Задание 1. Маршрутное шифрование

```
import numpy as np

ALPHABET = 'абвгдеёжзийклмнопрстуфхцчшщъыьэюя'

m, n = int(input()), int(input())  # вводим числа
# m, n = 4, 5
if m < 1 or n < 1:  # проверка на подходящее число
    raise(ValueError('m and n must be more then 1!'))

# ввод сообщения
text = input().replace(' ', '').lower().replace('ё', 'е')
# text = 'я хорошо покушал с утра'.lower().replace(' ', '').replace('ё', 'е')

if all(map(lambda x: x not in ALPHABET + ' ', text)):  # проверка на подходящий текст
    raise(ValueError('Text has not acceptable symbols!'))

# дополнение текста случайными буквами, если не хватает символов до полной таблицы
if m * n != len(text):
    text += ''.join([np.random.choice(list(ALPHABET)) for _ in range(m * n - len(text))])

# введение пароля
password = input().replace(' ', '').lower().replace('ё', 'е')
# password = 'карл'.lower().replace(' ', '').replace('ё', 'е')

if all(map(lambda x: x not in ALPHABET + ' ', password)):  # проверка пароля на допустимые символы
    raise(ValueError('Password has not acceptable symbols!'))
if len(password) != m:  # проверка длинны пароля
    raise(ValueError('Password len must be equall to matrix size!'))

# шифрование
''.join(
    np.array(list(text)).reshape((m, n))[
        [password.index(s) for s in sorted(password)]  # определения порядка следования столбцов символов
    ].reshape((n * m))
)
```

'шопокяхороутраэушалс'

Задание 2. Шифрование с помощью решеток

```
def nrot90(a: np.array, n: int) -> np.array:
    """Функция для поворота 2d массива с заданным числом поворотов

    Args:
        a (np.array): _description_
        n (int): _description_

    Returns:
        np.array: _description_
    """
    for _ in range(n):
        a = np.rot90(a)
    return a

k = int(input())  # ввод размера решетки

text = input().replace(' ', '').lower().replace('ё', 'е')  # ввод текста
if all(map(lambda x: x not in ALPHABET + ' ', text)):  # проверка на подходящий
текст
    raise(ValueError('Text has not acceptable symbols!'))

if 4 * k**2 > len(text):  # проверка на размер текста
    text += ''.join([np.random.choice(list(ALPHABET)) for _ in range(4 * k**2 -
len(text))])

# введение пароля
password = input().replace(' ', '').lower().replace('ё', 'е')

if all(map(lambda x: x not in ALPHABET + ' ', password)):  # проверка пароля на
допустимые символы
    raise(ValueError('Password has not acceptable symbols!'))
if len(password) != 2 * k:  # проверка длинны пароля
    raise(ValueError('Password len must be equall to matrix size!'))

print(f'Размер решетки:\t{k}')
print(f'Сообщение для шифрования:\t{text}')
print(f'Пароль:\t{password}')

# формирование матрицы из текста
text = np.array(list(text)).reshape((2 * k, 2 * k))
################################################################################

# формирование квадрата 2k x 2k
a = np.arange(1, k**2 + 1).reshape((k, k))
board = np.concatenate([
    np.concatenate([a, nrot90(a, 3)], axis=1),
    np.concatenate([nrot90(a, 1), nrot90(a, 2)], axis=1)
])
print('\nИзначальная решетка:')
print(board, end='\n\n')
```

```
# доставание всех индексов каждой цифры
d = {}
for i, y in enumerate(board):
    for j, x in enumerate(board[i]):
        if x in d:
            d[x].append((i, j))
        else:
            d[x] = [(i, j)]

# случайное доставание индексов каждой цифры, получение шифра
indexes = []
board = board.astype(str)
for v in d.values():
    board[v[np.random.choice(np.arange(k))]] = '#'
    indexes.append(v[np.random.choice(np.arange(k))])

print('Получившееся "решето":')
for v in board:
    print(*v, sep='\t')

################################################################################

cyphered = np.full((2 * k, 2 * k), '')  # формирование пустой решетки для
заполнения шифрованным сообщение
indexes = sorted(indexes, key=lambda x: (x[0], x[1]))  # сортировка индексов для
последовательного заполнения
for i in range(4, 0 , -1):  # проход по крученным массивам
    text = nrot90(text, i)
    cyphered = nrot90(cyphered, i)
    for idx in indexes:  # внесение символов на каждом развороте
        cyphered[idx] = text[idx]

print('\nЗашифрованное сообщение:', end='\t')
# зашифрование паролем и приведение в текстовый вид
print(''.join(cyphered[[password.index(v) for v in sorted(password)]].reshape(4
* k**2)))
```

Размер решетки: 3
Сообщение для шифрования:
сегодняхорошокушалсутраивобедщецйкат
Пароль: пароль

