

```
# Импорт библиотек
import random
import string
```

```
# Функция сложения двух строк по модулю
def xor_text_f(text1, text2):
    if len(text1) != len(text2): return "Ошибка: тексты разной длины"
    xor_text = ''
    for i in range(len(text1)):
        xor_text_symbol = ord(text1[i]) ^ ord(text2[i])
        xor_text += chr(xor_text_symbol)
    return xor_text
```

```
# Исходные тексты
P1 = "НаВашисходящийот1204"
P2 = "ВСеверныйфилиалБанка"
```

```
print("Исходный текст P1:", P1)
print("Исходный текст P2:", P2)
```

```
↵ Исходный текст P1: НаВашисходящийот1204
    Исходный текст P2: ВСеверныйфилиалБанка
```

```
# Создание ключа
random.seed(20)
key = ''.join(random.choice(string.ascii_letters + string.digits) for _ in range(len(P1)))
print("Ключ:", key)
```

```
↵ Ключ: 5URYX45jqR025g3uK5kb
```

```
# Шифрование P1 и P2 с помощью одного ключа
C1 = xor_text_f(P1, key)
C2 = xor_text_f(P2, key)
```

```
print("Зашифрованный текст C1:", C1)
print("Зашифрованный текст C2:", C2)
```

```
↵ Зашифрованный текст C1: ШѐрмАКѸЯяАёйуйэз[V
    Зашифрованный текст C2: ЧѸАжѸѸСшЖѸьйїїѸоJёђ
```

```
# Злоумышленник знает C1 и C2, но не знает ключ. Он может вычислить XOR двух текстов
P1_xor_P2 = xor_text_f(C1, C2)
print("Результат сложения исходных текстов P1 и P2:", P1_xor_P2)
```

```
↵ Результат сложения исходных текстов P1 и P2: 00'0}x|00pwr 0SÉЦЬЕ
```