

Министерство науки и высшего образования РФ
Федеральное государственное автономное образовательное
учреждение
высшего образования
«Московский политехнический университет»
(Московский политех)

Отчёт по курсу «Программирование криптографических алгоритмов»
Лабораторная работа 1. Шифрование шифрами однозначной замены



Выполнил:

Студент группы 221-352

Иванов В. В.

Проверил преподаватель: Бутакова Н. Г.

Москва 2024г.

Аннотация

- **Среда программирования**

- Visual Studio Code

- **Язык программирования**

- Python

- **Процедуры для запуска программы**

- Visual Studio Code (main.py)

- **Пословица-тест**

- Тот, кто ложится на два стула, падает на ребра.

- **Текст для проверки работы (не меньше 1000 знаков (1430))**

Жизнь - это удивительное приключение, полное разнообразных событий и встреч. В каждом моменте мы находим что-то новое и уникальное. Стремление к росту и саморазвитию вдохновляет нас на поиск новых горизонтов. Важно помнить, что каждый шаг вперед приносит с собой уроки и опыт.

Разнообразие культур, языков и традиций делает наш мир удивительно богатым. Общение с людьми разных национальностей расширяет кругозор, позволяя нам понимать и уважать друг друга. Взаимное уважение и терпимость создают основу для гармоничного сосуществования.

Природа тоже играет важную роль в нашей жизни. Красота закатов, шум океана, пение птиц - все это напоминает нам о величии мира природы. Забота о окружающей среде становится неотъемлемой частью ответственного образа жизни.

Работа и творчество придают смысл нашим усилиям. Стремление к достижению целей мотивирует нас на новые начинания. Каждый проект, даже самый маленький, приносит удовлетворение и чувство выполненного долга.

Семья и друзья являются надежной опорой в нашей жизни. Обмен историями, веселые посиделки и поддержка в трудные моменты создают теплую атмосферу взаимопонимания и любви.

Таким образом, наша жизнь - это мозаика различных моментов, соединенных воедино. Важно ценить каждый момент и стремиться делать мир вокруг нас ярче и лучше. С любовью, терпением и целеустремленностью мы можем создавать свою уникальную историю, наполненную смыслом и радостью.

- Код программы-интерфейса

```
• import sys
• from PyQt5.QtWidgets import QApplication, QWidget, QVBoxLayout, QHBoxLayout, QLabel, QLineEdit, QPushButton, QComboBox,
  QTextEdit, QCheckBox
• from PyQt5.QtCore import Qt
• from atbash import atbash_encrypt, atbash_decrypt
• from cesar import cesar_encrypt, cesar_decrypt, cesar_check_parameters
• from polibia import polibia_encrypt, polibia_decrypt
• from tritemiy import tritemiy_encrypt, tritemiy_decrypt
• from belazo import belazo_encrypt, belazo_decrypt, belazo_check_parameters
• from vigenere import vigenere_encrypt, vigenere_decrypt, vigenere_check_parameters
• # from S_block import s_block, combine_s_blocks, circular_shift_left
• from matrix import matrix_encrypt, matrix_decrypt, matrix_check_parameters, multiply_matrix, determinant,
  adjugate_matrix, inverse_matrix
• from playfair import playfair_encrypt, playfair_decrypt, playfair_check_parameters, get_alphabet_index
•
• available_ciphers = [
•     "Шифр АТБАШ", "Шифр Цезаря", "Шифр Полибия",
•     "Шифр Тритемия", "Шифр Белазо", "Шифр Виженера", "МАГМА(s_block)",
•     "Шифр Матричный", "Шифр Плейфера",
• ]
•
• alphabet = [
•     "а", "б", "в", "г", "д", "е", "ж", "з", "и", "й", "к", "л", "м",
•     "н", "о", "п", "р", "с", "т", "у", "ф", "х", "ц", "ч", "ш", "щ",
•     "ъ", "ы", "ь", "э", "ю", "я"
• ]
•
• alphabet_polibia = [
•     ["а", "б", "в", "г", "д", "е"],
•     ["ж", "з", "и", "й", "к", "л"],
•     ["м", "н", "о", "п", "р", "с"],
```

```

•     ["т", "у", "ф", "х", "ц", "ч"],
•     ["ш", "щ", "ь", "ы", "ъ", "э"],
•     ["ю", "я"]
• ]
•
• alphabet_playfair = [
•     "а", "б", "в", "г", "д", "е", "ж", "з", "и", "к", "л", "м", "н",
•     "о", "п", "р", "с", "т", "у", "ф", "х", "ц", "ч", "ш", "щ", "ь",
•     "ы", "э", "ю", "я"
• ]
•
• sBlocks = {
•     7: {'0': 'c', '1': '4', '2': '6', '3': '2', '4': 'a', '5': '5', '6': 'b', '7': '9', '8': 'e', '9': '8', 'a': 'd',
•         'b': '7', 'c': '0', 'd': '3', 'e': 'f', 'f': '1'},
•     6: {'0': '6', '1': '8', '2': '2', '3': '3', '4': '9', '5': 'a', '6': '5', '7': 'c', '8': '1', '9': 'e', 'a': '4',
•         'b': '7', 'c': 'b', 'd': 'd', 'e': '0', 'f': 'f'},
•     5: {'0': 'b', '1': '3', '2': '5', '3': '8', '4': '2', '5': 'f', '6': 'a', '7': 'd', '8': 'e', '9': '1', 'a': '7',
•         'b': '4', 'c': 'c', 'd': '9', 'e': '6', 'f': '0'},
•     4: {'0': 'c', '1': '8', '2': '2', '3': '1', '4': 'd', '5': '4', '6': 'f', '7': '6', '8': '7', '9': '0', 'a': 'a',
•         'b': '5', 'c': '3', 'd': 'e', 'e': '9', 'f': 'b'},
•     3: {'0': '7', '1': 'f', '2': '5', '3': 'a', '4': '8', '5': '1', '6': '6', '7': 'd', '8': '0', '9': '9', 'a': '3',
•         'b': 'e', 'c': 'b', 'd': '4', 'e': '2', 'f': 'c'},
•     2: {'0': '5', '1': 'd', '2': 'f', '3': '6', '4': '9', '5': '2', '6': 'c', '7': 'a', '8': 'b', '9': '7', 'a': '8',
•         'b': '1', 'c': '4', 'd': '3', 'e': 'e', 'f': '0'},
•     1: {'0': '8', '1': 'e', '2': '2', '3': '5', '4': '6', '5': '9', '6': '1', '7': 'c', '8': 'f', '9': '4', 'a': 'b',
•         'b': '0', 'c': 'd', 'd': 'a', 'e': '3', 'f': '7'},
•     0: {'0': '1', '1': '7', '2': 'e', '3': 'd', '4': '0', '5': '5', '6': '8', '7': '3', '8': '4', '9': 'f', 'a': 'a',
•         'b': '6', 'c': '9', 'd': 'c', 'e': 'b', 'f': '2'}
• }
•
• reversedSBlocks = {

