

Решение заданий №24 ЕГЭ по информатике

Задача (Самая простая)

Текстовый файл состоит не более чем из 10^6 символов А, В и С.

Определите максимальное количество идущих подряд символов В. Для выполнения этого задания следует написать программу. Ниже приведён файл, который необходимо обработать с помощью данного алгоритма.

Решение:
на Python

```
f=open('24_1.txt')
s=f.read()
k=0
kmax=0

for i in range(0, len(s)):
    if s[i]=='B':
        k=k+1
        kmax=max(k, kmax)
    else:
        k=0

print(kmax)
```

- 1) С помощью команды **open()** подвязываемся к файлу. Чтобы не прописывать полный путь, файл должен лежать в той же папке, что и программа.
- 2) С помощью команды **.read()** зачитываем в переменную **s** всё содержимое файла.
- 3) Переменная **k** - это текущий счётчик символов "B". Задача переменной **kmax** сохранить максимальное значение **k**.
- 4) С помощью цикла **for** перебираем все символы из строки **s**. Переменная **i** пробегается по номерам всех символов. Счёт символов начинается в строке с нуля. В начале берём нулевой символ, потом первый и т.д. Конструкция *for i in range(0, len(s))* позволяет пройти по всем символам строки до конца.
- 5) Если нам встретился нужный символ "B", то мы счётчик прибавляем на 1. Это значит, что мы подсчитываем текущую цепочку. Если будет стоять 3 символа "B" подряд, значит, счёт **k** покажет значение 3. Как только встретится другой символ, то это означает, что цепочка прервалась и счётчик переводится в первоначальное положение 0.
- 6) При любом увеличении счётчика происходит анализ этого счётчика на максимальность.
- 7) Функция **max** выбирает максимальное значение из старого значения **kmax** и нового показания счётчика **k**. Если счётчику **k** удалось победить **kmax**, то его значение будет считаться максимальным на данный момент времени.
- 8) В ответ идёт значение **kmax**. Это и есть длина максимальной цепочки. В этой задаче ответ получается 11.

Ответ: 11

 C:\Users\User\AppData\Local\Programs\Python\Python39\python.exe

```
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>> f=open('24_1.txt')
>>> s=f.read()
>>> k=0
>>> kmax=0
>>>
>>> for i in range(0, len(s)):
...     if s[i]=='B':
...         k=k+1
...         kmax=max(k, kmax)
...     else:
...         k=0
...
>>> print(kmax)
11
>>> _
```

Закрепим это простое задание из **ЕГЭ по информатике 2023**.

Задача (Простая, закрепление)

Текстовый файл состоит не более чем из 10^6 символов X, Y и Z. Определите максимальное количество идущих подряд символов, среди которых нет символа Z. Для выполнения этого задания следует написать программу.

Если нет символов "Z", то подсчитываем, иначе сбрасываем.

Ответ: 34

```
f=open('24_2.txt')
```

```
s=f.read()
```

```
k=0
```

```
kmax=0
```

```
for i in range(0, len(s)):
```

```
    if s[i]!='Z':
```

```
        k=k+1
```

```
        kmax=max(k, kmax)
```

```
    else:
```

```
        k=0
```

```
print(kmax)
```

C:\Users\User\AppData\Local\Programs\Python\Python39\python.exe

```
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>> f=open('24_2.txt')
>>> s=f.read()
>>> k=0
>>> kmax=0
>>>
>>> for i in range(0, len(s)):
...     if s[i]!='Z':
...         k=k+1
...         kmax=max(k, kmax)
...     else:
...         k=0
...
>>> print(kmax)
34
>>> _
```

Задача (Обращаемся к соседу)

Текстовый файл состоит не более чем из 10^6 символов X, Y и Z. Определите максимальное количество идущих подряд символов, расположенных в алфавитном порядке (возможно с повторением символов). Для выполнения этого задания следует написать программу.

```
f=open('24_2.txt')
s=f.read()
k=1
kmax=0

for i in range(0, len(s)-1):
    if s[i]<=s[i+1]:
        k=k+1
        kmax=max(k, kmax)
    else:
        k=1

print(kmax)
```

