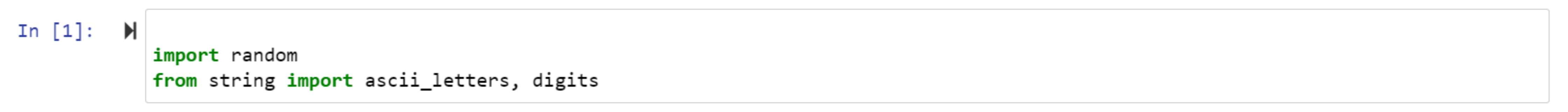
Презентация к лабораторной работе №7

Цель работы

Освоить на практике применение режима однократного гаммирования.

Выполнение работы

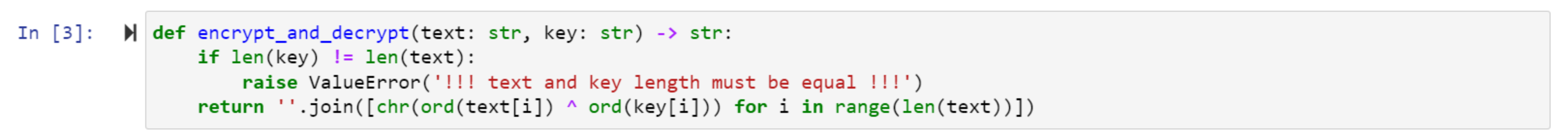
Импорт библиотек / функция для генерации случайного ключа



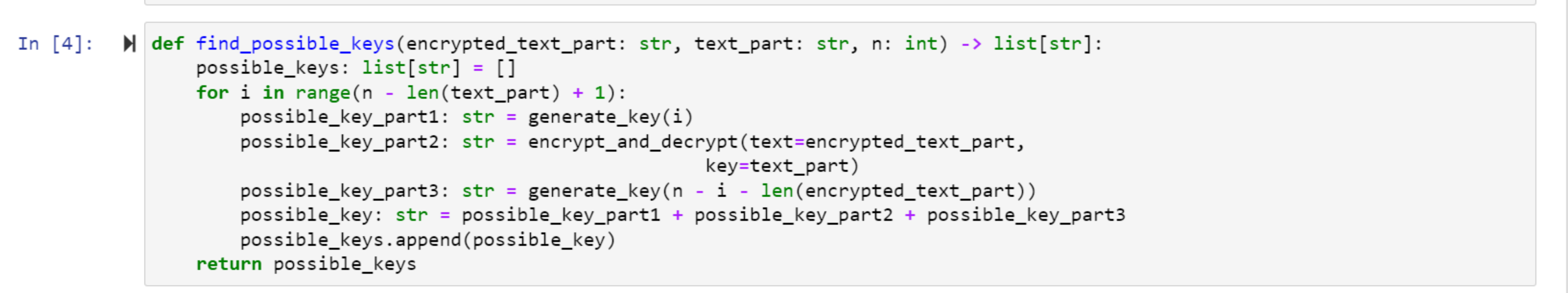
```python
import random
from string import ascii_letters, digits
```

```python
def generate_key(key_length: int) -> str:
    return ''.join([random.choice(ascii_letters + digits) for _ in range(key_length)])
```