```

```

• 7: {'c': '0', '4': '1', '6': '2', '2': '3', 'a': '4', '5': '5', 'b': '6', '9': '7', 'e': '8', '8': '9', 'd': 'a',
• '7': 'b', '0': 'c', '3': 'd', 'f': 'e', '1': 'f'},
• 6: {'6': '0', '8': '1', '2': '2', '3': '3', '9': '4', 'a': '5', '5': '6', 'c': '7', '1': '8', 'e': '9', '4': 'a',
• '7': 'b', 'b': 'c', 'd': 'd', '0': 'e', 'f': 'f'},
• 5: {'b': '0', '3': '1', '5': '2', '8': '3', '2': '4', 'f': '5', 'a': '6', 'd': '7', 'e': '8', '1': '9', '7': 'a',
• '4': 'b', 'c': 'c', '9': 'd', '6': 'e', '0': 'f'},
• 4: {'c': '0', '8': '1', '2': '2', '1': '3', 'd': '4', '4': '5', 'f': '6', '6': '7', '7': '8', '0': '9', 'a': 'a',
• '5': 'b', '3': 'c', 'e': 'd', '9': 'e', 'b': 'f'},
• 3: {'7': '0', 'f': '1', '5': '2', 'a': '3', '8': '4', '1': '5', '6': '6', 'd': '7', '0': '8', '9': '9', '3': 'a',
• 'e': 'b', 'b': 'c', '4': 'd', '2': 'e', 'c': 'f'},
• 2: {'5': '0', 'd': '1', 'f': '2', '6': '3', '9': '4', '2': '5', 'c': '6', 'a': '7', 'b': '8', '7': '9', '8': 'a',
• '1': 'b', '4': 'c', '3': 'd', 'e': 'e', '0': 'f'},
• 1: {'8': '0', 'e': '1', '2': '2', '5': '3', '6': '4', '9': '5', '1': '6', 'c': '7', 'f': '8', '4': '9', 'b': 'a',
• '0': 'b', 'd': 'c', 'a': 'd', '3': 'e', '7': 'f'},
• 0: {'1': '0', '7': '1', 'e': '2', 'd': '3', '0': '4', '5': '5', '8': '6', '3': '7', '4': '8', 'f': '9', 'a': 'a',
• '6': 'b', '9': 'c', 'c': 'd', 'b': 'e', '2': 'f'}
• }
•
• mem = {
•     "bigTextFlag": False,
•     "vigenereSwitch": False,
•     "mode": "encrypt",
• }
•
• class CipherApp(QWidget):
•     def __init__(self):
•         super().__init__()
•         self.initUI()
•
•     def initUI(self):
•         self.setWindowTitle('Шифры')
•         self.resize(960, 640)

```

```
• layout = QVBoxLayout()
•
• # Выбор шифра
• cipher_layout = QHBoxLayout()
• cipher_label = QLabel('Выберите шифр:')
• self.cipher_combo = QComboBox()
• self.cipher_combo.addItem(available_ciphers)
• cipher_layout.addWidget(cipher_label)
• cipher_layout.addWidget(self.cipher_combo)
•
• # Ввод открытого текста
• open_text_label = QLabel('Введите открытый текст(Расшифрованный):')
• self.open_text_edit = QTextEdit()
•
• # Ввод зашифрованного текста
• cipher_text_label = QLabel('Шифрованный текст:')
• self.cipher_text_edit = QTextEdit()
•
• # Ввод сдвига для шифра Цезаря
• self.cesar_shift_edit = QLineEdit()
• self.cesar_shift_edit.setPlaceholderText('Введите сдвиг для шифра Цезаря')
• self.cesar_shift_edit.textChanged.connect(self.check_cesar_shift)
•
• # Ввод ключевого слова для шифра Белазо или Плейфера
• self.keyword_edit = QLineEdit()
• self.keyword_edit.setPlaceholderText('Введите ключевое слово для шифра Белазо или Плейфера')
•
• # Ввод ключевой буквы для шифра Виженера
• self.vigener_key_edit = QLineEdit()
• self.vigener_key_edit.setPlaceholderText('Введите ключевую букву для шифра Виженера')
• self.vigener_key_edit.textChanged.connect(self.check_vigener_key)
```

```
• # Ввод ключевой матрицы для шифра Матричный
• self.matrix_edit = QLineEdit()
• self.matrix_edit.setPlaceholderText('Введите ключевую матрицу для шифра Матричный')
•
• # Режим работы шифра (шифрование или дешифрование)
• mode_layout = QHBoxLayout()
• mode_label = QLabel('Выберите режим:')
• self.mode_combo = QComboBox()
• self.mode_combo.addItem('Шифрование', 'Расшифрование'])
• mode_layout.addWidget(mode_label)
• mode_layout.addWidget(self.mode_combo)
•
• # Кнопка для запуска шифрования/дешифрования
• self.encrypt_button = QPushButton('Выполнить')
•
• layout.addLayout(cipher_layout)
• layout.addWidget(open_text_label)
• layout.addWidget(self.open_text_edit)
• layout.addWidget(cipher_text_label)
• layout.addWidget(self.cipher_text_edit)
• layout.addWidget(self.cesar_shift_edit)
• layout.addWidget(self.keyword_edit)
• layout.addWidget(self.vigener_key_edit)
• layout.addWidget(self.matrix_edit)
• layout.addLayout(mode_layout)
• layout.addWidget(self.encrypt_button)
•
• self.setLayout(layout)
•
• # Переключатель для выбора режима текста
• self.text_mode_checkbox = QCheckBox('Расширенный текст')
• layout.addWidget(self.text_mode_checkbox)
```

```

•
•     # Подключение слотов к сигналам
•
•     self.encrypt_button.clicked.connect(self.cipher_parser)
•     self.text_mode_checkbox.stateChanged.connect(self.handle_text_mode_change)
•
•
• def handle_text_mode_change(self, state):
•     if state == Qt.Checked:
•         mem["bigTextFlag"] = True
•     else:
•         mem["bigTextFlag"] = False
•
•
• def check_cesar_shift(self):
•     shift_text = self.cesar_shift_edit.text()
•     try:
•         shift = int(shift_text)
•         if shift < 0 or shift >= len(alphabet):
•             self.cesar_shift_edit.setStyleSheet("QLineEdit { color: red; }")
•         else:
•             self.cesar_shift_edit.setStyleSheet("")
•     except ValueError:
•         self.cesar_shift_edit.setStyleSheet("QLineEdit { color: red; }")
•
•
• def check_vigener_key(self):
•     key_text = self.vigener_key_edit.text()
•     if len(key_text) != 1 or key_text.lower() not in alphabet:
•         self.vigener_key_edit.setStyleSheet("QLineEdit { color: red; }")
•     else:
•         self.vigener_key_edit.setStyleSheet("")
•
•
• def text_preparation(self, text):
•     bigTextFlag = mem["bigTextFlag"]
•     if bigTextFlag:

```



```

•         # Обработка расширенного текста
•         return text.replace("ё", "е").replace(".", "тчк").replace(",", "зпт").replace("-", "тире").replace(" ",
"прбл").replace(":", "двтч").replace(";", "тчсзп").replace("(", "отскб").replace(")", "зкскб").replace("?",
"впрзн").replace("!", "восклзн").replace("\n", "првст").lower()
•     else:
•         # Обработка обычного текста
•         return text.replace("ё", "е").replace(".", "тчк").replace(",", "зпт").replace("-", "тире").replace(" ",
"").replace(":", "").replace(";", "").replace("(", "").replace(")", "").replace("?", "").replace("!", "").replace("\n",
"").lower()
•
•     def cipher_parser(self):
•         cipher_choose_input = self.cipher_combo.currentText()
•         open_text_input = self.open_text_edit.toPlainText()
•         cipher_text_input = self.cipher_text_edit.toPlainText()
•         cesar_shift = self.cesar_shift_edit.text()
•         keyword = self.keyword_edit.text()
•         vigenere_keyletter = self.vigenere_key_edit.text()
•         matrix_input = self.matrix_edit.text()
•
•         # Определение режима работы (шифрование или дешифрование)
•         mode = 'encrypt' if self.mode_combo.currentText() == 'Шифрование' else 'decrypt'
•
•         # Определение флага для обработки больших текстов
•         bigTextFlag = len(open_text_input) > 1000 #ваш порог длины текста
•
•         if cipher_choose_input == "Шифр АТБАШ":
•             if mode == "encrypt":
•                 cipher_text_input = atbash_encrypt(self.text_preparation(open_text_input), alphabet)
•             elif mode == "decrypt":
•                 open_text_input = atbash_decrypt(cipher_text_input, alphabet)
•         elif cipher_choose_input == "Шифр Цезаря":
•             if cesar_shift: # Проверка на пустую строку

