

Строки. Переборный алгоритм

Пример типового задания

Текстовый файл состоит из символов, обозначающих знаки «-», «*» и цифры 0, 7, 8, 9. Определите в прилагаемом файле максимальное количество идущих подряд символов, которые образуют математически правильную последовательность, в которую входят знаки « - » или « * » и натуральные числа без незначащих нулей.

Решение типового задания

- 1 Откроем файл, считаем строку и заменим все арифметические знаки на знак умножения, затем разделим строку по этому символу.
- 2 Создадим переменные `aritm` и `max_aritm` для хранения длины текущей и максимальной подстрок соответственно.
- 3 Запустим цикл для перебора всех подстрок. Внутри цикла сделаем проверку, что если длина текущей подстроки не равна нулю и первый символ не равен нулю, то прибавим к переменной текущую подстроку и знак умножения.
- 4 Иначе если длина текущей подстроки не равна нулю, но первый символ является нулем, а числовое значение подстроки не равно нулю, то прибавим к строке текущую подстроку без незначащих нулей и со знаком умножения.
- 5 Иначе приравняем строку к пустой и обновим длину максимальной строки.



Код:

```
f = open("24.1.txt")
s = f.readline()
s = s.replace("-", "*")
a = s.split("*")
arifm = ""
max_arifm = ""

for x in a:
    if len(x) != 0 and x[0] != "0":
        arifm += x + "*"
    elif len(x) != 0 and x[0] == "0" and int(x) != 0:
        arifm = str(int(x)) + "*"
    else:
        arifm = ""
    max_arifm = max(arifm, max_arifm, key=len)

print(len(max_arifm) - 1)
```



Заметки

