Министерство образования Республики Беларусь Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ

И РАДИОЭЛЕКТРОНИКИ

КАФЕДРА ИНФОРМАТИКИ

Отчёт по лабораторной работе №1

По теме “Информационные сети. Основы безопасности. Шифр Цезаря.

Шифр Виженера.”

Выполнила: студентка гр. 053501 Шурко Т.А.

Проверил: ассистент кафедры информатики Протько М. И.

Минск 2023

Содержание

[1. Введение 3](file:///D:\downloads\Lab1.docx#_Toc7030)

[2. Теоретические сведения 3](file:///D:\downloads\Lab1.docx#_Toc7031)

[3. Блок-схема алгоритма 6](file:///D:\downloads\Lab1.docx#_Toc7032)

[4. Тестирование программного продукта 8](file:///D:\downloads\Lab1.docx#_Toc7033)

[5. Вывод 10](file:///D:\downloads\Lab1.docx#_Toc7035)

Примечание. Код программ 11

# Введение

Целью данной лабораторной работы является изучить теоретические сведения о шифрах Цезаря и Виженера, реализовать программные средства шифрования и дешифрования текстовых файлов при помощи Шифра Цезаря, (шифра сдвига, кода Цезаря) и шифра Виженера.

# Теоретические сведения

**Шифр Цезаря**

Шифр Цезаря, также известный, как шифр сдвига, код Цезаря или сдвиг Цезаря – один из самых простых и наиболее широко известных методов шифрования.

Шифр Цезаря – это вид шифра подстановки, в котором каждый символ в открытом тексте заменяется символом находящимся на некотором постоянном числе позиций левее или правее него в алфавите. Например, в шифре со сдвигом 4 А была бы заменена на Г, Б станет Д, и так далее.

Шифр назван в честь римского императора Гая Юлия Цезаря, использовавшего его для секретной переписки со своими генералами.

Шаг шифрования, выполняемый шифром Цезаря, часто включается как часть более сложных схем, таких как шифр Виженера, и все ещё имеет современное приложение в системе ROT13. Как и все моноалфавитные шифры, шифр Цезаря легко взламывается и не имеет практически никакого применения на практике.



Рис. 2.1. Шифр Цезаря

***Математическая модель***

Если сопоставить каждому символу алфавита его порядковый номер (нумеруя с 0), то шифрование и дешифрование можно выразить формулами модульной арифметики:

y=(x+k)\ \mod\ n

x=(y-k+n)\ \mod\ n,

где ~x – символ открытого текста, ~y – символ шифрованного текста, ~n – мощность алфавита, а ~k – ключ.

С точки зрения математики шифр Цезаря является частным случаем аффинного шифра.

***Пример***

Шифрование с использованием ключа k = 3. Буква «Е» «сдвигается» на три буквы вперёд и становится буквой «З». Твёрдый знак, перемещённый на три буквы вперёд, становится буквой «Э», и так далее:

Исходный алфавит:

***АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ***

Шифрованный:

***ГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯАБ***

Оригинальный текст:

*Съешь же ещё этих мягких французских булок, да выпей чаю.*

Шифрованный текст получается путём замены каждой буквы оригинального текста соответствующей буквой шифрованного алфавита:

*Фэзыя йз зьи ахлш пвёнлш чугрщцкфнлш дцосн, жг еютзм ъгб.*

**Шифр Виженера**

Шифр Виженера состоит из последовательности нескольких шифров Цезаря с различными значениями сдвига. Для зашифровывания может использоваться таблица алфавитов, называемая tabula recta или квадрат (таблица) Виженера. Применительно к латинскому алфавиту таблица Виженера составляется из строк по 26 символов, причём каждая следующая строка сдвигается на несколько позиций. Таким образом, в таблице получается 26 различных шифров Цезаря. На каждом этапе шифрования используются различные алфавиты, выбираемые в зависимости от символа ключевого слова. Например, предположим, что исходный текст имеет такой вид:

ATTACKATDAWN

Человек, посылающий сообщение, записывает ключевое слово («LEMON») циклически до тех пор, пока его длина не будет соответствовать длине исходного текста:

LEMONLEMONLE

Первый символ исходного текста A зашифрован последовательностью L, которая является первым символом ключа. Первый символ L шифрованного текста находится на пересечении строки L и столбца A в таблице Виженера. Точно так же для второго символа исходного текста используется второй символ ключа; то есть второй символ шифрованного текста X получается на пересечении строки E и столбца T. Остальная часть исходного текста шифруется подобным способом.