```

```

•         cesar_shift = int(cesar_shift)
•         if cesar_check_parameters(cesar_shift, alphabet):
•             if mode == "encrypt":
•                 cipher_text_input = cesar_encrypt(self.text_preparation(open_text_input), cesar_shift,
alphabet)
•             elif mode == "decrypt":
•                 open_text_input = cesar_decrypt(cipher_text_input, cesar_shift, alphabet)
•         else:
•             if mode == "encrypt":
•                 cipher_text_input = "Проверьте правильность ввода сдвига"
•             elif mode == "decrypt":
•                 open_text_input = "Проверьте правильность ввода сдвига"
•         else:
•             if mode == "encrypt":
•                 cipher_text_input = "Введите сдвиг для шифра Цезаря"
•             elif mode == "decrypt":
•                 open_text_input = "Введите сдвиг для шифра Цезаря"
•         elif cipher_choose_input == "Шифр Полибия":
•             if mode == "encrypt":
•                 cipher_text_input = polibia_encrypt(self.text_preparation(open_text_input), alphabet_polibia)
•             elif mode == "decrypt":
•                 open_text_input = polibia_decrypt(cipher_text_input, alphabet_polibia)
•         elif cipher_choose_input == "Шифр Тритемия":
•             if mode == "encrypt":
•                 cipher_text_input = tritemiy_encrypt(self.text_preparation(open_text_input), alphabet)
•             elif mode == "decrypt":
•                 open_text_input = tritemiy_decrypt(cipher_text_input, alphabet)
•         elif cipher_choose_input == "Шифр Белазо":
•             if keyword:
•                 if belazo_check_parameters(keyword.lower(), alphabet):
•                     if mode == "encrypt":

```

```

•         cipher_text_input = belazo_encrypt(self.text_preparation(open_text_input), keyword.lower(),
alphabet)
•
•         elif mode == "decrypt":
•             open_text_input = belazo_decrypt(cipher_text_input, keyword.lower(), alphabet)
•
•     else:
•         if mode == "encrypt":
•             cipher_text_input = "Проверьте правильность ввода ключевого слова"
•         elif mode == "decrypt":
•             open_text_input = "Проверьте правильность ввода ключевого слова"
•
•     else:
•         if mode == "encrypt":
•             cipher_text_input = "Введите ключевое слово для шифра Белазо"
•         elif mode == "decrypt":
•             open_text_input = "Введите ключевое слово для шифра Белазо"
•     elif cipher_choose_input == "Шифр Виженера":
•         if vigenere_keyletter:
•             if vigenere_check_parameters(vigenere_keyletter, alphabet):
•                 if mode == "encrypt":
•                     cipher_text_input = vigenere_encrypt(self.text_preparation(open_text_input), vigenere_keyletter,
"selfkey", alphabet)
•                 elif mode == "decrypt":
•                     open_text_input = vigenere_decrypt(cipher_text_input, vigenere_keyletter, "selfkey", alphabet)
•             else:
•                 if mode == "encrypt":
•                     cipher_text_input = "Проверьте правильность ввода ключевой буквы"
•                 elif mode == "decrypt":
•                     open_text_input = "Проверьте правильность ввода ключевой буквы"
•         else:
•             if mode == "encrypt":
•                 cipher_text_input = "Введите ключевую букву для шифра Виженера"
•             elif mode == "decrypt":
•                 open_text_input = "Введите ключевую букву для шифра Виженера"

```

```

• # elif cipher_choose_input == "МАГМА(s_block)":
• #     if mode == "encrypt":
• #         cipher_text_input = s_block(cipher_text_input)
• #     elif mode == "decrypt":
• #         open_text_input = s_block(open_text_input, reversedSBlocks)
• elif cipher_choose_input == "Шифр Матричный":
•     input_matrix = list(map(int, matrix_input.split()))
•     matrix_input = [input_matrix[:3], input_matrix[3:6], input_matrix[6:]]
•     if matrix_input:
•         if matrix_check_parameters(matrix_input):
•             if mode == "encrypt":
•                 cipher_text_input = matrix_encrypt(self.text_preparation(open_text_input), matrix_input,
alphabet)
•             elif mode == "decrypt":
•                 open_text_input = matrix_decrypt(cipher_text_input, matrix_input, alphabet)
•         else:
•             if mode == "encrypt":
•                 cipher_text_input = "Проверьте правильность ввода матрицы"
•             elif mode == "decrypt":
•                 open_text_input = "Проверьте правильность ввода матрицы"
•         else:
•             if mode == "encrypt":
•                 cipher_text_input = "Введите ключевую матрицу для шифра Матричный"
•             elif mode == "decrypt":
•                 open_text_input = "Введите ключевую матрицу для шифра Матричный"
• elif cipher_choose_input == "Шифр Плейфера":
•     if keyword:
•         if playfair_check_parameters(keyword, alphabet_playfair):
•             if mode == "encrypt":
•                 cipher_text_input = playfair_encrypt(self.text_preparation(open_text_input), keyword,
alphabet_playfair)
•             elif mode == "decrypt":

