**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное автономное образовательное учреждение высшего образования

**«Национальный исследовательский Томский политехнический университет»**

Инженерная школа информационных технологий и робототехники

Отделение информационных технологий

Направление «Информационные системы и технологии»

Отчёт по лабораторной работе №3

по дисциплине «**Информационная безопасность и защита информации**»

**Шифр, основанный на методе умножения матриц**

Выполнил:

Студент группы 8И7Б \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Балашов И.А.

Проверил:

Ассистент ОИТ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Григорьев Д. С.

Томск 2021

# Задание

1. Заполнить таблицу 2 в массив, в котором должны храниться все буквы русского алфавита от а до я и от А до Я плюс символы пробел, точка, двоеточие, восклицательный знак, вопросительный знак и запятая (всего 72 символа).
2. Зашифровать любое сообщение, введенное с клавиатуры, методом произведения матриц.
3. Определить какой должна быть матрица, чтобы зашифрованную фразу можно было расшифровать.
4. Расшифровать сообщение.

**Ход работы**

Программа реализована на языке python, при написании программы была использована библиотека numpy. В программе имеется две опции, расшифровать строку или зашифровать строку.

С алгоритмом работы программы можно ознакомиться на рисунке 1.

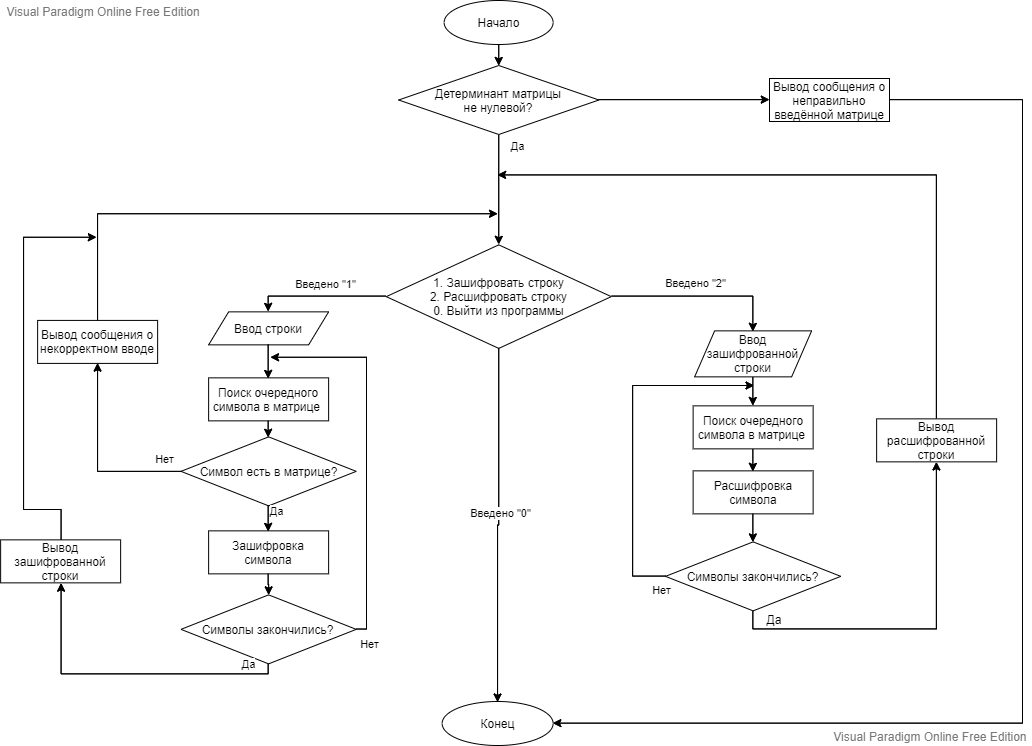


Рисунок 1 – Схема работы алгоритма программы

# Листинг программы

**import** numpy **as** np  
  
str = (**'АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЧШЦЩЪЫЬЭЮЯабвгдеёжзийклмнопрстуфхчшцщъыьэюя .:!?,'**)  
  