Исходный текст: ATTACKATDAWN

Ключ: LEMONLEMONLE

Зашифрованный текст: LXFOPVEFRNHR

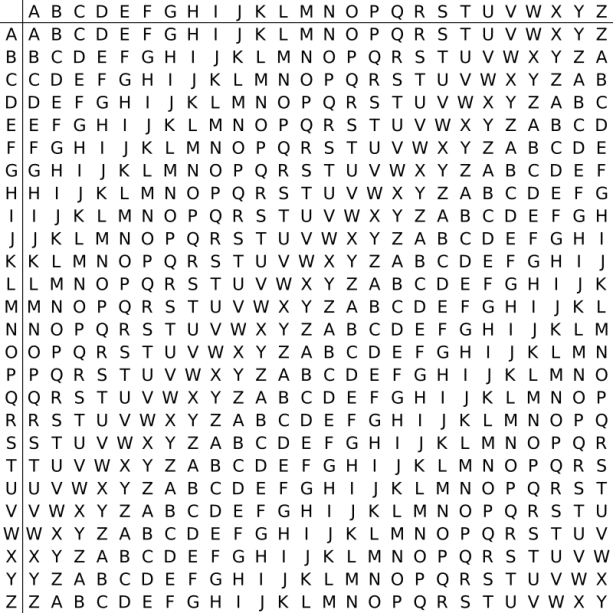


Рис. 2.2. Квадрат Виженера

Расшифровывание производится следующим образом: находим в таблице Виженера строку, соответствующую первому символу ключевого слова; в данной строке находим первый символ зашифрованного текста. Столбец, в котором находится данный символ, соответствует первому символу исходного текста. Следующие символы зашифрованного текста расшифровываются подобным образом.

Если n — количество букв в алфавите, m j — буквы открытого текста, k j — буквы ключа, то шифрование Виженера можно записать следующим образом:

c j = m j + k j ( mod n )

И расшифровывание:

m j = c j − k j ( mod n )

В компьютере такая операция соответствует сложению кодов ASCII символов сообщения и ключа по некоторому модулю. Кажется, что если таблица будет более сложной, чем циклическое смещение строк, то шифр станет надежнее. Это действительно так, если ее менять чаще, например, от слова к слову. Но составление таких таблиц, представляющих собой латинские квадраты, где любая буква встречается в строке или столбце один раз, трудоемко и его стоит делать лишь на ЭВМ. Для ручного же многоалфавитного шифра полагаются лишь на длину и сложность ключа, используя приведенную таблицу, которую можно не держать в тайне, а это упрощает шифрование и расшифровывание.