```

```

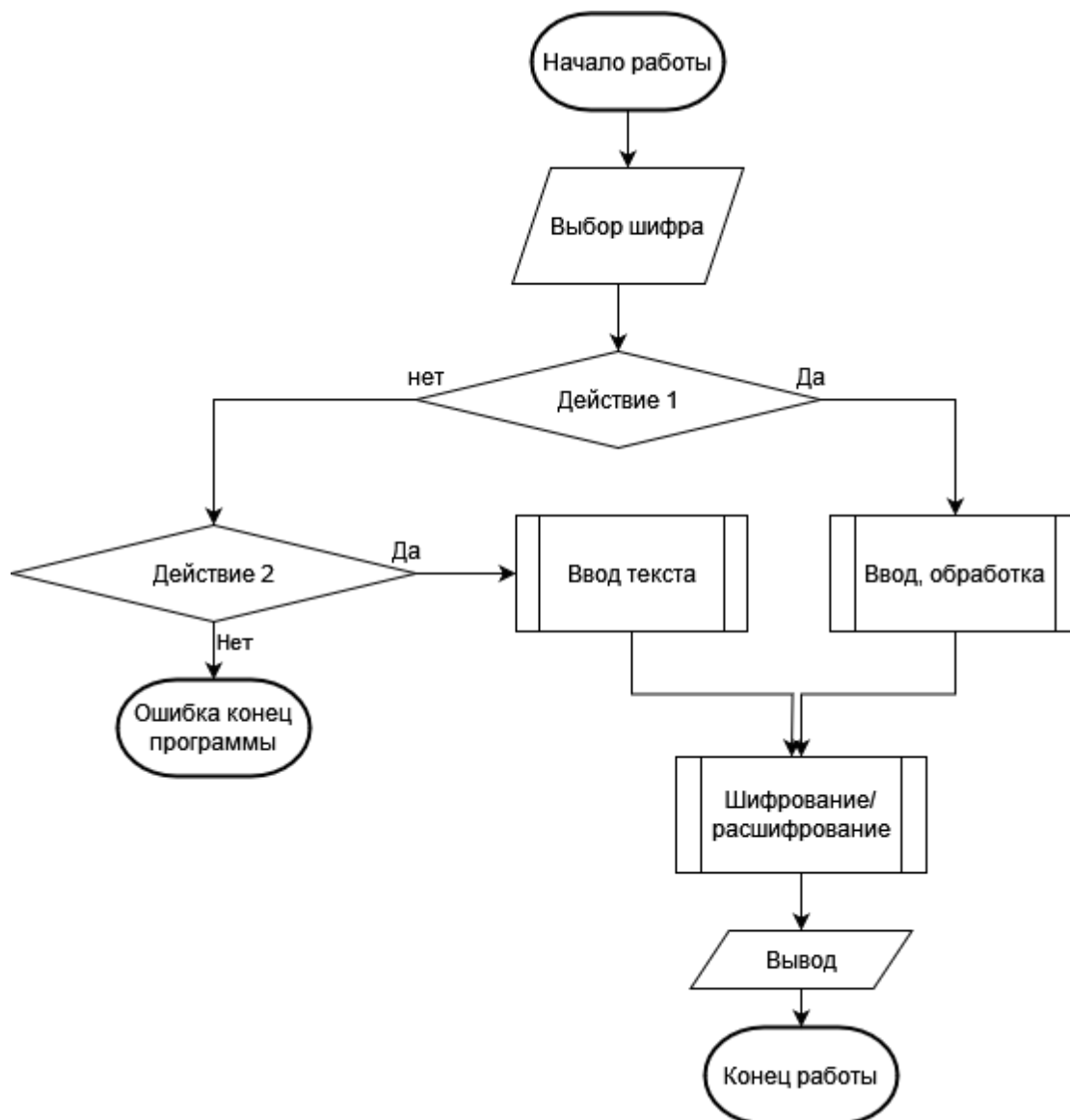
•         open_text_input = playfair_decrypt(cipher_text_input, keyword, alphabet_playfair)
•     else:
•         if mode == "encrypt":
•             cipher_text_input = "Проверьте правильность ввода ключевого слова"
•         elif mode == "decrypt":
•             open_text_input = "Проверьте правильность ввода ключевого слова"
•     else:
•         if mode == "encrypt":
•             cipher_text_input = "Введите ключевое слово для шифра Плейфера"
•         elif mode == "decrypt":
•             open_text_input = "Введите ключевое слово для шифра Плейфера"
•     else:
•         pass
•
•     # Обновление текста в виджетах
•     self.open_text_edit.setPlainText(open_text_input)
•     self.cipher_text_edit.setPlainText(cipher_text_input)
•
• if __name__ == '__main__':
•     app = QApplication(sys.argv)
•     ex = CipherApp()
•     ex.show()
•     sys.exit(app.exec_())

```

1. Шифр простой замены АТБАШ

Шифр Атбаш — это простой шифр подстановки для алфавитного письма, в котором каждая n -я буква алфавита заменяется буквой $m - n + 1$, где m — общее число букв в алфавите. Другими словами, первая буква заменяется на последнюю, вторая — на предпоследнюю и так далее.

Блок-схема программы



Код программы с комментариями

```
def atbash_encrypt(open_text, alphabet):
    encrypted_text = "" # Шифртекст
    for element in open_text: # Проход по всем символам открытого текста
        encrypted_text += alphabet[len(alphabet) - alphabet.index(element) - 1] # Добавление в итоговый шифртекст
зашифрованного символа
    return encrypted_text # Возврат шифртекста

def atbash_decrypt(encrypted_text, alphabet):
    decrypted_text = "" # Расшифрованный текст
    for element in encrypted_text: # Проход по всем символам шифртекста
        decrypted_text += alphabet[len(alphabet) - alphabet.index(element) - 1] # Добавление в итоговый текст расшифрованного
символа
    # Перевод символов из их текстовых значений в символные
    decrypted_text = decrypted_text.replace("тчк", ".").replace("зпт", ",").replace("тире", "-").replace('прбл', ' ')
.replace('двтч', ':').replace('тчсзп', ';').replace('отскб', '(').replace('зксб', ')').replace('впрзн',
'').replace('восклзн', '!').replace('првст', '\n')
    return decrypted_text # Возврат расшифрованного текста
```

Тестирование

Шифры

Выберите шифр: Шифр АТБАШ

Введите открытый текст:
Тот, кто ложится на два стула, падает на ребра.

Введите зашифрованный текст:
нсншрнхсфщчноатязяоннфяшрнряянтяпьюпях

Введите сдвиг для шифра Цезаря

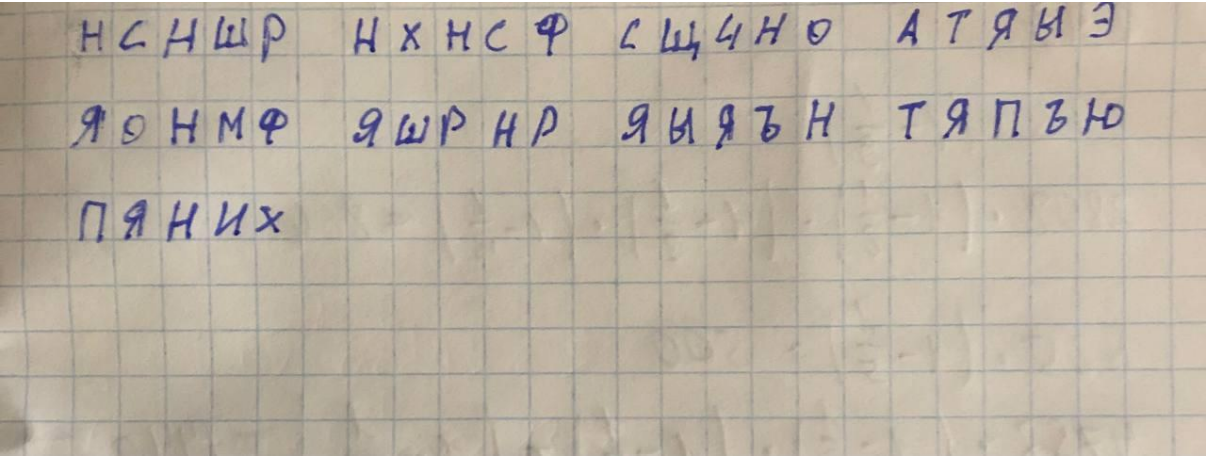
Введите ключевое слово для шифра Белазо или Плейфера