- 1) Символы можно сравнивать между собой с помощью знаков ">" или меньше "<" (а так же ">=" или "<="), причём это сравнение происходит в алфавитном порядке.
- 2) Например, символ "В" будет больше, чем "А" и т.п. Теперь нам нужно уже обратится к соседу, уже важна связь между соседними символами. Раз мы обращаемся к следующему символу, то должны пробегать до $\text{len}(s)-1$, иначе куда мы будем обращаться, когда дойдём до последнего символа?
- 3) В самом начале цепочки мы анализируем два символа, а к счётчику прибавляем всего лишь 1. Чтобы это учесть, в начале счётчику (переменной **k**) присваиваем 1. Затем, по мере продвижения по цепочке, мы добавляем в наш анализ 1 новый символ, и к счётчику добавляется одна 1. Дальше уже таких проблем нет.
- 4) Раз мы присвоили в переменную **k** единицу, то и сбрасывать в ветке `else` мы тоже должны на 1.
- 5) Эти значения обычно взаимосвязаны.
- 6) Если мы обращаемся к соседнему символу, как правило, счётчик и сброс устанавливаются в 1. Это не стопроцентная истина, но иметь ввиду это нужно.

Ответ: 15



C:\Users\User\AppData\Local\Programs\Python\Python39\python.exe

```
...  
>>> print(kmax)  
34  
>>> f=open('24_2.txt')  
>>> s=f.read()  
>>> k=1  
>>> kmax=0  
>>>  
>>> for i in range(0, len(s)-1):  
...     if s[i]<=s[i+1]:  
...         k=k+1  
...         kmax=max(k, kmax)  
...     else:  
...         k=1  
...  
>>> print(kmax)  
15  
>>> _
```

Задача (Обращаемся к соседу, закрепление)

Текстовый файл состоит не более чем из 10^6 символов арабских цифр (0, 1, ..., 9). Определите максимальное количество идущих подряд цифр, среди которых каждые две соседние различны. Для выполнения этого задания следует написать программу.

Если соседи различны, мы подсчитываем, иначе сбрасываем.

Ответ: 120

```
f=open('24_3.txt')
```

```
s=f.read()
```

```
k=1
```

```
kmax=0
```

```
for i in range(0, len(s)-1):
```

```
    if s[i]!=s[i+1]:
```

```
        k=k+1
```

```
        kmax=max(k, kmax)
```

```
    else:
```

```
        k=1
```

```
print(kmax)
```

C:\Users\User\AppData\Local\Programs\Python\Python39\python.exe

```
34
>>> f=open('24_2.txt')
>>> s=f.read()
>>> k=1
>>> kmax=0
>>>
>>> for i in range(0, len(s)-1):
...     if s[i]<=s[i+1]:
...         k=k+1
...         kmax=max(k, kmax)
...     else:
...         k=1
...
>>> print(kmax)
15
>>> f=open('24_3.txt')
>>> s=f.read()
>>> k=1
>>> kmax=0
>>>
>>> for i in range(0, len(s)-1):
...     if s[i]!=s[i+1]:
...         k=k+1
...         kmax=max(k, kmax)
...     else:
...         k=1
...
>>> print(kmax)
120
>>>
```

Задача (Обращаемся к соседу, ещё сложнее)

Текстовый файл состоит не более чем из 10^6 символов арабских цифр (0, 1, ..., 9). Определите максимальное количество идущих подряд нечётных цифр, расположенных в неубывающем порядке. Для выполнения этого задания следует написать программу.

Здесь опять можно применить знаки сравнения прям к символам. Удобно проверить принадлежность к нечётным цифрам конструкцией `s[i] in '13579'`. Нечётной цифрой должна быть как текущая, так и следующая.

Ответ: 8

```
f=open('24_3.txt')
s=f.read()
k=1
kmax=0

for i in range(0, len(s)-1):
    if s[i]<=s[i+1] and s[i] in '13579' and s[i+1] in '13579' :
        k=k+1
        kmax=max(k, kmax)
    else:
        k=1

print(kmax)
```

C:\Users\User\AppData\Local\Programs\Python\Python39\python.exe

```
f=open('24_3.txt')
IndentationError: unexpected indent
>>> s=f.read()
>>> k=1
>>> kmax=0
>>>
>>> for i in range(0, len(s)-1):
...     if s[i]<=s[i+1] and s[i] in '13579' and s[i+1] in '13579' :
...         k=k+1
...         kmax=max(k, kmax)
...     else:
...         k=1
...
>>> print(kmax)
0
>>> f=open('24_3.txt')
>>> s=f.read()
>>> k=1
>>> kmax=0
>>>
>>> for i in range(0, len(s)-1):
...     if s[i]<=s[i+1] and s[i] in '13579' and s[i+1] in '13579' :
...         k=k+1
...         kmax=max(k, kmax)
...     else:
...         k=1
...
>>> print(kmax)
8
>>>
```