

## 24 (высокий уровень, время – 18 минут)

**Тема:** Обработка символьных строк

**Что проверяется:**

Умение создавать собственные программы (10–20 строк) для обработки символьной информации.

1.5.2. Цепочки (конечные последовательности), деревья, списки, графы, матрицы (массивы), псевдослучайные последовательности.

1.1.3. Строить информационные модели объектов, систем и процессов в виде алгоритмов.

**Что нужно знать:**

- сначала нужно прочитать строку из файла; эта задача в разных языках программирования решается несколько по-разному
- в языке Python удобнее всего использовать такую конструкцию:

```
with open("k7.txt", "r") as F:
    s = F.readline()
```

конструкция **with-as** – это *контекстный менеджер*, в данном случае он открывает указанный файл в режиме чтения (второй аргумент «r» при вызове функции **open**), записывает ссылку на него в файловую переменную **F**, выполняет тело блока (читает первую строку файла в переменную **s**) и закрывает (освобождает) файл

- в языке PascalABC.NET можно выполнить перенаправление потока ввода:

```
assign( input, 'k7.txt' );
readln(s);
```

программа будет «думать», что читает данные, введённые с клавиатуры (с консоли), а на самом деле эти данные будут прочитаны из файла **k7.txt**

- в языке FreePascal также можно выполнить перенаправление потока ввода, но нужно дополнительно открывать входной поток:

```
assign( input, 'k7.txt' );
reset( input );           { для FreePascal!!! }
readln(s);
```

- **при работе в среде FreePascal** нужно убедиться, что в параметрах компилятора включена поддержка **длинных символьных строк**; на всякий случай стоит добавить в первой строке программы директиву

```
{ $H+ }
```

- Среда PascalABC НЕ ПОДДЕРЖИВАЕТ работу с **длинными символьными строками**, поэтому для решения задачи использовать версию PascalABC.NET, которую можно бесплатно скачать с сайта автора [www.pascalabc.net](http://www.pascalabc.net).

- в языке C++ используем потоки:

```
#include <fstream>
#include <string>
using namespace std;
int main()
{
    ifstream F("k7.txt");
    string s;
    getline( F, s );
    ...
}
```

**Самая длинная цепочка символов «С»**

- пусть требуется найти самую длинную цепочку символов С (или каких-то других, в соответствии с заданием) в символьной строке **s**;
- можно использовать такой алгоритм:

```
while не конец строки:
    найти очередную букву С
    длина := длина текущей цепочки букв С
    if длина > максимальной длины:
        максимальная длина := длина
```

однако этот алгоритм содержит вложенный цикл и при составлении программы легко запутаться и не учесть какой-то особый случай (например, когда строка состоит только из букв С)

- лучше применить однопроходный алгоритм без вложенного цикла

```
for c in s:
    обработать символ c
```

- будем использовать переменные  
**cLen** – длина текущей цепочки букв С  
**maxLen** – максимальная длина цепочки букв С на данный момент
- рассмотрим очередной символ строки; если это буква С, увеличиваем **cLen** на 1 и, если нужно, запоминаем новую максимальную длину; если это не буква С, просто записываем с **cLen** ноль:

```
maxLen = 0
cLen = 0
for c in s:
    if c == 'C':
        cLen += 1          # ещё одна буква С
        if cLen > maxLen:  # возможно, новая максимальная длина
            maxLen = cLen
    else:
        cLen = 0          # цепочка букв С кончилась
```

- проверим правильность работы алгоритма в особых случаях:
  - а) если вся строка состоит из букв С, значение переменной **cLen** постоянно увеличивается и в конце станет равно длине символьной строки; то же значение окажется и в переменной **maxLen**;
  - б) если в строке нет символов С, переменная **cLen** всегда равна 0, такое же значение будет и в переменной **maxLen**

**Самая длинная цепочка любых символов**

- теперь поставим задачу найти самую длинную цепочку символов в символьной строке **s**; сложность состоит в том, что мы (в отличие от предыдущей задачи) не знаем, из каких именно символов состоит самая длинная цепочка
- если символов в алфавите немного (скажем, А, В и С), то можно с помощью описанного выше алгоритма найти самые длинные цепочки из букв А, В и С, а затем выбрать из них «длиннейшую»; такая идея может сработать при аккуратной реализации, но плохо обобщается на случай, когда возможных символов много (например, используются все заглавные латинские буквы и цифры)
- поэтому лучше применить однопроходный алгоритм без вложенного цикла
- будем использовать переменные  
**curLen** – длина текущей цепочки одинаковых символов  
**maxLen** – максимальная длина цепочки одинаковых символов на данный момент