Введите ключевую букву для шифра Виженера

Введите ключевую матрицу для шифра Матричный

Выберите режим: Шифрование

Выполнить



Зашифрование

☒ Расширенный текст

Расшифрование

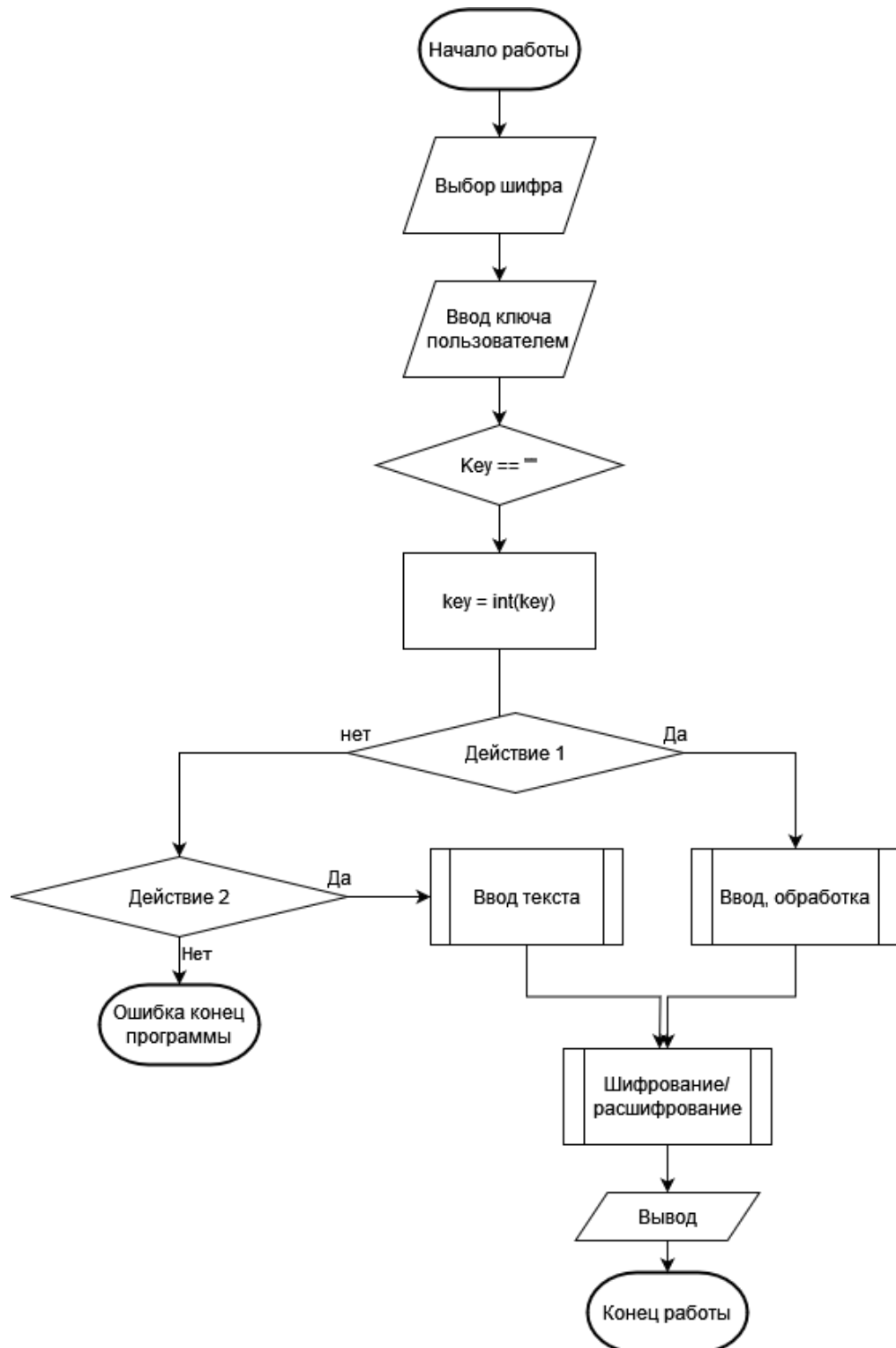
[illegible]

2. ШИФР ЦЕЗАРЯ

Шифр Цезаря — это шифр, при использовании которого каждая буква из открытого текста заменяется на такую букву, которая в алфавите находится на некотором постоянном числе позиций левее или правее от рассматриваемой буквы.

Например, при сдвиге букв русского алфавита вправо на 3 позиции буква «А» заменяется на букву «Г», буква «Б» заменяется на букву «Д» и так далее

Блок-схема программы



Код программы с комментариями

```
def cesar_check_parameters(shift, alphabet):
    if shift <= len(alphabet) - 1:
        return True # Возврат истины, если сдвиг меньше длины алфавита
    return False # Возврат лжи, если сдвиг больше длины алфавита

def cesar_encrypt(open_text, shift, alphabet):
    encrypted_text = "" # Шифртекст
    for i in range(len(open_text)): # Проход по всем символам открытого текста
        element = open_text[i] # Символ
        encrypted_text += alphabet[(alphabet.index(element) + shift) % len(alphabet)] # Добавление в итоговый шифртекст
        # зашифрованного символа
    return encrypted_text # Возврат шифртекста

def cesar_decrypt(encrypted_text, shift, alphabet):
    decrypted_text = "" # Расшифрованный текст
    for i in range(len(encrypted_text)): # Проход по всем символам шифртекста
        element = encrypted_text[i] # Символ
        decrypted_text += alphabet[(alphabet.index(element) - shift + len(alphabet)) % len(alphabet)] # Добавление в итоговый
        # текст расшифрованного символа
    # Перевод символов из их текстовых значений в символные
    decrypted_text = decrypted_text.replace("тчк", ".").replace("зпт", ",").replace("тире", "-").replace('прбл', ' ')
    decrypted_text = decrypted_text.replace('двтч', ':').replace('тчсзп', ';').replace('отскб', '(').replace('зксб', ')').replace('впрзн',
    '?').replace('восклзн', '!').replace('првст', '\n')
    return decrypted_text # Возврат расшифрованного текста
```

Тестирование

Шифры

Выберите шифр:

Шифр Цезаря

Введите открытый текст:

Тот, кто ложится на два стула, падает на ребра.

Введите зашифрованный текст:

хсхктхнхсойлхфвргзегфхцогктхтгзгигхргуйдугхън

3

Введите ключевое слово для шифра Белазо или Плейфера

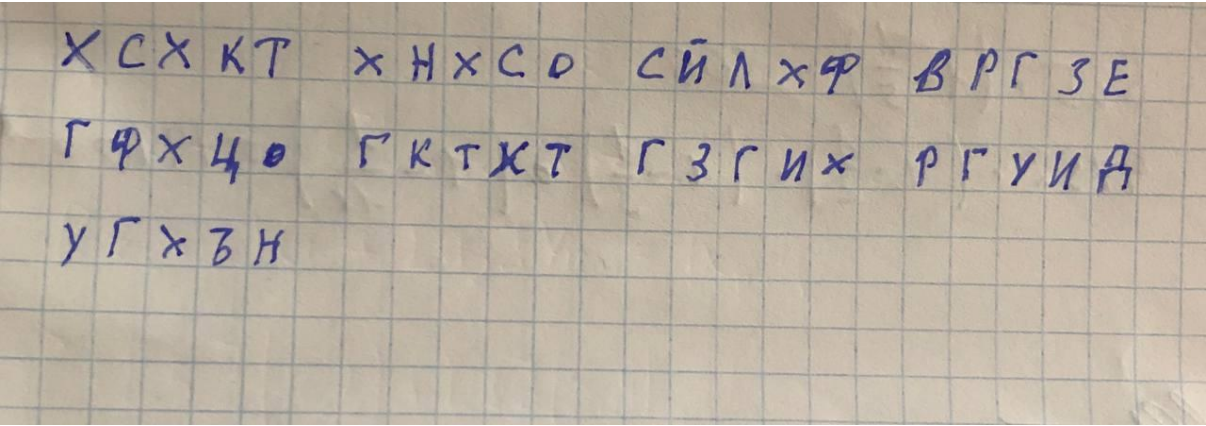
Введите ключевую букву для шифра Виженера

Введите ключевую матрицу для шифра Матричный

Выберите режим:

Шифрование

Выполнить



Зашифрование

☒ Расширенный текст

Расшифрование

Шифры

Выберите шифр:

Шифр Цезаря

Введите открытый текст (Расшифрованный):

жизнь - это удивительное приключение, полное разнообразных событий и встреч. в каждом моменте мы находим что-то новое и уникальное. стремление к росту и саморазвитию вдохновляет нас на поиск новых горизонтов. важно помнить, что каждый шаг вперед приносит с собой уроки и опыт.