# Блок-схема алгоритма

Рис. 3.1. Блок-схема алгоритма шифрования Цезаря

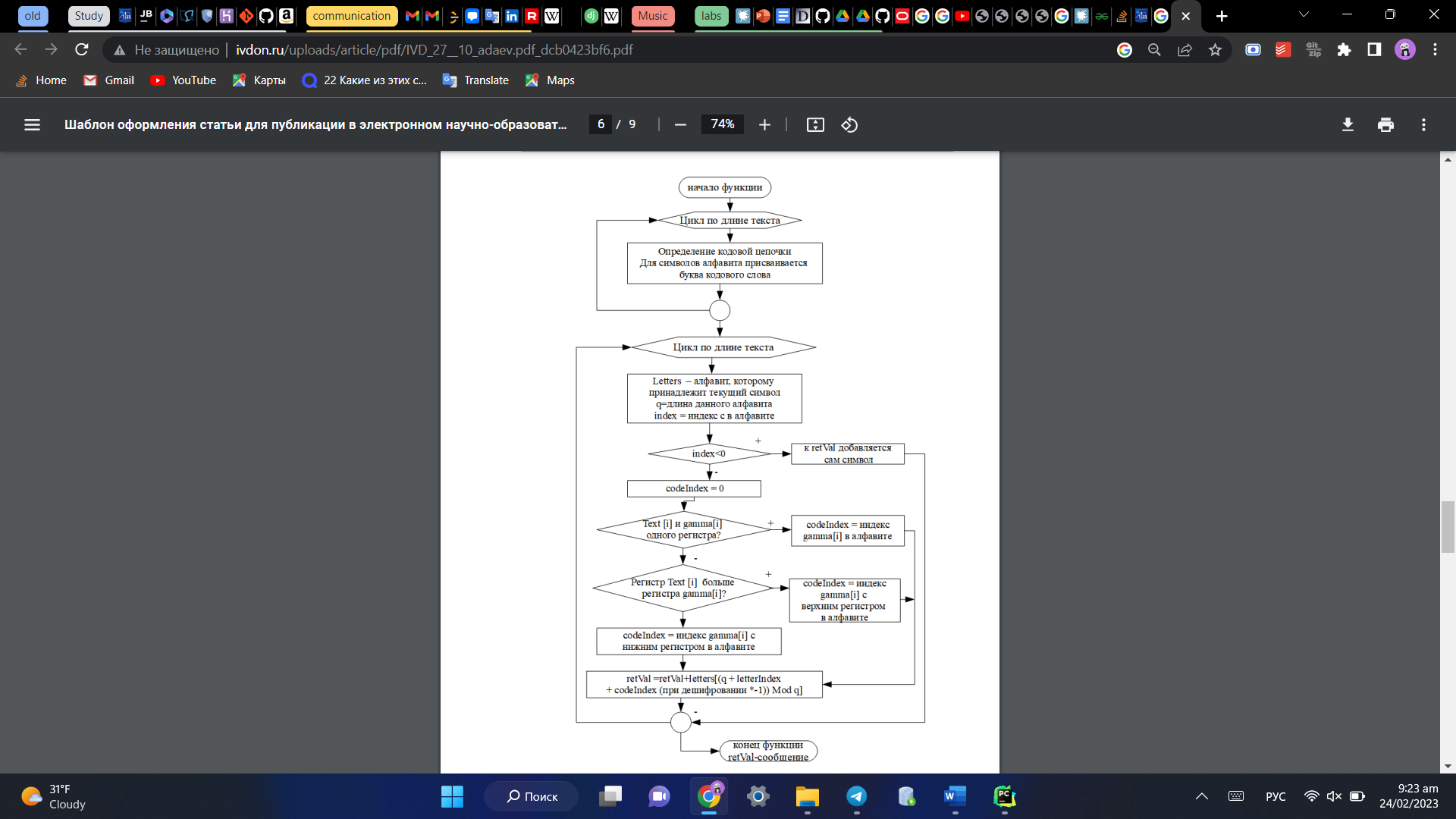


Рис. 3.2. Блок-схема алгоритма шифрования Виженера

# Тестирование программного продукта

# Рассмотрим примеры, которые присутствовали в теоретической части.

# Шифр Цезаря с шагом 3 (как в примере):

message = 'Съешь же ещё этих мягких французских булок, да выпей чаю.'  
res = сaesar\_encryption.encrypt(message, 3)  
  
print("Encrypted: ", res)  
print("Decrypted: ", сaesar\_encryption.decrypt(res, 3))