**c** – символ, из которого строится самая длинная подцепочка

- в начальный момент рассмотрим один первый символ (цепочка длины 1 есть всегда!):

```
maxLen = 1
curLen = 1
c = s[0]
```

- будем перебирать в цикле все символы, начиная с **s[1]** (второго по счёту) до конца строки, постоянно «оглядываясь назад», на предыдущий символ

```
for i in range(1, len(s)):
    обработать пару символов s[i-1] и s[i]
```

- если очередной символ **s[i]** такой же, как и предыдущий, цепочка одинаковых символов продолжается, и нужно увеличить значение переменной **curLen**; если значение **curLen** стало больше **maxLen**, обновляем **maxLen** и запоминаем новый базовый символ в переменной **c**:

```
if s[i] == s[i-1]:      # цепочка продолжается
    curLen += 1         # увеличиваем длину
    if curLen > maxLen:  # если цепочка побила рекорд
        maxLen = curLen # запоминаем её длину...
        c = s[i]        # и образующий символ
else:
    curLen = 1          # началась новая цепочка
```

если очередной символ не совпал с предыдущим, началась новая цепочка, и её длина пока равна 1 (это значение записывается в переменную **curLen**)

- получается такой цикл обработки строки:

```
maxLen, curLen, c = 1, 1, s[0]
for i in range(1, len(s)):
    if s[i] == s[i-1]:
        curLen += 1
        if curLen > maxLen:
            maxLen = curLen
            c = s[i]
    else:
        curLen = 1
```

- проверим правильность работы алгоритма в особых случаях:
  - в) если вся строка состоит из одинаковых символов, значение переменной **curLen** постоянно увеличивается и в конце станет равно длине символьной строки; то же значение окажется и в переменной **maxLen**;
  - г) если в строке нет пар одинаковых символов, переменная **curLen** всегда равна 1, такое же значение будет и в переменной **maxLen**

## Пример задания:

**P-07 (демо-2021).** Текстовый файл **24.txt** состоит не более чем из  $10^6$  символов X, Y и Z. Определите максимальное количество идущих подряд символов, среди которых каждые два соседних различны. Для выполнения этого задания следует написать программу.

### Решение:

- 1) считывание из файла и перебор символов аналогичен задачам P00-P02 (см. ниже).
- 2) чтобы считать длину цепочки, соответствующей условию, нам нужно будет ввести два счётчика:  
**curLen** – длина текущей цепочки (которая сейчас обрабатывается)

**maxLen** – длина самой длинной на данный момент цепочки в уже обработанной части строки

- 3) обработка строки сводится к тому, что текущая длина цепочки увеличивается, если соседние символы, `s[i-1]` и `s[i]`, различны; если это не так, сбрасываем длину текущей цепочки в 1
- 4) можно заметить, что эта задача очень напоминает Р-05, только тут обратное условие – нужно искать цепочку, где все соседние символы не одинаковые, а разные, поэтому и решение сводится к изменению условия (см. выделение маркером):

```
with open( "24.txt", "r" ) as F:
    s = F.readline()
    maxLen, curLen = 1, 1
    for i in range(1, len(s)):
        if s[i] != s[i-1]:
            curLen += 1
            if curLen > maxLen:
                maxLen = curLen
        else:
            curLen = 1
    print( maxLen )
```

- 5) Ответ: 35.