разнообразие культур, языков и традиций делает наш мир удивительно богатым. общение с людьми разных национальностей расширяет кругозор, позволяя нам понимать и уважать друг друга. взаимное уважение и терпимость создают основу для гармоничного сосуществования.

природа тоже играет важную роль в нашей жизни. красота закатов, шум океана, пение птиц - все это напоминает нам о величии мира природы. забота о окружающей среде становится неотъемлемой частью ответственного образа жизни.

работа и творчество придают смысл нашим усилиям. стремление к достижению целей мотивирует нас на новые начинания. каждый проект, даже самый маленький, приносит удовлетворение и чувство выполненного долга.

семья и друзья являются надежной опорой в нашей жизни. обмен историями, веселые посиделки и поддержка в трудные моменты создают теплую атмосферу взаимопонимания и любви.

Шифрованный текст:

йлкрятудохлуитудоахстуодцзлелхиярситудотулнобьирликтхтудотсорситудоугкрссдугкрюштудофсдюохлнтудолтудоефхуиьхьнтудоетудонгйзсптудопспирхитудопютудоргшзлпттудоъхслуихстудорсеитудолтудоцрлнгоряхихьнтудофхуипоирлитудонтудоусфхцтудолтудофгпсугкелхлбтудоезсрсеовихтудорфтудортудотслфнтудорсеюштудоцояхцуктхтудовкюнсетудолтудохугзлшлнтудозюигихтудоргытудопллтудоцзлелхиярстудодсжхюпхьнтудосдырлитудофтудообзяплтудоугкрюштудоргшлсргоярсфхитудооугфелувихтудонужксуктхтудотокесоветудоргптудотсрлпгхятудолтудоцегйгхятудозуцктудозуцкхьнтудоекглпритудоцегйрлитудолтудохиултлпсфхятудофскзгбхтудосфсецтудозовтудожгупрлрскстудофсфхифхесегрлхьнтудефхтудефтулусзгтудохйитудолжугихтудоегйрбтудоусоятудоетудоргынтудойлкрлхьнтудонугфсхтудокнгбхсекстхтудоситтудоснигркхтхтудотирлитудотхлштудохлуитудоефитудоахстудортсплргихтудоргптудостудоеюльолтудоплуттудотулусзюхьнтудокдсхтудостудоснуцйгбыйнтудофузитудофхгрселхфетудорисжипоипнтудоьгфхябтудоскеихфеиррокстудосдугктудойлкрлхьнтудефхтудефхугдсхтудолтудохесуьфхестудотулзбхтудофлюфотудоргылттудоцфлловпхьнтудофхуипоирлитудонтудозсфхлирлбтудоцоиюнтудопслелуцхтхтудоргфтудорттудорсеюитудоргьлргрлхьнтудонгйюнтудотуснхкхтхтудозгйитудогфпюнтудоптоирялнкхтхтудотулрслфхтудоцсеюихесуирлитудолтудоеьефхестудоеютсориррскстудозсожхьнтудефхтудефхфилпавтудолтудозуцкявтудовевобхфвтудоргзииронтудостсунтудоетудоргынтудойлкрлхьнтудосдпиртудолфхсулвлпктхтудоеифиюнтудотсфлзионлтудолтудотсзиюнгтудоетудохуцрюнтудопспирхютудофскзгбхтудохитоцбтудогхлсфхиуцтудоекглптсрлгрлвтудолтудообделхьнтудефхтудефххглптудосдугкспктхтхтудоргытудойлкрятудохлуитудоахстудопсгкглнтудогукольрштудопспирхсекстхтудофизлррюштудоесизлрсьнтудоегйрстудоширлхятудонгйюнтудопспирхтудолтудофхуиплхяфвтудозюгхятудоплуттудоеснужктудоргфтудовуьнтудолтудооцъ

3

Введите ключевое слово для шифра Белазо или Плейфера

Введите ключевую букву для шифра Вигенера

Введите ключевую матрицу для шифра Матричный

Выберите режим:

Расшифрование

Выполнить

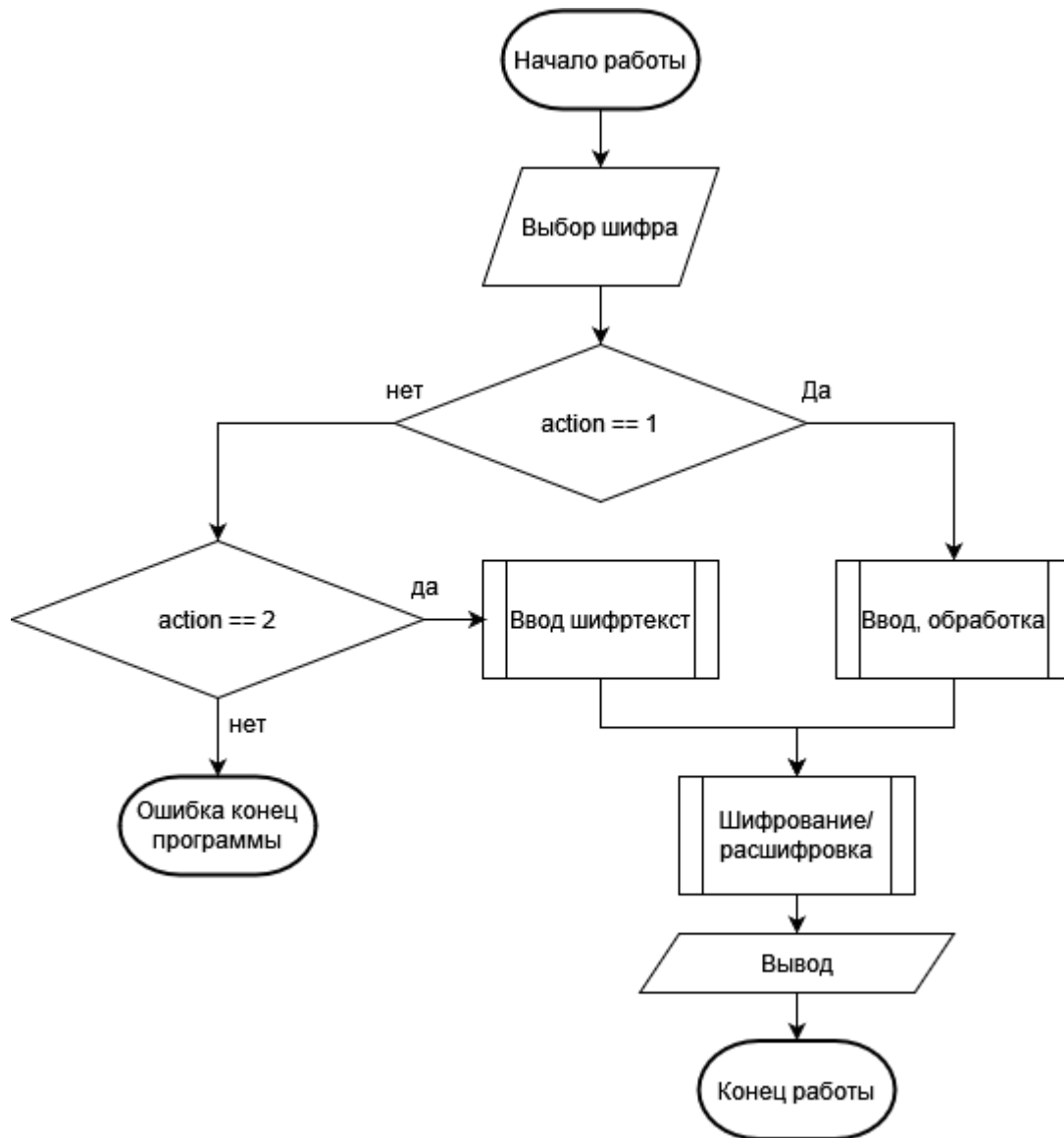
☒ Расширенный текст

3. Квадрат Полибия

Квадрат Полибия — это способ шифрования, разработанный древнегреческим историком и полководцем Полибием.