n = 3  
  
key = np.array([[1, 3, 2], [1, 3, 5], [3, 2, 1]])  
**while** (np.linalg.det(key)==0):  
 print(**'детерминант матрицы ключа равен 0, измените матрицу ключа'**)  
 input()  
  
  
print(**'Зашифровать строку введите 1'**)  
print(**'Расшифровать строку введите 2'**)  
print(**'Закончить 0'**)  
  
run = input()  
**while** (run != **'0'**):  
 **if** (run == **'1'**):  
 print(**'Введите строку'**)  
 inp = input()  
 **if**(len(inp) % 3 == 2):  
 inp +=**' '  
 if** (len(inp) % 3 == 1):  
 inp += **' '** codeALf = []  
 code = []  
  
 sucsessFlag = **True  
 for** m **in** range(len(inp)):  
 flag = **True  
 for** l **in** str:  
 **if** l == inp[m]:  
 codeALf.append(str.index(l)+1) *# +1* flag = **False  
 if** flag:  
 sucsessFlag = **False** print(**'В введёной строке присутствует недопустимый символ'**)  
 **break  
  
 if** sucsessFlag:  
 codeALfnp = np.reshape(np.array(codeALf), (len(inp) // 3, 3))  
  
 **for** i **in** range(len(inp) // 3):  
 code.append(key.dot(codeALfnp[i].transpose()))  
 print(**'Зашифрованная строка:'**)  
 print(np.reshape(np.array(code), (1,len(inp))))  
  
 **if** run == **'2'**:  
 print(**'Введите зашифрованную строку'**)  
 code = input()  
 codeSPL = code.split(**' '**)  
 codeNP = np.reshape(np.array(codeSPL), (len(codeSPL)//3, 3))  
 print(codeNP)  
 inpCod = []  
 keyOb = np.linalg.inv(key)  
 **for** i **in** range(len(codeSPL) // 3):  
 inpCod.append(keyOb.dot(codeNP[i].transpose().astype(float)))  
 inpCodVek = (np.reshape(inpCod, (1, len(codeSPL)))+0.5).astype(int)  
 print(**'Исходный текст:'**)  
 **for** i **in** inpCodVek[0]:  
 print(str[i-1], end = **""**)  
 print(2)  
  
  
 print(**'Зашифровать строку введите 1'**)  
 print(**'Расшифровать строку введите 2'**)  
 print(**'Закончить 0'**)  
 run = input()

# Результаты работы программы

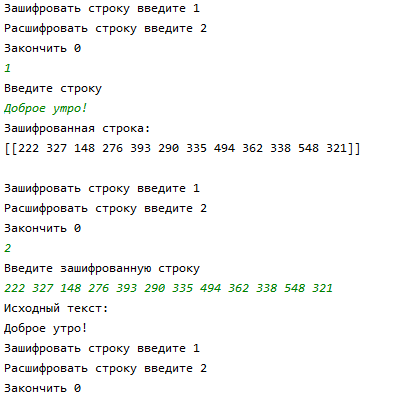


Рисунок 2 – Результат работы программы

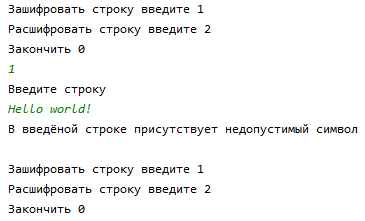


Рисунок 3 – Результат работы при введении символов, не присутствующих в алфавите



Рисунок 3 – Результат работы при использовании матрици с детерминантом равным 0

# Вывод

В ходе лабораторной работы была разработана программа использующая шифрование, основанное на методе умножения матриц, программа способна как зашифровывать, так и расшифровывать строки, также предусмотрена отказоустойчивость при введении неверных символов или при равенстве детерминанта 0. Все задачи выполнены, лабораторная сделана в полном объёме.