- 6) программа на Паскале:

```
var maxLen, curLen, i: integer;
    s: string;
begin
    assign(input, '24.txt');
    readln(s);
    maxLen := 1;
    curLen := 1;
    for i:=2 to Length(s) do
        if s[i] <> s[i-1] then begin
            curLen := curLen + 1;
            if curLen > maxLen then
                maxLen := curLen;
        end
        else
            curLen := 1;
    writeln(maxLen);
end.
```

- 7) программа на C++:

```
#include <iostream>
#include <fstream>
#include <string>
using namespace std;
int main()
{
    ifstream F("24.txt");
    string s;
    getline( F, s );
    int maxLen = 1, curLen = 1;
    for( int i = 1; i < s.length(); i++ )
        if( s[i] != s[i-1] ) {
```

```

        curLen ++;
        if( curLen > maxLen )
            maxLen = curLen;
        }
        else curLen = 1;
        cout << maxLen;
    }

```

### Пример задания:

**P-06.** В текстовом файле `k8.txt` находится цепочка из символов, в которую могут входить заглавные буквы латинского алфавита A...Z и десятичные цифры. Найдите длину самой длинной подцепочки, состоящей из одинаковых символов. **Для каждой цепочки максимальной длины** выведите в отдельной строке сначала символ, из которого строится эта цепочка, а затем через пробел – длину этой цепочки.

#### Решение:

- 1) особенность этой задачи в сравнении с P-05 состоит в следующем: если найдено несколько цепочек одинаковой максимальной длины, для каждой из них нужно вывести символ, из которого состоит цепочка, и длину цепочки
- 2) это значит, что для хранения символа нужна не одна переменная, а массив (в Python – список); если найдена первая цепочка (выполнено условие)
- 3) итак, теперь `c` – это массив (список); когда найдена первая цепочка максимальной длины (на данный момент), в этот массив записывается символ этой цепочки; если же найдена новая цепочка такой же длины, в массив **добавляется** символ этой цепочки
- 4) таким образом, в конце прохода в массиве `c` находятся все символы, из которых состоят самые длинные цепочки, и остаётся вывести их на экран; справа от каждого символа выводится длина цепочки:

```

for c1 in c:
    print( c1, maxLen )

```

- 5) вот полная программа (изменения в сравнении с решением задачи P-05 выделены):

```

with open( "k8.txt", "r" ) as F:
    s = F.readline()
    maxLen, curLen, c = 1, 1, [s[0]] # c – массив из одного символа
    for i in range(1, len(s)):
        if s[i] == s[i-1]:
            curLen += 1
            if curLen == maxLen: # новая цепочка максимальной длины
                c.append( s[i] ) # добавить символ в массив
            elif curLen > maxLen:
                maxLen = curLen
                c = [s[i]] # c – массив из одного символа
        else:
            curLen = 1
    for c1 in c: # для всех символов в массиве
        print( c1, maxLen ) # вывести символ и длину

```

Решение (программа на языке Pascal):

- 1) проблема состоит в том, что мы не знаем, сколько цепочек максимальной длины может быть в файле; тут нужен динамический массив (список), для этого далее мы будем использовать язык PascalABC.NET, в котором есть тип данных **List** (список)
- 2) вначале создаём новый список и записываем у него первый символ строки:

```
var c := new List<char>;
c.Add( s[1] );
```

- 3) если нашли новую (не первую) цепочку максимальной длины, добавляем новый символ в список:

```
c.Add( s[i] );
```

- 4) если нашли новую самую длинную цепочку с длиной БОЛЬШЕЙ, чем все предыдущие, очищаем список и добавляем в него новый символ.

```
c.Clear;
c.Add( s[i] );
```

- 5) после окончания обработки нужно вывести все символы и длины цепочек, удобнее всего использовать для этого цикл **foreach**; получается почти так же, как и на Python:

```
foreach var c1 in c do
    writeln( c1, ' ', maxLen );
```

- 6) вот полная программа:

```
var maxLen, curLen, i: integer;
    s: string;
begin
    assign(input, 'k8.txt');
    readln(s);
    maxLen := 1;
    curLen := 1;
    var c := new List<char>;
    c.Add( s[1] );
    for i:=2 to Length(s) do
        if s[i] = s[i-1] then begin
            curLen := curLen + 1;
            if curLen = maxLen then begin
                c.Add( s[i] );
            end
        else if curLen > maxLen then begin
            maxLen := curLen;
            c.Clear;
            c.Add( s[i] );
        end
    end
    foreach var c1 in c do
        writeln( c1, ' ', maxLen );
    end.
```

#### Решение (программа на языке C++):

- 1) динамический массив (список), для этого далее мы будем использовать тип данных **list** (список) из библиотеки STL; не забудьте, что нужно подключить заголовочный файл **list**:

```
#include <list>
```

