

«Решение сложных задач на обработку символьных строк».  
(примеры заданий сайт К.Поляков, ЕГЭ24)

**91. (П.Е. Финкель, г. Тимашевск)** Текстовый файл **24-1.txt** состоит не более чем из  $10^6$  символов. Определите самое большое число, состоящее только из нечётных цифр.

*Полный перебор*

```
s=open('24-1.txt').readline()

def f(s,p): #s-строка, p-номер текущего символа
    if s[p] not in '0123456789' and s[p+1] in '13579':
        st=''
        for i in range(p+1,len(s)):
            if s[i] in '13579':
                st+=s[i]
            elif s[i] not in '0123456789':
                return int(st)
            else:
                return 0
    return 0

mx=0
for p in range(len(s)-1):
    mx=max(mx,f(s,p))
print(mx)
```

*Эффективный алгоритм*

*Вариант1. Перебор с флагом - 1 проход по строке*

```
s=open('24-1.txt').readline()
st=''
stm=0
km=0
f_in=False
for i in range(len(s)):
    if not(f_in) and s[i] in '13579':
        f_in=True
    if f_in:
        if s[i] in '13579':
            st+=s[i]
            stm=max(int(st),stm)
        else:
            st=''
            f_in=False
print(stm)
```

*Вариант 2. Разбиение строки на подстроки + перебор подстрок в цикле с анализом каждой подстроки.*

```
#s = 'AB003571'
s = open("24-1.txt").read()
a = [c if c in '0123456789' else ' ' for c in s ]
s1 = "".join(a)

a = s1.split()
m = 0
for sub in a:
    if all(c in '13579' for c in sub):
        if int(sub) > m:
            m = int(sub)

print(m)
```

### Вариант 3 Используя регулярные выражения

```
from re import *
s=open('24-1.txt').readline()
print(max(map(str, findall(r'[13579]{1,}', s)), key=len))
```

## Использование словарей

**147.**Текстовый файл **24-s2.txt** содержит только заглавные буквы латинского алфавита (ABC...Z). Определите символ, который чаще всего встречается в файле сразу после буквы A. В ответе запишите сначала этот символ, а потом сразу (без разделителя) сколько раз он встретился после буквы A. Если таких символов несколько, нужно вывести тот, который стоит раньше в алфавите. Например, в тексте ABCAABADDD после буквы A два раза стоит B, по одному разу – A и D. Для этого текста ответом будет B2.

```
s=open('24-s2.txt').readline()
d={}
for i in range(len(s)-1):
    if s[i]=='A':
        d[s[i+1]]=d.get(s[i+1],0)+1
for c in d:
    if d[c]==max(d.values()):
        print(c,d[c])
```

### Многострочный файл

**255.**Текстовый файл **24-164.txt** состоит не более чем из  $10^6$  символов и содержит только буквы латинского алфавита. Текст разбит на строки различной длины. Для каждой строки нужно определить букву (или буквы), которая встречается в этой строке чаще всего после буквы X. Все эти буквы добавляются в новый список. Найдите букву, которая чаще всего встречается в построенном списке, и в качестве ответа укажите, сколько раз она там встретилась. Например, пусть файл содержит две строки:

```
ХАХВХАХВСХХ
ВХАХСХСХАХD
```

В первой строке чаще всего после буквы X встречаются буквы A и B (по 2 раза), а во второй строке – буквы A и C (по 2 раза). В итоге должен быть построен список [A, B, A, C], в котором чаще всего (2 раза) встречается буква A. Ответ: 2.

```
s=open('24-164.txt').readlines()
d={}
for x in s:
    dl = {}
    for i in range(len(x)-1):
        if x[i] == "X":
            dl[x[i + 1]] = dl.get(x[i + 1], 0) + 1
    mx = max(dl.values())
    res=[key for key, value in dl.items() if value==mx]
    print(res)

    for i in res:
        if i not in d: d[i]=1
        elif i in d: d[i]+=1
print(d)
print(max(d.values()))
```

**182.**Текстовый файл **24-181.txt** содержит строку из заглавных латинских букв и точек, всего не более чем из  $10^6$  символов. Определите максимальное количество идущих подряд символов, среди которых не более двух точек.

