

Задание 1

Текстовый файл состоит из слов, взятых из текста условия к данному заданию. Слова расставлены в случайном порядке. Определите количество сочетаний слов, образующих предложения из условия задания. Знаки препинания и заглавность букв не учитывать.

Решение на PythonС комментариями:

```
# Считываем файл в строку
with open('24_1.txt') as file:
    string = file.readline()

# Все предложения из задания без знаков препинания и заглавных букв
arr = ['текстовый файл состоит из слов взятых из текста условия к данному заданию',
       'слова расставлены в случайном порядке',
       'определите количество сочетаний слов образующих предложения из условия задания',
       'знаки препинания и заглавность букв не учитывать']

answer = 0

# Подсчитываем сколько раз встречается 0, 1, 2 и 3 предложения в строке
# и добавляем к ответу
for i in range(len(arr)):
    answer += string.count(arr[i])

print(answer)
```

Без комментариев:

```
with open('24_1.txt') as file:
    string = file.readline()

arr = ['текстовый файл состоит из слов взятых из текста условия к данному заданию',
       'слова расставлены в случайном порядке',
       'определите количество сочетаний слов образующих предложения из условия задания',
       'знаки препинания и заглавность букв не учитывать']

answer = 0

for i in range(len(arr)):
    answer += string.count(arr[i])

print(answer)
```

Задание 2

Текстовый файл состоит из строк заглавных латинских букв.

Определите наибольший нечетный номер строки, в которой встречается комбинация DG*U, где звездочка (*) - любой символ из набора "VINODEL".

Гарантируется, что такая строка существует.

Примечание: нумерация строк начинается с 1.

Решение на Python

С комментариями:

```
# Заполним массив arr строками из файла
with open('24_2.txt') as file:
    arr = [line for line in file]

alph = 'VINODEL'
answer = 0

# Переберем каждую строку из массива
for i in range(len(arr)):

    # Число комбинаций в текущей строке
    currCount = 0

    # Перебираем символы из набора alph
    for j in alph:
        # .count('DG' + j + 'U') - подсчет количества
        # найденных комбинаций в строке
        if arr[i].count('DG' + j + 'U') > 0:
            currCount += 1

    # Если подходящие комбинации в строке есть (currCount > 0) и
    # номер строки не кратен 2,
    # то answer присваиваем i + 1, т.к. нумерация в Python
    # начинается с 0. Т.к. мы перебираем i по возрастанию,
    # answer будет наибольшим
    if currCount > 0:
        if (i + 1) % 2 != 0:
            answer = i + 1

print(answer)
```

Без комментариев:

```
with open('24_2.txt') as file:
    arr = [line for line in file]

alph = 'VINODEL'
answer = 0

for i in range(len(arr)):
    currCount = 0

    for j in alph:
        if arr[i].count('DG' + j + 'U') > 0:
            currCount += 1

    if currCount > 0:
        if (i + 1) % 2 != 0:
            answer = i + 1

print(answer)
```

Задание 3

Текстовый файл состоит из строк заглавных латинских букв. Найдите наибольшее расстояние между строчками, в которых есть сочетание EGE и нет сочетания VUZ.

Решение на Python

С комментариями:

```
# Заполним массив arr строками из файла
with open('24_3.txt') as file:
    arr = [line for line in file]

# maxLen минимальное, т.к. мы ищем макс. длину
# prevString максимальное, т.к. при вычитании индексов,
# мы ищем макс. длину
maxLen = float('-inf')
prevString = float('inf')

# Перебираем каждую строку массива
for i in range(len(arr)):

    # Если в строке есть EGE и нет VUZ,
    # то строчка подходит. Искомое расстояние -
    # разница между текущим индексом строки и предыдущим
    if (arr[i].count('EGE') > 0 and
        arr[i].count('VUZ') == 0):
        maxLen = max(maxLen, i - prevString)
        prevString = i

print(maxLen)
```

Без комментариев:

```
with open('24_3.txt') as file:
    arr = [line for line in file]

maxLen = float('-inf')
prevString = float('inf')

for i in range(len(arr)):
    if (arr[i].count('EGE') > 0 and
        arr[i].count('VUZ') == 0):
        maxLen = max(maxLen, i - prevString)
        prevString = i

print(maxLen)
```