- 2) вначале создаём список символов, состоящий из одного первого символа строки:

```
list<char> c( s[0] );
```

- 3) если нашли новую (не первую) цепочку максимальной длины, добавляем новый символ в список с помощью метода `push_back`:

```
c.push_back( s[i] );
```

- 4) если нашли новую самую длинную цепочку с длиной большей, чем все предыдущие, очищаем список и добавляем в него новый символ:

```
c.clear();
```

```
c.push_back( s[i] );
```

- 5) после окончания обработки нужно вывести все символы и длины цепочек, удобнее всего использовать для этого особую форму цикла `for`; получается почти так же, как и на Python:

```
for( char c1: c )
    cout << c1 << ' ' << maxLen << endl;
```

- 6) вот полная программа:

```
#include <iostream>
#include <fstream>
#include <string>
#include <list>
using namespace std;
int main()
{
    ifstream F("k8.txt");
    string s;
    getline( F, s );
    int maxLen = 1, curLen = 1;
    list<char> c(s[0]);
    for( int i = 1; i < s.length(); i++ )
        if( s[i] == s[i-1] ) {
            curLen ++;
            if( curLen == maxLen )
                c.push_back(s[i]);
            else if( curLen > maxLen ) {
                maxLen = curLen;
                c.clear();
                c.push_back( s[i] );
            }
        }
        else curLen = 1;
    for( char c1: c )
        cout << c1 << ' ' << maxLen << endl;
}
```

### Ещё пример задания:

**Р-05.** В текстовом файле `k8.txt` находится цепочка из 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.

**Замечание (Б.С. Михлин).** Может случиться так, что в файле будут несколько самых длинных цепочек (одинаковой длины), состоящих из разных символов. На этот случай условие задачи требует

уточнения – какой именно символ выводить в ответе? Далее мы будем считать, что в этом случае нужно вывести символ, который формирует первую цепочку максимальной длины.

#### Решение:

- 1) сначала нужно открыть файл и прочитать все символы в символьную строку:

```
with open("k8.txt", "r") as F:
    s = F.readline()
```

- 2) теперь задача свелась к определению наибольшего количества подряд идущих одинаковых символов в символьной строке `s` (этот алгоритм приведён в начале файла)
- 3) полная программа на языке Python:

```
with open( "k8.txt", "r" ) as F:
    s = F.readline()
    maxLen, curLen, c = 1, 1, s[0]
    for i in range(1, len(s)):
        if s[i] == s[i-1]:
            curLen += 1
            if curLen > maxLen:
                maxLen = curLen
                c = s[i]
        else:
            curLen = 1
    print( c, maxLen )
```

Обратим внимание, что условие

```
if curLen > maxLen: ...
```

гарантирует, что будет запомнена именно первая цепочка максимальной длины, так как это условие выполнится, когда новая цепочка строго длиннее предыдущего «рекорда». Если бы нужно было вывести символ, формирующий **последнюю** из самых длинных цепочек, неравенство следовало бы сделать нестрогим:

```
if curLen >= maxLen: ...
```

#### Решение (программа на языке Pascal):

- 7) далее мы будем использовать язык PascalABC.NET, который обладает двумя важными достоинствами, упрощающими решение таких задач:

- позволяет легко переключать входной поток с консоли на нужный файл:

```
assign( input, 'k8.txt' );
```

теперь можно писать программу так же, как и при вводе данных с клавиатуры, а она на самом деле будет читать их из указанного файла;

- не имеет ограничения на длину строк (переменных типа **string**); в устаревших версиях языка Pascal длина строки не может превышать 255 символов

- 8) читаем одну строку из файла в строковую переменную `s`:

```
readln( s );
```

- 9) теперь можно в цикле перебрать все символы строки `s`, начиная **со второго** (чтобы сравнивать его с предыдущим):

```
for i:=2 to Length(s) do
    обработать пару s[i-1] и s[i]
```

- 10) обработка выполняется по алгоритму, описанному выше (см. программу на Python)

- 11) полная программа на языке Pascal:

```
var maxLen, curLen, i: integer;
    s: string;
    c: char;
```



```

begin
  assign(input, 'k8.txt');
  readln(s);
  maxLen := 1;
  curLen := 1;
  c := s[1];
  for i:=2 to Length(s) do
    if s[i] = s[i-1] then begin
      curLen := curLen + 1;
      if curLen > maxLen then begin
        maxLen := curLen;
        c := s[i];
      end
    end
  else
    curLen := 1;
  writeln(c, ' ', maxLen);
end.