*Разбиение строки на подстроки + перебор подстрок в цикле с анализом каждой подстроки.*

```
s=open("24-181.txt").readline()
a=s.split('.')
print(a)
m=0
for i in range(len(a)-2):
    m=max(m, len(a[i])+len(a[i+1])+len(a[i+2])+2)
print(m)
```

**190.** Текстовый файл **24-181.txt** содержит строку из заглавных латинских букв и точек, всего не более чем из  $10^6$  символов. Определите максимальное количество идущих подряд символов, среди которых нет гласных букв (символов А, Е, I, О, U, Y), но есть не менее 6 точек.

*Разбиение строки на подстроки + перебор генератором*

```
s=open("24-181.txt").readline()
s1=''
for c in s:
    if c in 'AEIOUY':
        s1+=' '
    else:
        s1+=c
print(s1.split())
slen=[len(s) for s in s1.split() if s.count('.')>=6]
print(max(slen))
```

**219.**Текстовый файл **24-215.txt** содержит строку из символов А, В, С и цифр 1, 2, 3, всего не более чем  $10^6$  символов. Определите максимальное количество идущих подряд троек символов вида «цифра + буква + цифра».

*Полный перебор*

```
s=open('24-215.txt').readline()
def f(s,p):
    k=0
    for i in range(p, len(s)-2, 3):
        if s[i] in '123' and s[i+1] in 'ABC' and s[i+2] in '123':
            k+=1
        else: break
    return k
mx=0
for i in range(len(s)):
    mx=max(mx, f(s, i))
print(mx)
```

*Разбиение на подстроки+ перебор подстрок в цикле с анализом каждой подстроки.*

```
s=open('24-215.txt').readline()
s=s.replace('C', 'A').replace('B', 'A')
s=s.replace('2', '1').replace('3', '1')
t_max=0
for i in range(len(s)):
    z,t=i,0
    while s[z:z+3]=='1A1':
        t+=1
        z+=3
    if t>t_max:t_max=t
print(t_max)
```

*Эффективный алгоритм - 3 прохода по строке*

```
s=open('24-215.txt').readline()
k=km=0
for b in 0,1,2:
    for i in range(b,len(s)-2,3):
        if s[i] in '123' and s[i+1] in 'ABC' and s[i+2] in '123':
            k += 1
            km=max(km,k)
        else:
            k=0
print(km)
```

**265.(ЕГЭ-2023)** Текстовый файл **24-263.txt** состоит не более чем из  $10^6$  символов и содержит только заглавные буквы латинского алфавита. Определите максимальную длину подстроки, в которой символ Y встречается не более 150 раз.

*По индексу символов*

```
s='Y'+open('24-263.txt').readline()+'Y'
e=[]
for i in range(len(s)):
    if s[i]=='Y':
        e.append(i)
mx=0
for x in range(len(e)-151):
    mx=max(mx,e[x+151]-e[x]-1)
print(mx)
```

*Метод двух указателей*

```
s = open('24-263.txt').readline()
NY = 150
mx = 0
left = 0
for right in range(len(s)):
    while s[left:right+1].count('Y') > NY:
        left += 1
    mx = max(mx, right - left + 1)
print(mx)
```

**266.ЕГЭ-2023)** Текстовый файл **24-263.txt** состоит не более чем из  $10^6$  символов и содержит только заглавные буквы латинского алфавита. Определите минимальную длину подстроки, в которой символ Z встречается не менее 120 раз.

*Метод двух указателей*

```
s = open('24-263.txt').readline()
left = 0
mi = 10**10
for right in range(len(s)):
    while s[left:right+1].count('Z') == 120:
        mi = min(mi, right - left + 1)
        left += 1
print(mi)
```

**298\_ЕГЭ24(ЕГЭ-2024)** Текстовый файл **24-298.txt** состоит не более чем из  $10^6$  символов и содержит только символы, обозначающие 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.