Он представляет собой квадрат 6х6, столбцы и строки которого нумеруются цифрами от 1 до 6. Все буквы алфавита вписываются в квадрат по одной на клетку. Буквы расположены в алфавитном порядке

Блок-схема программы



Код программы с комментариями

```
def polibia_encrypt(open_text, alphabet):
    encrypted_text = "" # Шифртекст
    for letter in open_text: # Проход по всем символам открытого текста
        for i in range(len(alphabet)): # Проход по строкам алфавита
            row = alphabet[i] # Строка алфавита
            for j in range(len(row)): # Проход по символам строки
                if letter == row[j]: # Если зашифровываемая буква совпадает с текущей
                    encrypted_text += f"{i + 1}{j + 1} " # Добавление в итоговый шифртекст зашифрованного символа
    return encrypted_text # Возврат шифртекста

def polibia_decrypt(encrypted_text, alphabet):
    decrypted_text = "" # Расшифрованный текст
    for letter in encrypted_text.split(): # Проход по всем символам шифртекста
        if letter: # Если буква существует
            row_index = int(letter[0]) - 1
            col_index = int(letter[1]) - 1
            decrypted_text += alphabet[row_index][col_index] # Добавление в итоговый текст расшифрованного символа
    # Перевод символов из их текстовых значений в символные
    decrypted_text = decrypted_text.replace("тчк", ".").replace("зпт", ",").replace("тире", "-").replace('прбл', ' ')
    decrypted_text = decrypted_text.replace('двтч', ':').replace('тчсзп', ';').replace('отскб', '(').replace('зксб', ')').replace('впрзн', '?')
    decrypted_text = decrypted_text.replace('восклзн', '!').replace('првст', '\n')
    return decrypted_text # Возврат расшифрованного текста
```

Тестирование

Шифры

Выберите шифр:

Шифр Полибия

Введите открытый текст:

Тот, кто ложится на два стула, падает на ребра.

Введите зашифрованный текст:

41 33 41 22 34 41 25 41 33 26 33 21 23 41 36 62 32 11 15 13 11 36 41 42 26 11 22 34 41 34 11 15 11 16 41 32 11 35 16 12 35 11 41 46 25

3

Введите ключевое слово для шифра Белазо или Плейфера

Введите ключевую букву для шифра Виженера

Введите ключевую матрицу для шифра Матричный

Выберите режим:

Шифрование

Выполнить

| | | | | | | | | |
|----|----|----|----|----|----|----|----|----|
| 41 | 33 | 41 | 22 | 34 | 41 | 25 | 41 | 33 |
| 26 | 33 | 21 | 23 | 41 | 36 | 62 | 32 | 11 |
| 15 | 13 | 11 | 36 | 41 | 42 | 26 | 11 | 22 |
| 34 | 41 | 34 | 11 | 15 | 11 | 16 | 41 | 32 |
| 11 | 35 | 16 | 12 | 35 | 11 | 41 | 46 | 25 |

Работа с текстом не менее 1000 знаков

Зашифрование

Шифры

Выберите шифр:

Шифр Полибия

Введите открытый текст (Расшифрованный):

Жизнь - это удивительное приключение, полное разнообразных событий и встреч. В каждом моменте мы находим что-то новое и уникальное. Стремление к росту и саморазвитию вдохновляет нас на поиск новых горизонтов. Важно помнить, что каждый шаг вперед приносит с собой уроки и опыт.

Разнообразие культур, языков и традиций делает наш мир удивительно богатым. Общение с людьми разных национальностей расширяет кругозор, позволяя нам понимать и уважать друг друга. Взаимное уважение и терпимость создают основу для гармоничного сосуществования.

Природа тоже играет важную роль в нашей жизни. Красота закатов, шум океана, пение птиц - все это напоминает нам о величии мира природы. Забота о окружающей среде становится неотъемлемой частью ответственного образа жизни.

Работа и творчество придают смысл нашим усилиям. Стремление к достижению целей мотивирует нас на новые начинания. Каждый проект, даже самый маленький, приносит удовлетворение и чувство выполненного долга.

Семья и друзья являются надежной опорой в нашей жизни. Обмен историями, веселые посиделки и поддержка в трудные моменты создают теплую атмосферу взаимопонимания и любви.

Шифрованный текст:

21 23 22 32 55 34 35 12 26 41 23 35 16 34 35 12 26 56 41 33 34 35 12 26 42 15 23 13 23 41 16 26 55 32 33 16 34 35 12 26 34 35 23 25 26 61 46 16 32 23 16 22 34 41 34 35 12 26 34 33 26 32 33 16 34 35 12 26 35 11 22 32 33 33 12 35 11 22 32 54 44 34 35 12 26 36 33 12 54 41 23 24 34 35 12 26 23 34 35 12 26 13 36 41 35 16 46 41 46 25 34 35 12 26 13 34 35 12 26 25 11 21 15 33 31 34 35 12 26 31 33 31 16 32 41 16 34 35 12 26 31 54 34 35 12 26 32 11 44 33 15 23 31 34 35 12 26 46 41 33 41 23 35 16 41 33 34 35 12 26 32 33 13 33 16 34 35 12 26 23 34 35 12 26 32 33 13 33 16 34 35 12 26 36 11 31 33 35 11 34 35 12 26 42 32 23 25 11 26 55 32 33 16 41 46 25 34 35 12 26 36 41 35 16 31 26 16 32 23 16 34 35 12 26 25 34 35 12 26 35 33 36 41 42 34 35 12 26 23 34 35 12 26 36 11 31 33 35 11 22 13 23 41 23 61 34 35 12 26 13 15 33 44 32 33 13 26 62 16 41 34 35 12 26 32 11 36 34 35 12 26 32 11 34 35 12 26 34 33 23 36 25 34 35 12 26 32 33 13 54 44 34 35 12 26 14 33 35 23 22 33 32 41 33 13 41 46 25 34 35 12 26 13 11 21 32 33 34 35 12 26 34 33 31 32 23 41 55 22 34 41 34 35 12 26 46 41 33 34 35 12 26 25 11 21 15 54 24 34 35 12 26 51 11 14 34 35 12 26 13 34 16 35 16 15 34 35 12 26 34 35 23 32 33 36 23 41 34 35 12 26 36 34 35 12 26 36 33 12 33 24 34 35 12 26 42 35 33 25 23 34 35 12 26 23 34 35 12 26 33 34 54 41 41 46 25 34 35 13 36 41 34 35 13 36 41 35 11 22 32 33 33 12 35 11 22 23 16 34 35 12 26 25 42 26 55 41 42 35 22 34 41 34 35 12 26 62 22 54 25 33 13 34 35 12 26 23 34 35 12 26 41 35 11 15 23 45 23 24 34 35 12 26 15 16 26 11 16 41 34 35 12 26 32 11 51 34 35 12 26 31 23 35 34 35 12 26 42 15 23 13 23 41 16 26 55 32 33 34 35 12 26 12 33 14 11 41 54 31 41 46 25 34 35 12 26 33 12 52 16 32 23 16 34 35 12 26 36 34 35 12 26 26 61 15 55 31 23 34 35 12 26 35 11 22 32 54 44 34 35 12 26 32 11 45 23 33 32 11 26 55 32 33 36 41 16 24 34 35 12 26 35 11 36 51 23 35 62 16 41 34 35 12 26 25 35 42 14 33 22 33 35 22 34 41 34 35 12 26 34 33 22 13 33 26 62 62 34 35 12 26 32 11 31 34 35 12 26 34 33 32 23 31 11 41 55 34 35 12 26 23 34 35 12 26 42 13 11 21 11 41 55 34 35 12 26 15 35 42 14 34 35 12 26 15 35 42 14 11 41 46 25 34 35 12 26 13 22 11 23 31 32 33 16 34 35 12 26 42 13 11 21 16 32 23 16 34 35 12 26 23 34 35 12 26 41 16 35 34 23 31 33 36 41 55 34 35 12 26 36 33 22 15 11 61 41 34 35 12 26 33 36 32 33 13 42 34 35 12 26 15 26 62 34 35 12 26 14 11 35 31 33 32 23 46 32 33 14 33 34 35 12 26 36 33 36 42 52 16 36 41 13 33 13 11 32 23 62 41 46 25 34 35 13 36 41 34 35 13 36 41 34 35 23 35 33 15 11 34 35 12 26 41 33 21 16 34 35 12 26 23 14 35 11 16 41 34 35 12 26 13 11 21 32 42 61 34 35 12