Функция шифрования и дешифрования / функция find_possible_keys

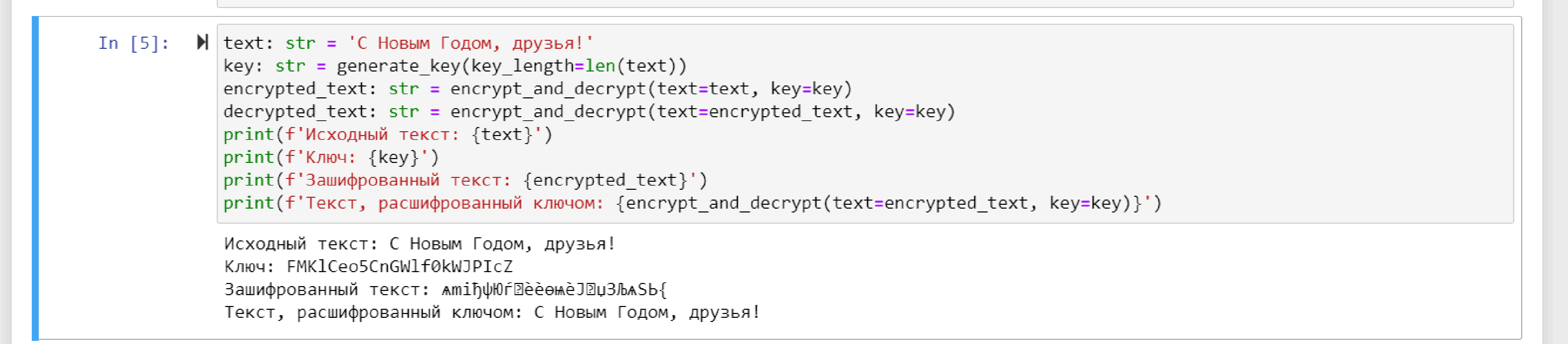


```python
def encrypt_and_decrypt(text: str, key: str) -> str:
    if len(key) != len(text):
        raise ValueError('!!! text and key length must be equal !!!')
    return ''.join([chr(ord(text[i]) ^ ord(key[i])) for i in range(len(text))])
```



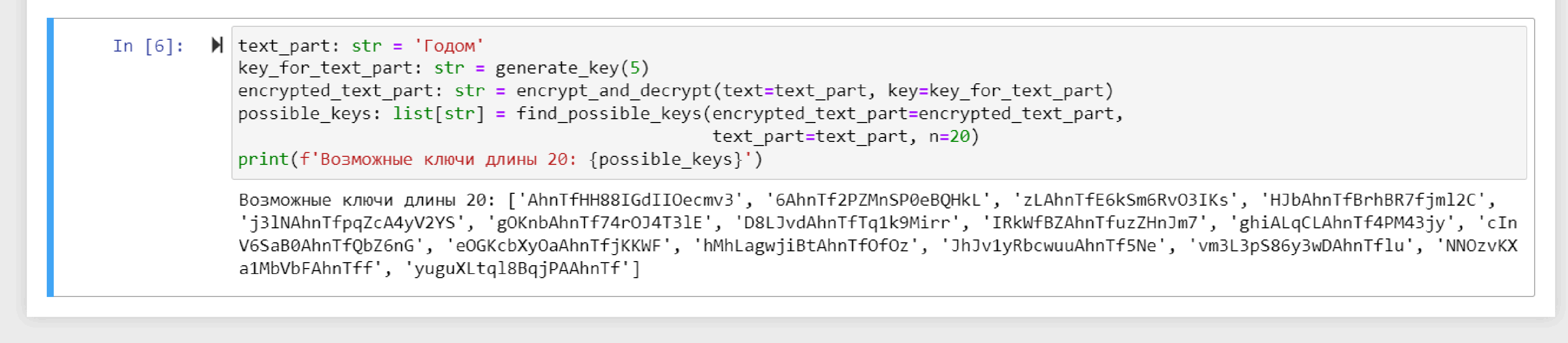
```python
def find_possible_keys(encrypted_text_part: str, text_part: str, n: int) -> list[str]:
    possible_keys: list[str] = []
    for i in range(n - len(text_part) + 1):
        possible_key_part1: str = generate_key(i)
        possible_key_part2: str = encrypt_and_decrypt(text=encrypted_text_part,
                                                      key=text_part)
        possible_key_part3: str = generate_key(n - i - len(encrypted_text_part))
        possible_key: str = possible_key_part1 + possible_key_part2 + possible_key_part3
        possible_keys.append(possible_key)
    return possible_keys
```

Проверка корректности работы функций



```
In [5]:
text: str = 'С Новым Годом, друзья!'
key: str = generate_key(key_length=len(text))
encrypted_text: str = encrypt_and_decrypt(text=text, key=key)
decrypted_text: str = encrypt_and_decrypt(text=encrypted_text, key=key)
print(f'Исходный текст: {text}')
print(f'Ключ: {key}')
print(f'Зашифрованный текст: {encrypted_text}')
print(f'Текст, расшифрованный ключом: {encrypt_and_decrypt(text=encrypted_text, key=key)}')
```

```
Исходный текст: С Новым Годом, друзья!
Ключ: FMKlCeo5CnGWlf0kWJPIcZ
Зашифрованный текст: ʍmiђψЮґ□èèѳӝèJ□џЗЉʍSЬ{
Текст, расшифрованный ключом: С Новым Годом, друзья!
```