# Рис. 4.1. Запускаемый код

Рис. 4.2. Результат выполнения кода

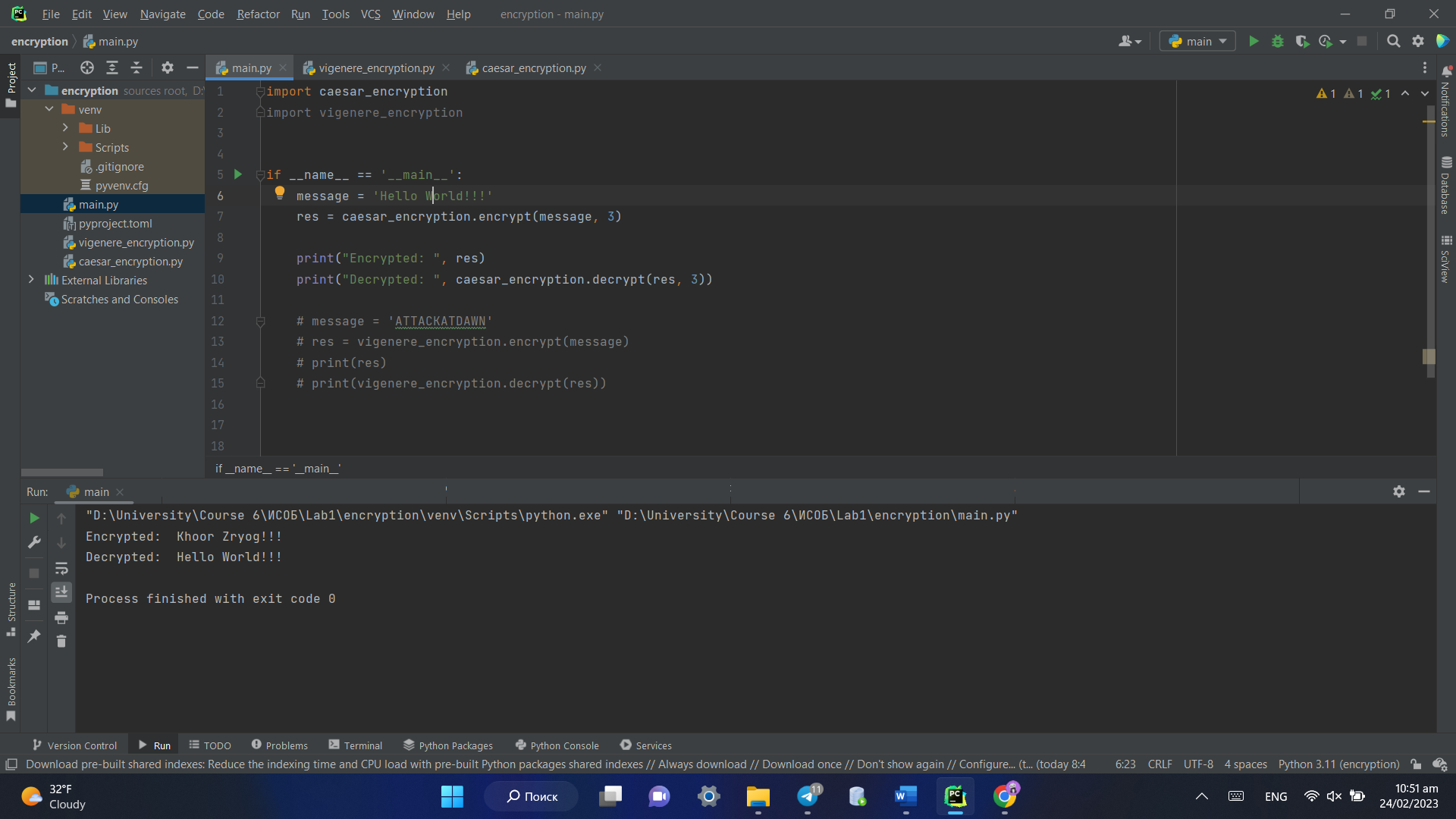
В результате проверки работоспособности (см. рис. 4.1 - 4.3), ожидаемый результат, указанный в теоретических сведениях, совпал с выводом программного продукта.