```

#### Решение (программа на языке C++):

- 1) в C++ удобно работать с файлами через файловые потоки; для того, чтобы использовать эту возможность, нужно подключить заголовочный файл **fstream**:

```
#include <fstream>
```

- 2) теперь можно открыть файловый поток, связать его с нужным файлом и прочитать из потока строку в переменную типа **string**:

```

ifstream F("k8.txt");
string s;
getline( F, s );

```

- 3) алгоритм обработки строки тот же, что и в программах на языках Python и Pascal, рассмотренных выше
- 4) поскольку в программе используется много объектов из пространства имён **std**, имеет смысл подключить его в начале программы:

```
using namespace std;
```

- 5) полная программа на языке C++:

```

#include <iostream>
#include <fstream>
#include <string>
using namespace std;
int main()
{
  ifstream F("k8.txt");
  string s;
  getline( F, s );
  int maxLen = 1, curLen = 1;
  char c = s[0];
  for( int i = 1; i < s.length(); i++ )
    if( s[i] == s[i-1] ) {
      curLen ++;
      if( curLen > maxLen ) {
        maxLen = curLen;
        c = s[i];
      }
    }
}

```

```

    }
    }
    else curLen = 1;
    cout << c << ' ' << maxLen;
}

```

### Ещё пример задания:

**P-04. (А.М. Кабанов)** В текстовом файле **k7.txt** находится цепочка из символов латинского алфавита A, B, C, D, E. Найдите количество цепочек длины 3, удовлетворяющих следующим условиям:

- 1-й символ – один из символов B, C или D;
- 2-й символ – один из символов B, D, E, который не совпадает с первым;
- 3-й символ – один из символов B, C, E, который не совпадает со вторым.

#### Решение:

- 6) Считывание из файла и перебор символов аналогичен задачам P00-P02 (см. ниже).
- 7) Переберём все тройки символов. Примем, что переменная *i* будет хранить номер первого элемента в тройке, то есть, будем рассматривать тройки (*s*[*i*], *s*[*i*+1], *s*[*i*+2]).
- 8) Организуем цикл который перебирает значения *i* от 1 до *len(s) - 2*

```

for i in range(len(s)-2):
    ...

```

- 9) Проверяем символы в каждой тройке на соответствие условию. Проверка принадлежности символов набору аналогична заданию 1. Дополнительно необходимо указать условия неравенства символов, указанных в условии задачи. Если условия выполняются, то к переменной количества прибавляется единица.
- 10) полная программа на Python:

```

s = open('k7.txt').read()
count = 0
for i in range(len(s)-2):
    if s[i] in 'BCD' and s[i+1] in 'BDE' \
        and s[i+2] in 'BCE' and s[i]!=s[i+1] \
        and s[i+1]!=s[i+2]:
        count += 1
print(count)

```

#### Решение (программа на языке PascalABC.NET):

```

begin
    var s: string;
    var i, count: integer;
    assign(input, 'k7.txt');
    readln(s);
    count:=0;
    for i:=1 to Length(s)-2 do
        if (s[i] in 'BCD') and (s[i+1] in 'BDE')
            and (s[i+2] in 'BCE') and (s[i]<>s[i+1])
            and (s[i+1]<>s[i+2]) then
            count := count+1;
    writeln(count);
end.

```

#### Решение (программа на языке C++):

```

using namespace std;
#include <iostream>
#include <fstream>
#include <string>
int main() {
    ifstream F("k7.txt");
    string s;
    getline( F, s );
    int count = 0;
    for( int i = 0; i < s.length()-2; i++ )
        if( (s[i]=='B' || s[i]=='C' || s[i]=='D')
            && (s[i+1]=='B' || s[i+1]=='D' || s[i+1]=='E')
            && (s[i+2]=='B' || s[i+2]=='C' || s[i+2]=='E')
            && s[i]!=s[i+1] && s[i+1]!=s[i+2] )
            count++;
    cout << count;
}

```

### Ещё пример задания:

**P-03. (А.М. Кабанов)** В текстовом файле `k7.txt` 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).

#### Решение:

- 1) Считывание из файла и перебор символов аналогичен задачам P00-P02 (см. ниже).
- 2) Проверка того, что символ принадлежит цепочке, производится следующим образом. Заметим, что в искомой цепочке чередуется группа из трёх символов (**EAB**). Пронумеруем символы искомой цепочки, начиная с нуля.

