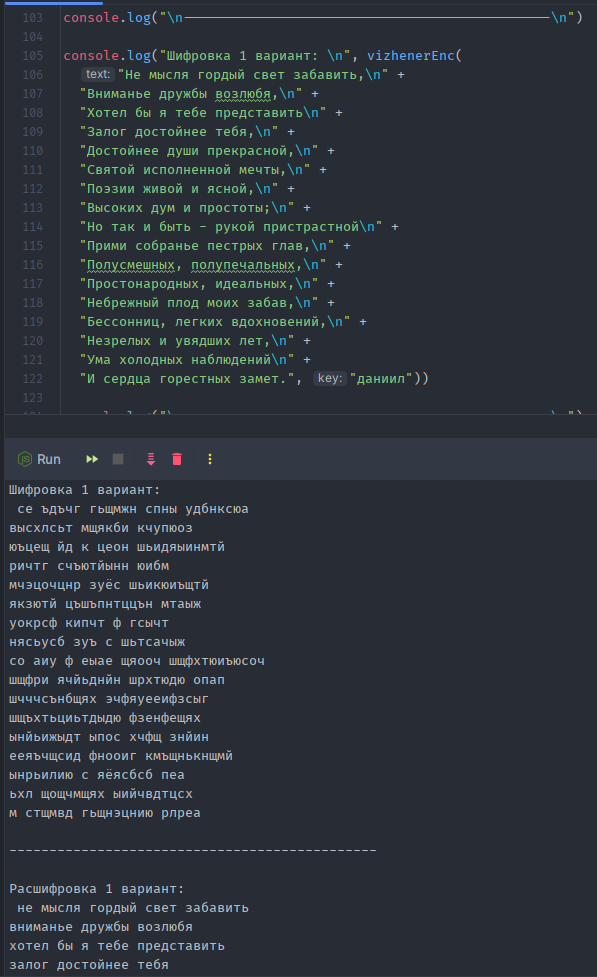


Рисунок . Задание 3 - вариант 1

* 1. Вариант 2

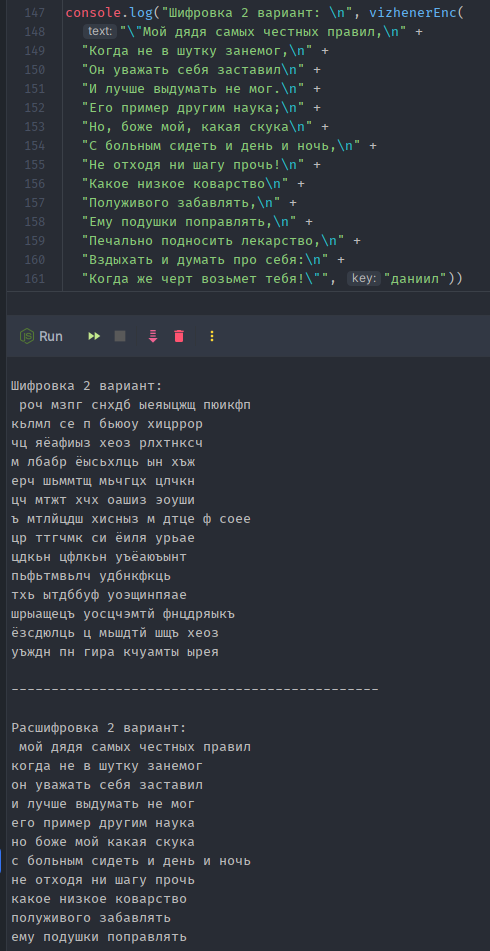


Рисунок . Задание 3 - вариант 2

ВЫВОД

В ходе выполнения работы был реализован криптографический алгоритма шифрования с использованием открытого текста в качестве ключа по таблице Виженера.