### Полный перебор

```
s = open("24-298.txt").read()
def f(s, p):
    k = 0
    if s[p] not in ['*', '-', '0'] :
        for i in range(p, len(s)):
            if s[i: i+2] in ['**', '-*', '*-', '--', '*0', '-0']:
                break
            else:
                k += 1
    return k

m = 0
for p in range(len(s)):
    k = f(s, p)
    if k > m:
        m = k
        ans = s[p:p+k]
print(m, ans)
```

*Эффективный алгоритм. Разбиение на подстроки+ перебор подстрок генератором*

```
s = open('24-298.txt').readline()
# s = '789*90***90'
s = s.replace('-', '*').replace('7', '1').replace('9', '1',).replace('8', '1')
while '***' in s: s = s.replace('***', '**')
while '*0' in s: s = s.replace('*0', '* ')
s = s.replace('**', ' ')
a = [sub.strip('*') for sub in s.split()]
print(len(max(a, key = len)))
```

### Через регулярные выражения

```
from re import *
s = open("24-298.txt").read()
nat=r'(?:[789][0789]{0,})'
# -123*123-123*123
# число (знак число)*
res=fr'{nat}(?:[-*]{nat})*'
print(max(map(len, findall(res, s))))
```

**299. (ЕГЭ-2024)** Текстовый файл **24-299.txt** состоит не более чем из  $10^6$  символов и содержит только десятичные цифры, а также знаки «+» и «\*» (сложения и умножения). Определите максимальное количество символов в непрерывной последовательности, являющейся корректным арифметическим выражением с целыми неотрицательными числами (без знака), значение которого равно нулю. В этом выражении никакие два знака арифметических операций не стоят рядом, порядок действий определяется по правилам математики. В записи чисел отсутствуют незначащие (ведущие) нули. В ответе укажите количество символов в найденном выражении.

### Полный перебор

```
s = open("24-299.txt").read()
def f(s, p):
    k = 0
    if s[p] not in ['*', '+']:
        for i in range(p, len(s)):
            if s[i: i+2] in ['**', '++', '*+', '+*']:
                break
    return k
```

```

        else:
            try:
                if eval(s[p:i+1]) == 0:
                    k = i - p + 1
            except:
                pass
    return k

m = 0
print(len(s))
for p in range(len(s)):
    if p%10000 == 0: print("...", p)
    k = f(s, p)
    if k > m:
        m = k
        ans = s[p:p+k]
print(m, ans)

```

*Эффективный алгоритм. Разбиение на подстроки + перебор подстрок генератором*

```

s = open("24-299.txt").read()
#s = '*11+0*11+++0+1*0*11*111+1+1111111111*0'
while '***' in s: s = s.replace('***','**')
while '+++' in s: s = s.replace('+++','++')
s = s.replace('*', ' ').replace('++', ' ').replace('+++', ' ').replace('+++', ' ')
print(s)
a = [sub.strip('*').strip('+') for sub in s.split()]
print(a)
m = 0
for sub in a: #sub содержит сумму произведений, нас интересует самая длинная
    сумма, состоящая из нулевых слагаемых
        k = -1
        for p in sub.split('+'):
            if p == '0' or '0*' in p or '*0' in p:
                k += len(p) + 1
                m = max(m, k)
                print(p)
            else:
                k = -1

print(m)

```

*Через регулярные выражения*

```

from re import *
s = open("24-299.txt").read()
n=r'(?:[1-9][0-9]*|0)'
# число * число * ....* 0 * число....*число
pr0=fr'(?:{n}\*)*0(?:\{n}\*)*'
# pr0 + pr0 + ....+pr0
res=fr'{pr0}(?:\{pr0}\*)*'
print(max(map(len, findall(res,s))))

```