

Задание 24

Обработка строк

Чтение из файла (Python)

#однострочный файл

```
f = open("24.txt")
```

```
s = f.readline()
```

```
...
```

```
f.close()
```

#многострочный файл

```
f = open("24.txt")
```

```
for s in f:
```

```
...
```

```
f.close()
```

Чтение из файла (Pascal)

```
//однострочный файл
```

```
var f:text;
```

```
    s:string;
```

```
begin
```

```
    assign(f, '24.txt');
```

```
    reset(f);
```

```
    readln(f, s);
```

```
    ...
```

```
    close(f);
```

```
end.
```

Чтение из файла (Pascal)

```
//многострочный файл
```

```
var f:text;
```

```
    s:string;
```

```
begin
```

```
    assign(f, '24.txt');
```

```
    reset(f);
```

```
    while not eof(f) do begin
```

```
        readln(f, s);
```

```
        ...
```

```
    end;
```

```
    close(f);
```

```
end.
```

Поиск максимальной подстроки

Задача 1а. В текстовом файле находится цепочка из символов латинского алфавита A, B, C длиной не более 10^6 символов. Найдите длину самой длинной подцепочки, состоящей из символов C.

Описание решения

Начальные значения:

`curr = 0` #длина текущей цепочки

`maxS = 0` #длина максимальной цепочки

Циклом проходим все символы в строке, для каждого символа проверяем:

- Если это символ `C`, то увеличиваем длину текущей цепочки на один, сравниваем её с длиной максимальной цепочки.
- Если это другой символ, то длина текущей цепочки сбрасывается к начальному значению.

```
var f : text;
    s : string;
    i, curr, maxS : integer;
begin
assign(f, "24.txt");
reset(f);
readln(f, s);
curr:=0; maxS:=0;
for i:=1 to length(s) do begin
    if s[i]="C" then begin
        curr:=curr+1;
        if curr>maxS then maxS:=curr;
    end
    else curr:=0;
end;
writeln(maxS);
close(f);
end.
```

```
f = open("24.txt")
s = f.readline()
curr, maxS = 0, 0
for i in range(len(s)):
    if s[i]=="C":
        curr+=1
        maxS = max(curr, maxS)
    else:
        curr = 0
print(maxS)
f.close()
```

Поиск максимальной подстроки

Задача 1б. В текстовом файле находится цепочка из символов латинского алфавита X, Y, Z длиной не более 10^6 символов. Найдите максимальное количество идущих подряд символов, среди которых нет символов Z.

Описание решения

Начальные значения:

`curr = 0` #длина текущей цепочки

`maxS = 0` #длина максимальной цепочки

Циклом проходим все символы в строке, для каждого символа проверяем:

- Если это не символ Z, то увеличиваем длину текущей цепочки на один, сравниваем её с длиной максимальной цепочки.
- Если это символ Z, то длина текущей цепочки сбрасывается к начальному значению.

```
var f : text;  
    s : string;  
    i, curr, maxS : integer;  
begin  
assign(f, "24.txt");  
reset(f);  
readln(f, s);  
curr:=0; maxS:=0;  
for i:=1 to length(s) do begin  
    if s[i]<>"Z" then begin  
        curr:=curr+1;  
        if curr>maxS then maxS:=curr;  
    end  
    else curr:=0;  
end;  
writeln(maxS);  
close(f);  
end.
```

```
f = open("24.txt")
s = f.readline()
curr, maxS = 0, 0
for i in range(len(s)):
    if s[i]!="Z":
        curr+=1
        maxS = max(curr, maxS)
    else:
        curr = 0
print(maxS)
f.close()
```

Поиск максимальной подстроки

Задача 2. Текстовый файл состоит не более чем из 10^6 символов X, Y и Z. Определите максимальное количество идущих подряд символов, среди которых каждые два соседних различны.

Описание решения

Начальные значения:

В качестве начального эталона берём первый символ!

`curr = 1` #длина текущей цепочки

`maxS = 1` #длина максимальной цепочки

Циклом проходим все символы в строке, начиная со второго, для каждого символа проверяем:

- Если это символ отличается от предыдущего, то увеличиваем длину текущей цепочки на один, сравниваем её с длиной максимальной цепочки.
- Если символы совпадают, то длина текущей цепочки сбрасывается к начальному значению.

```
var f : text;
    s : string;
    i, curr, maxS : integer;
begin
assign(f, "24.txt");
reset(f);
readln(f, s);
curr:=1; maxS:=1;
for i:=2 to length(s) do begin
    if s[i]<>s[i-1] then begin
        curr:=curr+1;
        if curr>maxS then maxS:=curr;
    end
    else curr:=1;
end;
writeln(maxS);
close(f);
end.
```

```
f = open("24.txt")
s = f.readline()
curr, maxS = 1, 1
for i in range(1, len(s)):
    if s[i] != s[i-1]:
        curr += 1
        maxS = max(curr, maxS)
    else:
        curr = 1
print(maxS)
f.close()
```

Поиск максимальной подстроки

Задача 3. (А.М. Кабанов) В текстовом файле находится цепочка из символов латинского алфавита A, B, C, D, E. Найдите максимальную длину цепочки вида EABEABEABE... (составленной из фрагментов EAB, последний фрагмент может быть неполным).