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

- 3) Видно, что позиция каждого символа имеет одинаковый остаток от деления на 3. Позиция есть значения переменной счётчика в момент проверки символа. Поэтому если совпадает символ и соответствующий ему остаток от деления, то он принадлежит цепочке. Для приведённого примера условие проверки выглядит так

```

if (char == 'E' and count%3==0) or \
    (char == 'A' and count%3==1) or \
    (char == 'B' and count%3==2):

```

- 4) Если символ не является частью этой цепочки, но может являться её началом (E), длина цепочки принимается равной единице, в противном случае длина обнуляется

```

elif (char=='E'):
    count = 1
else:
    count = 0

```

- 5) Полная программа на языке Python:

```

s = open('k7.txt').read()
count = 0
maxCount = 0
for char in s:
    if (char == 'E' and count%3 == 0) or \

```

```

(char == 'A' and count%3 == 1) or \
(char == 'B' and count%3 == 2):
    count += 1
    if count > maxCount:
        maxCount = count
elif (char=='E'):
    count = 1
else:
    count = 0
print(maxCount)

```

Решение (полная программа на языке PascalABC.NET):

```

begin
    var s: string;
    var i, count, maxCount: integer;
    assign(input, 'k7.txt');
    readln(s);
    count:=0;
    maxCount:=0;
    for i:=1 to Length(s) do
        if ((s[i]='E') and (count mod 3=0)) or
            ((s[i]='A') and (count mod 3=1)) or
            ((s[i]='B') and (count mod 3=2)) then begin
            count := count+1;
            if count > maxCount then
                maxCount := count;
        end
        else if s[i]='E' then count:=1
        else count := 0;

        writeln(maxCount);
    end.

```

Решение (полная программа на языке C++):

```

using namespace std;
#include <iostream>
#include <fstream>
#include <string>
int main(){
    ifstream F("k7.txt");
    string s;
    getline( F, s );
    int count = 0, maxCount = 0;
    for( int i = 0; i < s.length(); i++ )
        if( (s[i] == 'E' && count%3 == 0) ||
            (s[i] == 'A' && count%3 == 1) ||
            (s[i] == 'B' && count%3 == 2) ) {
            count ++;
            if( count > maxCount )
                maxCount = count;
        }
    else if(s[i]== 'E') count = 1

```

```

        else count = 0
        cout << maxCount;
    }

```

Решение (программа на Python, М. Магомедов):

```

with open( "k7.txt", "r" ) as F:
    s = F.readline()
    k = ''
    while k in s:
        k += 'E'
        if k in s:
            k += 'A'
            if k in s: k += 'B'
    print ( len (k) - 1 )

```

Ещё пример задания:

**Р-02. (А.М. Кабанов)** В текстовом файле `k7.txt` находится цепочка из символов латинского алфавита A, B, C, D, E. Найдите длину самой длинной подцепочки, состоящей из символов A, B или C (в произвольном порядке).

Решение:

- 1) сначала нужно открыть файл и прочитать все символы в символьную строку:

```
s = open('k7.txt').read()
```

- 2) теперь можно в цикле перебрать все символы строки `s`:

```

for char in s:
    ...

```

- 3) теперь задача свелась к определению наибольшей подстроки, состоящей из символов A, B или C, в символьной строке `s`.
- 4) Проверку того, что символ – один из набора A, B, C удобно записывать с помощью условия

```
if char in 'ABC':
```

- 5) Полная программа на языке Python:

```

s = open('k7.txt').read()
count = 0
maxCount = 0
for char in s:
    if char in 'ABC':
        count += 1
        if count > maxCount:
            maxCount = count
    else:
        count = 0
print(maxCount)

```

Решение (программа на языке PascalABC.NET):

- 1) В начале переключим входной поток с консоли на нужный файл, а затем считаем одну строку из файла в строковую переменную `s`

```

assign(input, 'k7.txt');
readln(s);

```

- 2) теперь можно в цикле перебрать все символы строки `s`:

```

for i:=1 to Length(s) do
    ...