3

Введите ключевое слово для шифра Белазо или Плейфера

Введите ключевую букву для шифра Виженера

Введите ключевую матрицу для шифра Матричный

Выберите режим:

Шифрование

Выполнить

☒ Расширенный текст

Расшифрование

Шифры

Выберите шифр:

Шифр Полибия

Введите открытый текст(Расшифрованный):

жизнь - это удивительное приключение, полное разнообразных событий и встреч. в каждом моменте мы находим что-то новое и уникальное. стремление к росту и саморазвитию вдохновляет нас на поиск новых горизонтов. важно помнить, что каждый шаг вперед приносит с собой уроки и опыт.

разнообразие культур, языков и традиций делает наш мир удивительно богатым. общение с людьми разных национальностей расширяет кругозор, позволяя нам понимать и уважать друг друга. взаимное уважение и терпимость создают основу для гармоничного сосуществования.

природа тоже играет важную роль в нашей жизни. красота закатов, шум океана, пение птиц - все это напоминает нам о величии мира природы. забота о окружающей среде становится неотъемлемой частью ответственного образа жизни.

работа и творчество придают смысл нашим усилиям. стремление к достижению целей мотивирует нас на новые начинания. каждый проект, даже самый маленький, приносит удовлетворение и чувство выполненного долга.

семья и друзья являются надежной опорой в нашей жизни. обмен историями, веселые посиделки и поддержка в трудные моменты создают теплую атмосферу взаимопонимания и любви.

Шифрованный текст:

21 23 22 32 55 34 35 12 26 41 23 35 16 34 35 12 26 56 41 33 34 35 12 26 42 15 23 13 23 41 16 26 55 32 33 16 34 35 12 26 34 35 23 25 26 61 46 16 32 23 16 22 34 41 34 35 12 26 34 33 26 32 33 16 34 35 12 26 35 11 22 32 33 33 12 35 11 22 32 54 44 34 35 12 26 36 33 12 54 41 23 24 34 35 12 26 23 34 35 12 26 13 36 41 35 16 46 41 46 25 34 35 12 26 13 34 35 12 26 25 11 21 15 33 31 34 35 12 26 31 33 31 16 32 41 16 34 35 12 26 31 54 34 35 12 26 32 11 44 33 15 23 31 34 35 12 26 46 41 33 41 23 35 16 41 33 34 35 12 26 32 33 13 33 16 34 35 12 26 23 34 35 12 26 42 32 23 25 11 26 55 32 33 16 41 46 25 34 35 12 26 36 41 35 16 31 26 16 32 23 16 34 35 12 26 25 34 35 12 26 35 33 36 41 42 34 35 12 26 23 34 35 12 26 36 11 31 33 35 11 22 13 23 41 23 61 34 35 12 26 13 15 33 44 32 33 13 26 62 16 41 34 35 12 26 32 11 36 34 35 12 26 32 11 34 35 12 26 34 33 23 36 25 34 35 12 26 32 33 13 54 44 34 35 12 26 14 33 35 23 22 33 32 41 33 13 41 46 25 34 35 12 26 13 11 21 32 33 34 35 12 26 34 33 31 32 23 41 55 22 34 41 34 35 12 26 46 41 33 34 35 12 26 25 11 21 15 54 24 34 35 12 26 51 11 14 34 35 12 26 13 34 16 35 16 15 34 35 12 26 34 35 23 32 33 36 23 41 34 35 12 26 36 34 35 12 26 36 33 12 33 24 34 35 12 26 42 35 33 25 23 34 35 12 26 23 34 35 12 26 33 34 54 41 41 46 25 34 35 13 36 41 34 35 13 36 41 35 11 22 32 33 33 12 35 11 22 23 16 34 35 12 26 25 42 26 55 41 42 35 22 34 41 34 35 12 26 62 22 54 25 33 13 34 35 12 26 23 34 35 12 26 41 35 11 15 23 45 23 24 34 35 12 26 15 16 26 11 16 41 34 35 12 26 32 11 51 34 35 12 26 31 23 35 34 35 12 26 42 15 23 13 23 41 16 26 55 32 33 34 35 12 26 12 33 14 11 41 54 31 41 46 25 34 35 12 26 33 12 52 16 32 23 16 34 35 12 26 36 34 35 12 26 26 61 15 55 31 23 34 35 12 26 35 11 22 32 54 44 34 35 12 26 32 11 45 23 33 32 11 26 55 32 33 36 41 16 24 34 35 12 26 35 11 36 51 23 35 62 16 41 34 35 12 26 25 35 42 14 33 22 33 35 22 34 41 34 35 12 26 34 33 22 13 33 26 62 34 35 12 26 32 11 31 34 35 12 26 34 33 32 23 31 11 41 55 34 35 12 26 23 34 35 12 26 42 13 11 21 11 41 55 34 35 12 26 15 35 42 14 34 35 12 26 15 35 42 14 11 41 46 25 34 35 12 26 13 22 11 23 31 32 33 16 34 35 12 26 42 13 11 21 16 32 23 16 34 35 12 26 23 34 35 12 26 41 16 35 34 23 31 33 36 41 55 34 35 12 26 36 33 22 15 11 61 41 34 35 12 26 33 36 32 33 13 42 34 35 12 26 15 26 62 34 35 12 26 14 11 35 31 33 32 23 46 32 33 14 33 34 35 12 26 36 33 36 42 52 16 36 41 13 33 13 11 32 23 62 41 46 25 34 35 13 36 41 34 35 13 36 41 34 35 23 35 33 15 11 34 35 12 26 41 33 21 16 34 35 12 26 23 14 35 11 16 41 34 35 12 26 13 11 21 32 42 61 34 35 17

3

Введите ключевое слово для шифра Белазо или Плейфера

Введите ключевую букву для шифра Виженера

Введите ключевую матрицу для шифра Матричный

Выберите режим:

Расшифрование

Выполнить

☒ Расширенный текст