Рис. 4.3. Результат выполнения с английским алфавитом

Во всех запускаемых примерах шаг шифра = 3. Однако, при желании, его можно указать при вызове функции.

# Шифр Виженера с ключом = “LEMON” (как в примере):

message = 'ATTACKATDAWN'  
res = vigenere\_encryption.encrypt(message)  
print("Encrypted: ", res)  
print("Decrypted: ", vigenere\_encryption.decrypt(res))

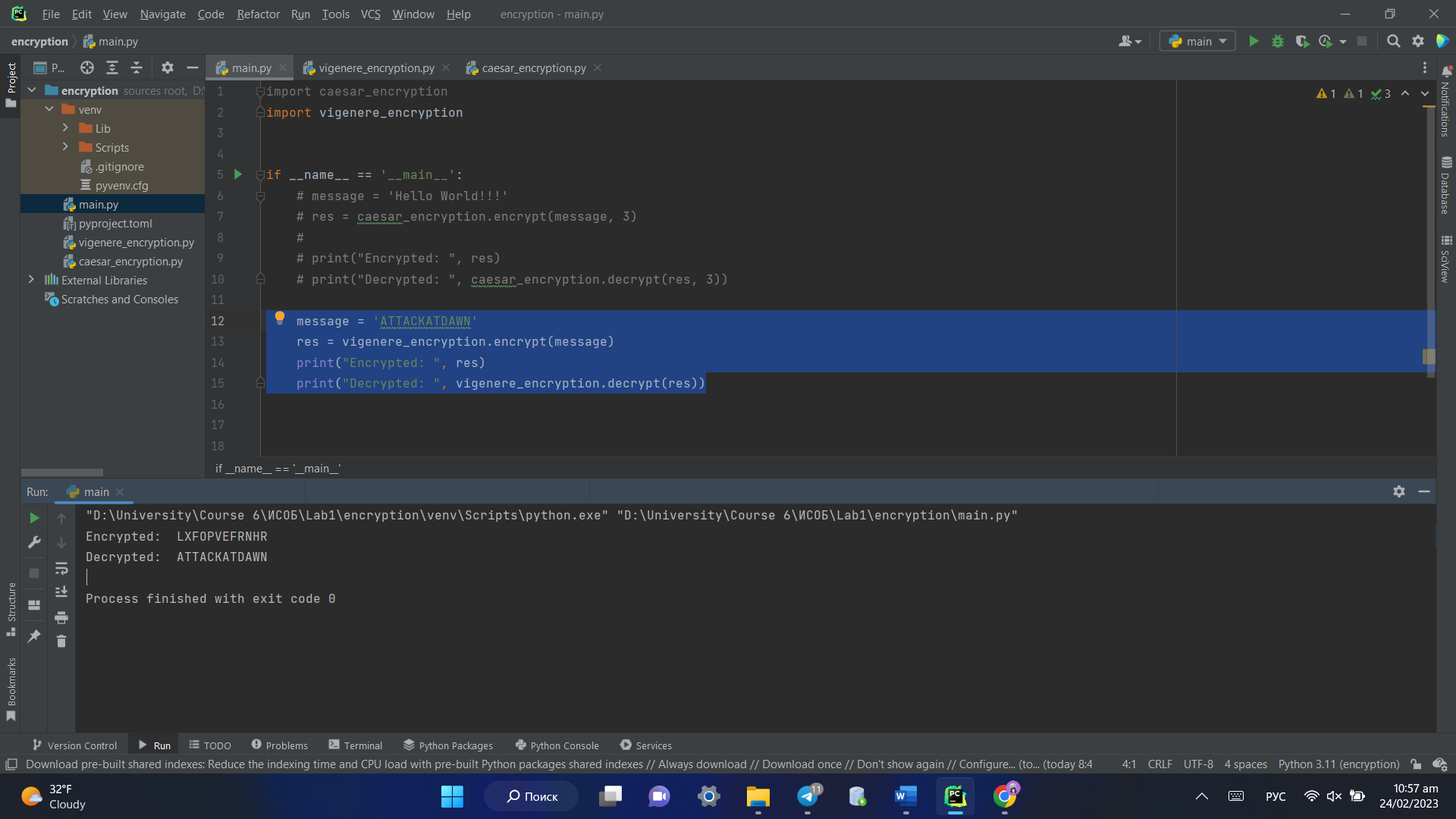
Рис. 4.4. Запускаемый код для проверки шифра Виженера