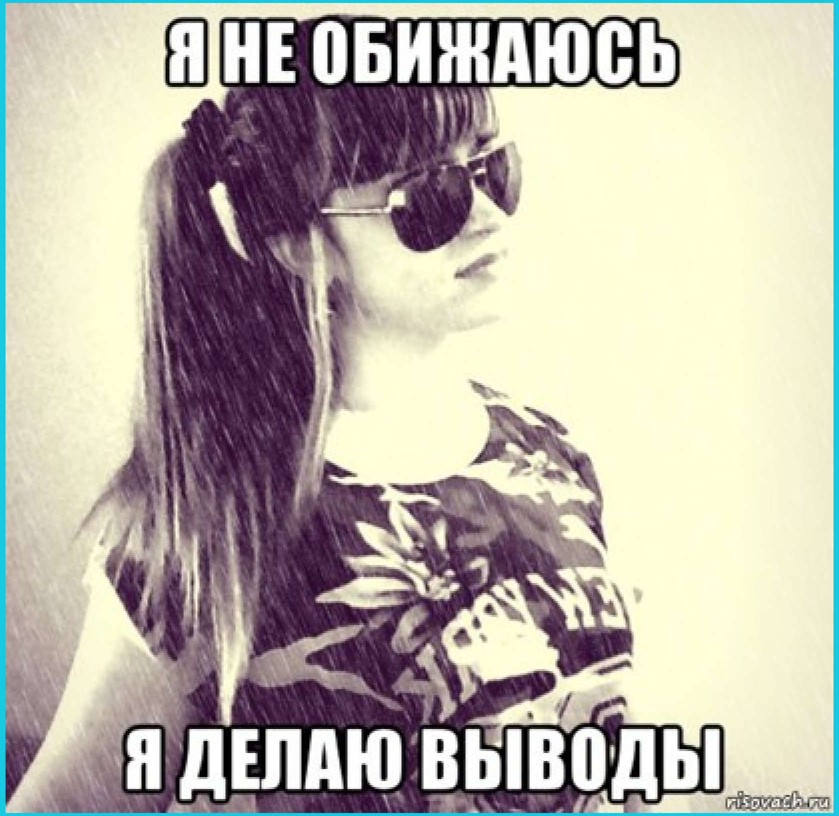


```
In [6]:
text_part: str = 'Годом'
key_for_text_part: str = generate_key(5)
encrypted_text_part: str = encrypt_and_decrypt(text=text_part, key=key_for_text_part)
possible_keys: list[str] = find_possible_keys(encrypted_text_part=encrypted_text_part,
                                              text_part=text_part, n=20)
print(f'Возможные ключи длины 20: {possible_keys}')
```

```
Возможные ключи длины 20: ['AhnTfHH88IGdIIOecmv3', '6AhnTf2PZMnSP0eBQHkL', 'zLAhnTfE6kSm6RvO3IKs', 'HJbAhnTfBrhBR7fjml2C',
'j3lNAhnTfpqZcA4yV2YS', 'gOKnbAhnTf74rOJ4T3lE', 'D8LJvdAhnTfTq1k9Mirr', 'IRkWfBZAhnTfuzZHnJm7', 'ghiALqCLAhnTf4PM43jy', 'cIn
V6SaB0AhnTfQbZ6nG', 'eOGKcbXyOaAhnTfjKKWF', 'hMhLagwjiBtAhnTfOfOz', 'JhJv1yRbcwuuAhnTf5Ne', 'vm3L3pS86y3wDAhnTflu', 'NNOzvKX
a1MbVbFAhnTff', 'yuguXLtql8BqjPAAhnTf']
```

Вывод

В рамках выполнения работы я освоил на практике применение режима однократного гаммирования.



Финал