

## Метод 1: Маршрутное перестановление

---

```
function route_transposition(text::String, rows::Int, cols::Int, key::String) # Дополняем текст пробелами до
    # нужной длины required_length = rows * cols if length(text) < required_length text = text * repeat(" ",
    required_length - length(text)) elseif length(text) > required_length error("Текст слишком длинный для
    заданных размеров таблицы.") end
```

```
# Заполняем таблицу символами текста
table = reshape(collect(text), (rows, cols))

# Сортируем столбцы по ключу
sorted_indices = sortperm(collect(key))
encrypted_table = table[:, sorted_indices]

# Читаем текст по строкам
return join(encrypted_table[:])
```

end

## Метод 2: Шифрование с помощью решеток

---

```
function grille_cipher(text::String, k::Int) # Создаем пустую матрицу размером 2k x 2k grid = fill(' ', (2k, 2k)) n
    = length(text)
```

```
# Заполняем решетку символами текста
idx = 1
for i in 1:k, j in 1:k
    if idx > n
        break
    end
    grid[i, j] = text[idx]
    idx += 1
end

# Поворачиваем решетку 3 раза и записываем символы
for _ in 1:3
    grid = rotl90(grid)
    for i in 1:k, j in 1:k
        if idx > n
            break
        end
        grid[i, j] = text[idx]
        idx += 1
    end
end
```

```
# Читаем зашифрованный текст
return join(grid[:])
```

end

## Метод 3: Таблица Виженера

---

```
function vigenere_encrypt(text::String, key::String) alphabet =
"АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ" key = uppercase(key) text = uppercase(text)
```

```
# Преобразуем текст и алфавит в массивы символов
text_chars = collect(text)
alphabet_chars = collect(alphabet)

# Генерируем повторяющийся ключ
repeated_key = repeat(collect(key), div(length(text), length(key)) + 1)
[1:length(text)]

encrypted_text = ""
for (t_char, k_char) in zip(text_chars, repeated_key)
    if t_char in alphabet_chars
        # Находим индексы символов в алфавите
        t_idx = findfirst(==(t_char), alphabet_chars)
        k_idx = findfirst(==(k_char), alphabet_chars)

        # Вычисляем новый символ
        new_idx = mod(t_idx + k_idx - 2, length(alphabet_chars)) + 1
        push!(encrypted_text, alphabet_chars[new_idx])
    else
        push!(encrypted_text, t_char)
    end
end

return encrypted_text
```

end

## Главная программа для выбора метода

---

```
function main() println("Выберите метод шифрования:") println("1. Маршрутное перестановление")
println("2. Шифрование с помощью решеток") println("3. Таблица Виженера") print("Ваш выбор (1, 2 или 3): ")
```

```
choice = readline()

if choice == "1"
    println("\n=== Маршрутное перестановление ===")
    text = "НЕБЗАНЕОПЕЧАТАМНИК"
    rows, cols = 5, 6
    key = "пароль"
    println("Исходный текст: ", text)
    println("Зашифрованный текст: ", route_transposition(text, rows, cols, key))

elseif choice == "2"
    println("\n=== Шифрование с помощью решеток ===")
    text = "ШИФРРЕШЕТКА"
    k = 2
    println("Исходный текст: ", text)
    println("Зашифрованный текст: ", grille_cipher(text, k))

elseif choice == "3"
    println("\n=== Таблица Виженера ===")
    text = "КРИПТОГРАФИЯСЕРЬЕЗНАЯНАУКА"
    key = "МАТЕМАТИКА"
    println("Исходный текст: ", text)
    println("Зашифрованный текст: ", vigenere_encrypt(text, key))

else
    println("Неверный выбор. Пожалуйста, выберите 1, 2 или 3.")
end
```

end

## Запуск главной программы

---

main()