Описание решения

Паттерн (шаблон) строки

Буква E подходит, если текущая длина строки 0, 3, 6...

Буква A подходит, если текущая длина строки 1, 4, 7...

Буква B подходит, если текущая длина строки 2, 5, 8...

Видно, что каждому символу соответствует определённый остаток от деления на 3.

СИМВОЛ	E	A	B	E	A	B	E	A
Count	0	1	2	3	4	5	6	7
Count%3	0	1	2	0	1	2	0	1

Описание решения

Начальные значения:

`curr` = 0 #длина текущей цепочки

`maxS` = 0 #длина максимальной цепочки

Циклом проходим все символы в строке, для каждого символа проверяем:

- Если это символ подходит под паттерн, то увеличиваем длину текущей цепочки на один, сравниваем её с длиной максимальной цепочки.
- Если символ не подходит, но это буква E, то длина текущей цепочки равна 1.
- В противном случае длина текущей цепочки сбрасывается к начальному значению.

```
var f : text;
    s : string;
    i, curr, maxS : integer;
begin
assign(f, "24.txt");
reset(f);
readln(f, s);
curr:=0; maxS:=0;
for i:=1 to length(s) do begin
    if (s[i]="E") and (curr mod 3 = 0) or
        (s[i]="A") and (curr mod 3 = 1) or
        (s[i]="B") and (curr mod 3 = 2) then begin
        curr:=curr+1;
        if curr>maxS then maxS:=curr;
    end
    else if s[i]="E" then curr:=1
    else curr:=0;
end;
writeln(maxS);
close(f);
end.
```

```
f = open("24.txt")
s = f.readline()
curr, maxS = 0, 0
for i in range(len(s)):
    if (s[i]=="E" and curr%3==0) or
        (s[i]=="A" and curr%3==1) or
        (s[i]=="B" and curr%3==2):
        curr+=1
        maxS = max(curr, maxS)
    elif s[i]=="E":
        curr = 1
    else:
        curr = 0
print(maxS)
f.close()
```

Поиск максимальной подстроки

Задача 4. В текстовом файле находится цепочка из не более чем 10^6 символов, в которую могут входить заглавные буквы латинского алфавита A...Z и десятичные цифры. Найдите длину самой длинной подцепочки, состоящей из одинаковых символов. Выведите сначала символ, из которого строится цепочка, а затем через пробел – длину этой цепочки. Если таких цепочек (максимальной длины) несколько, выведите информацию о первой встретившейся цепочке.

Описание решения

Начальные значения:

В качестве начального эталона берём первый символ!

`curr = 1` #длина текущей подстроки

`maxS = 1` #длина максимальной подстроки

`maxch = s[0]` #символ максимальной подстроки

Циклом проходим все символы в строке, начиная со второго, для каждого символа проверяем:

- Если это символ совпадает с предыдущим, то увеличиваем длину текущей цепочки на один, сравниваем её с длиной максимальной цепочки.
- Если текущая цепочка самая большая, то сохраняется и её длина и символ из которого она состоит.
- Если символы совпадают, то длина текущей цепочки сбрасывается к начальному значению.

```
var f : text;  
    s : string;  
    i, curr, maxS : integer;  
    maxch : char;  
  
begin  
assign(f, "24.txt");  
reset(f);  
readln(f, s);  
curr:=1; maxS:=1; maxch:=s[1];  
for i:=2 to length(s) do begin  
    if s[i]=s[i-1] then begin  
        curr:=curr+1;  
        if curr>maxS then begin  
            maxS:=curr;  
            maxch:=s[i];  
        end;  
    end  
    else curr:=1;  
end;  
writeln(maxch, maxS);  
close(f);  
end.
```

```
f = open("24.txt")
s = f.readline()
curr, maxS, maxch = 1, 1, s[0]
for i in range(1, len(s)):
    if s[i]==s[i-1]:
        curr+=1
        if curr>maxS:
            maxS = curr
            maxch = s[i]
    else:
        curr = 1
print(maxch, maxS)
f.close()
```

Поиск максимальной подстроки

Задача 5. (П.Е. Финкель) Текстовый файл состоит не более чем из 10^6 символов - заглавных латинских букв и цифр. Числом назовём непрерывную последовательность цифр, слева и справа от которой строят буквы. Определите максимальное нечётное число, записанное в этом файле.