Рис. 4.5. Результат работы шифра Виженера

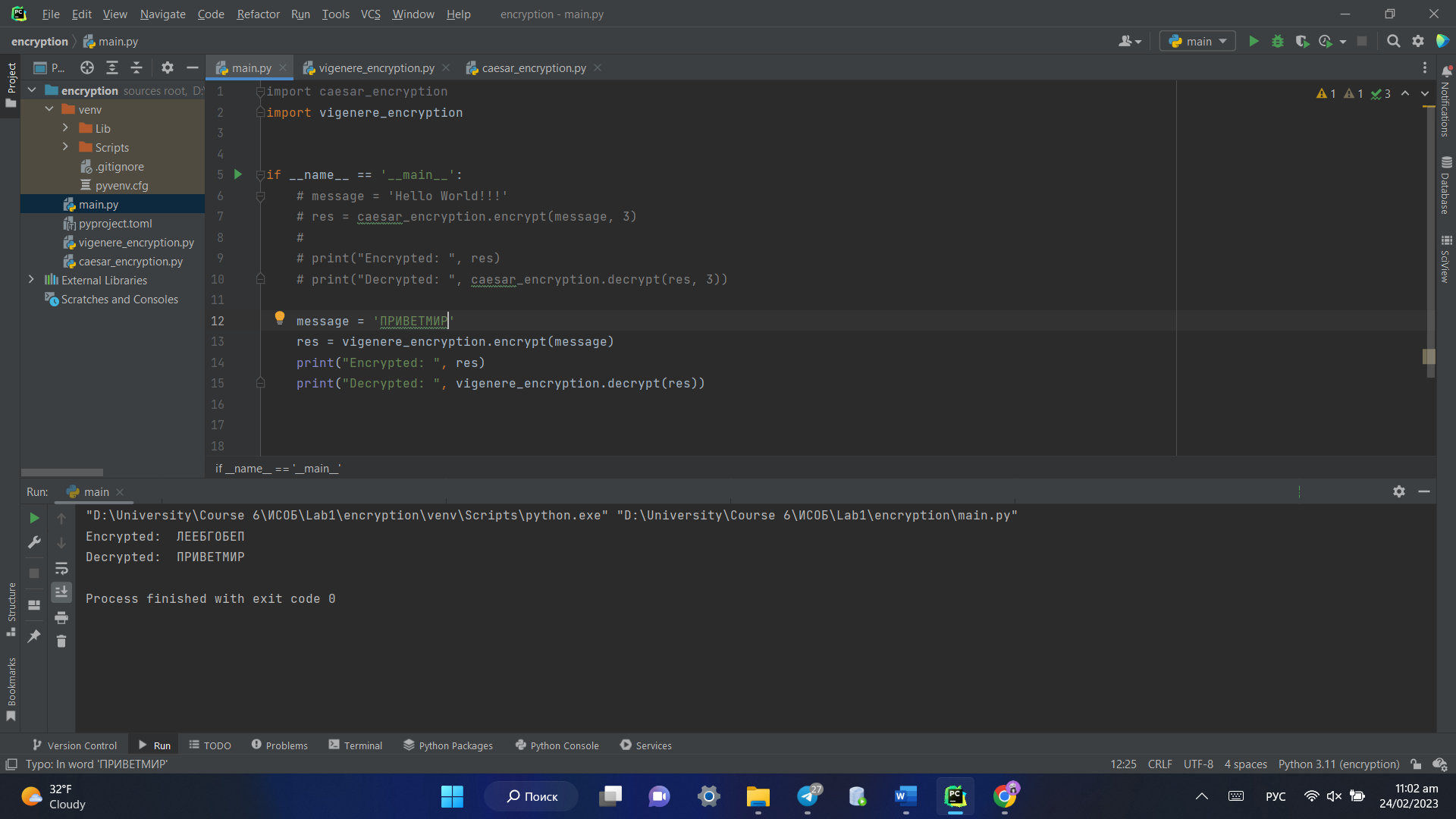
В результате запуска программы по шифрованию сообщения с помощью шифра Виженера (см. рис. 4.4 - 4.5), получили ожидаемый результат, который и был указан в примере. Данный шифр также работает и с русскими буками (см. рис. 4.6).