Изначальная решетка:
[[1 2 3 7 4 1]
[4 5 6 8 5 2]
[7 8 9 9 6 3]

```
 [3 6 9 9 8 7]
 [2 5 8 6 5 4]
 [1 4 7 3 2 1]]
```

Получившееся "решето":

```
1   2   #   7   #   #
4   5   6   8   5   #
7   #   9   #   #   3
3   6   9   9   8   #
2   #   8   6   5   4
1   4   7   3   2   1
```

Зашифрованное сообщение:
щдебовшорохялашукотакйцеиартусндогес

Задание 3. Таблица Виженера

```
vtable = [list(ALPHABET[i:] + ALPHABET[:i]) for i in range(len(ALPHABET))]

print('Таблица Виженера:')
for r in vtable:
    print(*r, sep=' ')
```

```
Таблица Виженера:
а б в г д е ё 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
б в г д е ё 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 а
в г д е ё 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 а б
г д е ё 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 а б в
д е ё 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 а б в г
е ё 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 а б в г д
ё 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 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 а б в г д е ё
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 а б в г д е ё ж
и й к л м н о п р с т у ф х ц ч ш щ ъ ы ь э ю я а б в г д е ё ж з
й к л м н о п р с т у ф х ц ч ш щ ъ ы ь э ю я а б в г д е ё ж з и
к л м н о п р с т у ф х ц ч ш щ ъ ы ь э ю я а б в г д е ё ж з и й
л м н о п р с т у ф х ц ч ш щ ъ ы ь э ю я а б в г д е ё ж з и й к
м н о п р с т у ф х ц ч ш щ ъ ы ь э ю я а б в г д е ё ж з и й к л
н о п р с т у ф х ц ч ш щ ъ ы ь э ю я а б в г д е ё ж з и й к л м
о п р с т у ф х ц ч ш щ ъ ы ь э ю я а б в г д е ё ж з и й к л м н
п р с т у ф х ц ч ш щ ъ ы ь э ю я а б в г д е ё ж з и й к л м н о
р с т у ф х ц ч ш щ ъ ы ь э ю я а б в г д е ё ж з и й к л м н о п
с т у ф х ц ч ш щ ъ ы ь э ю я а б в г д е ё ж з и й к л м н о п р
```

```
т у ф х ц ч ш щ ъ ы ь э ю я а б в г д е ё ж з и й к л м н о п р с
у ф х ц ч ш щ ъ ы ь э ю я а б в г д е ё ж з и й к л м н о п р с т
ф х ц ч ш щ ъ ы ь э ю я а б в г д е ё ж з и й к л м н о п р с т у
х ц ч ш щ ъ ы ь э ю я а б в г д е ё ж з и й к л м н о п р с т у ф
ц ч ш щ ъ ы ь э ю я а б в г д е ё ж з и й к л м н о п р с т у ф х
ч ш щ ъ ы ь э ю я а б в г д е ё ж з и й к л м н о п р с т у ф х ц
ш щ ъ ы ь э ю я а б в г д е ё ж з и й к л м н о п р с т у ф х ц ч
щ ъ ы ь э ю я а б в г д е ё ж з и й к л м н о п р с т у ф х ц ч ш
ъ ы ь э ю я а б в г д е ё ж з и й к л м н о п р с т у ф х ц ч ш щ
ы ь э ю я а б в г д е ё ж з и й к л м н о п р с т у ф х ц ч ш щ ъ
ь э ю я а б в г д е ё ж з и й к л м н о п р с т у ф х ц ч ш щ ъ ы
э ю я а б в г д е ё ж з и й к л м н о п р с т у ф х ц ч ш щ ъ ы ь
ю я а б в г д е ё ж з и й к л м н о п р с т у ф х ц ч ш щ ъ ы ь э
я а б в г д е ё 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
```

```python
text = input().replace(' ', '').lower()
if all(map(lambda x: x not in ALPHABET + ' ', text)):  # проверка на подходящий текст
    raise(ValueError('Text has not acceptable symbols!'))

password = input().replace(' ', '').lower()  # введение пароля

if all(map(lambda x: x not in ALPHABET + ' ', password)):  # проверка пароля на допустимые символы
    raise(ValueError('Password has not acceptable symbols!'))
if len(password) > len(text):  # проверка, что длинна пароля не больше, чем самого текста
    raise(ValueError('The length of password must be less than text length'))
if len(password) != len(text):  # проверка длинны пароля
    password = password * (len(text) // len(password)) +\
        password[:len(text) - len(password * (len(text) // len(password)))]

print(f'Сообщение для шифрования:\t{text}')
print(f'Пароль:\t{password}')

################################################################################

print('Шифр:',
    ''.join([vtable[ALPHABET.index(p)][ALPHABET.index(t)] for p, t in zip(password, text)]))
```

Сообщение для шифрования: яоченьлюблюкотовисобак
Пароль: жирафжирафжирафжирафжи
Шифр: ёчзевгфобаеуятгисвохжу