Описание решения

В данной задаче цепочки необходимо проверять только в конце!

Для корректной проверки рекомендуется в конец строки добавить неподходящий символ!

Начальные значения:

`curr = ''` #текущая подстрока

`n = 0` #значение очередного числа из строки

`maxN = 0` #значение максимального нечётного числа

Циклом проходим все символы в строке, для каждого символа проверяем:

- Если символ относится к цифрам, то прибавляем его к текущей цепочке.
- Если символ относится к буквам, то числовое значение цепочки (если она существует) сравниваем с максимальным нечётным числом, затем текущая цепочка становится пустой.

```
var f : text;
    s, curr : string;
    i, n, maxN : integer;
begin
assign(f, '24.txt');
reset(f);
readln(f, s);
s:=s+'A';
curr:=''; maxN:=0;
for i:=1 to length(s) do begin
    if ('0'<=s[i]) and (s[i]<='9') then curr:=curr+s[i];
    else if curr<>' ' then begin
        n:= strtoint(curr);
        if (n mod 2 <> 0) and (n>maxN) then maxN:=n;
        curr:=' ';
    end;
end;
writeln(maxN);
close(f);
end.
```

```
f = open('24.txt')
s = f.readline() + 'A'
curr, maxN = '', 0
for i in range(len(s)):
    if '0' <= s[i] <= '9':
        curr = curr + s[i]
    elif curr != '':
        n = int(curr)
        if n % 2 != 0 and n > maxN: maxN = n
        curr = ''
print(maxN);
f.close()
```

Поиск количества подстрок

Задача 6. (Е. Джобс) Текстовый файл состоит не более чем из 10^6 символов S, T, O, C, K. Сколько раз встречается в файле комбинация «КОТ»?

Описание решения

Начальные значения:

$k = 0$ #количество комбинаций КОТ

Циклом проходим все цепочки с длиной 3. Примем, что переменная i будет хранить номер первого элемента в тройке, то есть, будем рассматривать тройки $(s[i], s[i+1], s[i+2])$.

Если цепочка подходит под условие, то переменная количества увеличивается на единицу.

```
var f : text;  
    s : string;  
    i, k : integer;  
begin  
assign(f, "24.txt");  
reset(f);  
readln(f, s);  
k:=0;  
for i:=1 to length(s)-2 do  
    if s[i]+s[i+1]+s[i+2]="KOT" then  
        k:=k+1;  
writeln(k);  
close(f);  
end.
```

```
f = open("24.txt")
s = f.readline()
k = 0
for i in range(len(s)-2):
    if s[i]+s[i+1]+s[i+2]=='КОТ':
        k+=1
print(k)
f.close()
```

Поиск количества подстрок

Задача 7. (Е. Джобс) Текстовый файл состоит не более чем из 10^6 символов S, T, O, C, K. Сколько раз встречается комбинация «ОСК», не являющаяся при этом частью комбинации «STOCK».

Описание решения

Начальные значения:

$k = 0$ #количество комбинаций ОСК

$k1 = 0$ #количество комбинаций STOCK

Первым циклом проходим все цепочки с длиной 3. В цикле считаем количество комбинаций ОСК.

Вторым циклом проходим все цепочки с длиной 5. В цикле считаем количество комбинаций STOCK.

Ответом является разность этих значений.

```
var f : text;
    s : string;
    i, k, k1 : integer;
begin
assign(f, "24.txt");
reset(f);
readln(f, s);
k:=0; k1:=0;
for i:=1 to length(s)-2 do
    if s[i:i+3]="KOT" then
        k:=k+1;
for i:=1 to length(s)-4 do
    if s[i:i+5]="STOCK" then
        k1:=k1+1;
writeln(k-k1);
close(f);
end.
```

```
f = open("24.txt")
s = f.readline()
k, k1 = 0, 0
for i in range(len(s)-2):
    if s[i:i+3] == 'OCK':
        k += 1
for i in range(len(s)-4):
    if s[i:i+5] == 'STOCK':
        k1 += 1
print(k-k1)
f.close()
```

Поиск количества подстрок

Задача 8. Текстовый файл состоит не более чем из 10^6 заглавных латинских букв (A..Z). Текст разбит на строки различной длины. Определите количество строк, в которых буква J встречается чаще, чем буква E.

