

```
def route_cipher(text, password):
    n = len(password)
    m = (len(text) + n - 1) // n # Вычисляем количество строк
    # Дополняем текст, если необходимо
    text += ' ' * (m * n - len(text))
    # Записываем текст в таблицу
    table = [text[i * n:(i + 1) * n] for i in range(m)]
    # Сортируем столбцы по алфавитному порядку пароля
    sorted_columns = sorted(range(n), key=lambda x: password[x])
    # Собираем криптограмму
    ciphertext = ''.join([table[i][j] for j in sorted_columns for i in range(m)])
    return ciphertext
```

```
# Пример использования
text = "нельзя недооценивать противника"
password = "пароль"
ciphertext = route_cipher(text, password)
print("Криптограмма:", ciphertext)
```



Криптограмма: енеьи зоври ьдипн н цттален в яоаок

```
def create_grid(k):
    grid = [[(i * k + j + 1) for j in range(k)] for i in range(k)]
    return grid
```

```
def rotate_grid(grid):
    return [list(row) for row in zip(*grid[::-1])]
```

```
def grid_cipher(text, password):
    k = int(len(password) ** 0.5)
    grid = create_grid(k)
    # Создаем большую решетку
    big_grid = [[0] * (2 * k) for _ in range(2 * k)]
    for _ in range(4):
        for i in range(k):
            for j in range(k):
                big_grid[i][j] = grid[i][j]
        grid = rotate_grid(grid)
    # Заполняем разрезы текстом
    text_index = 0
    for i in range(2 * k):
        for j in range(2 * k):
            if big_grid[i][j] != 0:
                big_grid[i][j] = text[text_index]
                text_index += 1
    # Собираем криптограмму
    ciphertext = ''.join([big_grid[i][j] for j in range(2 * k) for i in range(2 * k)])
    return ciphertext
```

```
# Пример использования
text = "договор подписали"
password = "шифр"
ciphertext = grid_cipher(text, password)
```

```
print("Криптограмма:", ciphertext)
```



Криптограмма: дгоо

```
def vigenere_cipher(text, key):  
    alphabet = 'абвгдеёжзийклмнопрстуфхцчщъыьэюя'  
    key_length = len(key)  
    key_as_indices = [alphabet.index(k) for k in key]  
    ciphertext = ''  
    for i, char in enumerate(text):  
        if char in alphabet:  
            shift = key_as_indices[i % key_length]  
            original_index = alphabet.index(char)  
            new_index = (original_index + shift) % len(alphabet)  
            ciphertext += alphabet[new_index]  
        else:  
            ciphertext += char  
    return ciphertext
```

```
# Пример использования  
text = "криптография серьезная наука"  
key = "математика"  
ciphertext = vigenere_cipher(text, key)  
print("Криптограмма:", ciphertext)
```



Криптограмма: чрыфяохщкфхя цсронтнмя тмуэи