Рис. 4.6. Результат работы шифра Винежера с русским алфавитом

# Вывод

В результате выполнения лабораторной работы №1 были изучены теоретические сведения о шифрах Цезаря и Виженера, составлены блок-схемы этих алгоритмов, а также составлен программный продукт, выполняющий шифрование и дешифрование сообщений.

Приложение. Текст программ

сaesar\_encryption.py

alphabet = "ABCDEFGHIJKLMNOPQRSTYVWXYZ"  
# alphabet = "АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ"  
  
  
def func\_encr(i, step):  
 if alphabet.find(i) != -1:  
 return alphabet[(alphabet.find(i) + step) % len(alphabet)]  
 elif i.isalpha():  
 return alphabet[(alphabet.find(i.upper()) + step) % len(alphabet)].lower()  
 else:  
 return i  
  
  
def func\_decr(i, step):  
 if alphabet.find(i) != -1:  
 return alphabet[(alphabet.find(i) - step) % len(alphabet)]  
 elif i.isalpha():  
 return alphabet[(alphabet.find(i.upper()) - step) % len(alphabet)].lower()  
 else:  
 return i  
  
  
def encrypt(message, step=5):  
  
 encrypted\_message = ""  
  
 for i in message:  
 encrypted\_message += func\_encr(i, step)  
  
 return encrypted\_message  
  
  
def decrypt(message, step=5):  
 decrypted\_message = ""  
  
 for i in message:  
 decrypted\_message += func\_decr(i, step)  
  
 return decrypted\_message

vigenere\_encryption.py

from curses.ascii import isupper  
  
# alphabet = "ABCDEFGHIJKLMNOPQRSTYVWXYZ"  
alphabet = "АБВГДЕЁЖЗИКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ"  
  
  
def get\_final\_key(key, message):  
 final\_key = key  
  
 while len(final\_key) < len(message):  
 final\_key += key  
  
 return final\_key  
  
  
def func\_encr(i, k):  
 if isupper(i) or (1040 <= ord(i) <= 1071):  
 return chr((ord(i) + ord(k)) % len(alphabet) + ord(alphabet[0]))  
 else:  
 return chr((ord(i) - 32 + ord(k)) % len(alphabet) + ord(alphabet[0]) + 32)  
  
  
def encrypt(message, key='LEMON'):  
 key = get\_final\_key(key, message)  
  
 encrypted\_message = "".join(map(func\_encr, message, key))  
  
 return encrypted\_message  
  
  
def func\_decr(i, k):  
 if isupper(i) or (1040 <= ord(i) <= 1071):  
 return chr((ord(i) - ord(k)) % len(alphabet) + ord(alphabet[0]))  
 else:  
 return chr((ord(i) - 32 - ord(k)) % len(alphabet) + ord(alphabet[0]) + 32)  
  
  
def decrypt(message, key='LEMON'):  
 key = get\_final\_key(key, message)  
  
 decrypted\_message = "".join(map(func\_decr, message, key))  
  
 return decrypted\_message