

Варіант 8

Труднощі, які виникли: знову використовував неправильний порядок літер в алфавіті 🤪. Після виправлення труднощі закінчились. 😎

Завдання 2

перші 5 найчастіших біграм СТ

```
==== seq ngr count  ====
'жц' 65
'дэ' 62
'цэ' 60
'сц' 57
'оц' 55
```

Завдання 3-5

```
task_decrypt.py
**** 17, 94 ****
мальчизаулыбалисьисжаромвзялисьзаделоонирвализола
тистыецветыцветычтонаводняютьвесьмирпереплескивают
```

Ключ (17, 94) і шматок відкритого тексту

Розпізнавання відкритого тексту базується на порівнянні наперед відомих частот біграм (з 1ї лаби) і частоти в кандидаті на розшифрований текст.

```
# >= 0.007388 in mein kampf (>= 10k)
mc_bgrs: list[str] = ['ст', 'но', 'ен', 'то', 'на', 'ни', 'ос', 'ов', 'ро', 'пр', 'ра']
# <= 0.000010 in mein kampf ( <= 24 pcs )
lc_bgrs: list[str] = ['бж', 'жп', 'жс', 'фы', 'эс', 'гт', 'нж', 'нш', 'бш', 'жч', 'пв']
# 0 occurrences in mein kampf
forbidden_bgrs: list[str] = ['аы', 'аь', 'бй', 'вй', 'гй', 'гх', 'гщ', 'гы', 'гь', 'г
```

З відкритого тексту виділено 3 категорії біграм:

- Яких дуже багато: їх частка в тексті 1ї лаби ≥ 0.0074
- Яких дуже мало: в тексті 1ї лаби їх ≤ 0.00001
- Які в тексті першої лаби не зустрілись взагалі

Потім пробігаюсь по кожній групі і з деяким запасом на похибку рахую частку цих біграм в кандидаті.

Якщо більше 75% перевірених біграм у своїх групах, можна вважати, що кандидат розшифрований правильним ключем. Якщо алгоритм виведе декілька ключів як правильні, можна manually перевірити тексти на правльність, їх навряд чи буде більше кількох одиниць.