Описание решения

Начальные значения:

$k = 0$ #количество подходящих строк

$k_j = 0$ #количество букв J в конкретной строке

$k_e = 0$ #количество букв E в конкретной строке

Поочерёдно считываем строки из файла. Для каждой строки считаем количество букв J и E и если буква J встречается чаще E, то увеличиваем количество подходящих строк на 1.

```
var f : text;
    s : string;
    i, k, kj, ke : integer;
begin
assign(f, "24.txt");
reset(f);
k:=0;
while not eof(f) do begin
    readln(f, s);
    kj:=0; ke:=0;
    for i:=1 to length(s) do begin
        if s[i]="J" then kj:=kj+1;
        if s[i]="E" then ke:=ke+1;
    end;
    if kj>ke then k:=k+1;
end;
writeln(k);
close(f);
end.
```

```
f = open("24.txt")
k = 0
for s in f:
    ke, kj = 0, 0
    for i in range(len(s)):
        if s[i]=="E": ke+=1
        if s[i]=="J": kj+=1
    if kj>ke: k+=1
print(k)
f.close()
```

Поиск количества подстрок

Задача 9. Текстовый файл состоит не более чем из 10^6 заглавных латинских букв (A..Z). Текст разбит на строки различной длины. Определите количество строк, в которых встречается комбинация F*O, где звёздочка обозначает любой СИМВОЛ.

Описание решения

Начальные значения:

`k = 0` #количество подходящих строк

`good = False` #флаг наличия F*O

Поочерёдно считываем строки из файла. Для каждой проверяем наличие комбинации F*O, если комбинация присутствует, то увеличиваем количество подходящих строк на 1.

```
var f : text;
    s : string;
    i, k : integer;
    good: boolean;
begin
assign(f, "24.txt");
reset(f);
k:=0;
while not eof(f) do begin
    readln(f, s);
    good:=false;
    for i:=1 to length(s)-2 do
        if (s[i]="F") and (s[i+2]="0") then begin
            good:=true;
            break;
        end;
    if good then k:=k+1;
end;
writeln(k);
close(f);
end.
```

```
f = open("24.txt")
k = 0
for s in f:
    good = False
    for i in range(len(s)-2):
        if s[i]=="F" and s[i+2]=="0":
            good = True
            break
    if good: k+=1
print(k)
f.close()
```

Поиск количества подстрок

Задача 10. Текстовый файл состоит не более чем из 10^6 заглавных латинских букв. Определите символ, который чаще всего встречается в файле сразу после буквы X. В ответе запишите сначала этот символ, а потом сразу (без разделителя) сколько раз он встретился после буквы X. Например, в тексте XBCXXBXDDD после буквы X два раза стоит B, по одному разу – X и D. Для этого текста ответом будет B2.

Описание решения

Для перебора символов удобно использовать цифровое представление символов в кодировке.

`ord()` – порядковый номер символа в кодировке

`chr()` – символ с указанным порядковым номером

Пример: перебор заглавных латинских букв удобно делать с помощью следующего цикла:

```
for i in range(ord("A"), ord("Z")+1):  
    c = chr(i)  
    ...
```

Описание решения

Начальные значения:

$\text{maxk} = 0$ #наибольшая частота n -ого символа после X

$\text{maxch} = ""$ #самый частый символ после X

$\text{ch} = 0$ #текущий проверяемый символ

$k = 0$ #количество n -ого символа после X

Поочерёдно перебираем все заглавные латинские буквы. Для каждой считаем сколько раз она встречается после X и сравниваем найденное количество с наибольшим значением.

Если текущий символ встречается чаще, то сохраняется и количество и этот символ.

```
var f : text;
    s : string;
    i, j, maxk, k : integer;
    ch, maxch: char;
begin
assign(f, "24.txt");
reset(f);
readln(f, s);
maxk:=0; maxch:='';
for i:=ord('A') to ord('Z') do begin
    k:=0; c:=chr(i);
    for j:=1 to length(s)-1 do
        if s[j]+s[j+1] = 'X'+c then k:=k+1;
    if k>maxk then begin
        maxk:=k;
        maxch:=ch;
    end;
end;
writeln(maxch, maxk);
close(f);
end.
```

```
f = open("24.txt")
maxk, maxch = 0, ""
s = f.readline()
for i in range(ord("A"), ord("Z")+1):
    k = 0
    c = chr(i)
    for j in range(len(s)-1):
        if s[j]+s[j+1]==c+c:
            k+=1
    if k>maxk:
        maxk, maxch = k, c
print(maxch, maxk)
f.close()
```