```

- 3) обработка символа выполняется по алгоритму, описанному выше (см. программу на Python)
- 4) Проверка того, что символ – один из набора А, В, С – в PascalABC.NET записывается аналогично

```
if s[i] in 'ABC' then
```

а в среде FreePascal придётся использовать старинный способ:

```
if s[i] in ['A','B','C'] then
```

- 5) Полная программа на языке PascalABC.NET:

```
begin
  var s: string;
  var i, count, maxCount: integer;
  assign(input, 'k7.txt');
  readln(s);
  count:=0;
  maxCount:=0;
  for i:=1 to Length(s) do
    if s[i] in 'ABC' then { if s[i] in ['A','B','C'] }
    begin
      count := count+1;
      if count > maxCount then
        maxCount := count;
    end
    else
      count := 0;
  writeln(maxCount);
end.
```

#### Решение (программа на языке C++):

- 1) Для чтения из файла подключим заголовочный файл **fstream**, откроем файловый поток и считаем его в строковую переменную **s**

```
#include <fstream>
#include <string>
...
ifstream F("k7.txt");
string s;
getline( F, s );
```

- 2) алгоритм обработки строки тот же, что и в программах на языках Python и Pascal, рассмотренных выше, однако проверка того, что символ – один из набора А, В, С записывается по-другому

```
if( s[i]=='A' || s[i]=='B' || s[i]=='C' )
```

- 3) Полная программа на языке C++:

```
using namespace std;
#include <iostream>
#include <fstream>
#include <string>
int main(){
  ifstream F("k7.txt");
  string s;
  getline( F, s );
  int count = 0, maxCount = 0;
  for( int i = 0; i < s.length(); i++ )
```

```

        if( s[i] == 'A' || s[i] == 'B' || s[i] == 'C' ) {
            count ++;
            if( count > maxCount )
                maxCount = count;
        }
        else count = 0;
        cout << maxCount;
    }

```

### Ещё пример задания:

**P-01.** В текстовом файле **k7.txt** находится цепочка из символов латинского алфавита A, B, C. Найдите длину самой длинной подцепочки, состоящей из символов C.

#### Решение:

- 1) сначала нужно открыть файл и прочитать все символы в символьную строку:

```

with open("k7.txt", "r") as F:
    s = F.readline()

```

- 2) теперь задача свелась к определению наибольшего количества подряд идущих букв C в символьной строке **s** (этот алгоритм приведён в начале файла)
- 3) полная программа на языке Python:

```

with open( "k7.txt", "r" ) as F:
    s = F.readline()
    maxLen, cLen = 0, 0
    for c in s:
        if c == 'C':
            cLen += 1
            if cLen > maxLen:
                maxLen = cLen
        else:
            cLen = 0
    print( maxLen )

```

#### Решение (программа на языке Pascal):

- 7) далее мы будем использовать язык PascalABC.NET, который обладает двумя важными достоинствами, упрощающими решение таких задач:

- позволяет легко переключать входной поток с консоли на нужный файл:

```
assign( input, 'k7.txt' );
```

теперь можно писать программу так же, как и при вводе данных с клавиатуры, а она на самом деле будет читать их из указанного файла;

- не имеет ограничения на длину строк (переменных типа **string**); в устаревших версиях языка Pascal длина строки не может превышать 255 символов

- 8) читаем одну строку из файла в строковую переменную **s**:

```
readln(s);
```

- 9) теперь можно в цикле перебрать все символы строки **s**:

```

for i:=1 to Length(s) do
    обработать s[i]

```

- 10) обработка символа выполняется по алгоритму, описанному выше (см. программу на Python)

- 11) полная программа на языке Pascal:

```

var maxLen, cLen, i: integer;
    s: string;

```

```

begin
  assign(input, 'k7.txt');
  readln(s);
  maxLen := 0;
  cLen := 0;
  for i:=1 to Length(s) do
    if s[i] = 'C' then begin
      cLen := cLen + 1;
      if cLen > maxLen then maxLen := cLen;
    end
    else
      cLen := 0;
  writeln(maxLen);
end.

```

#### Решение (программа на языке C++):

- 6) в C++ удобно работать с файлами через файловые потоки; для того, чтобы использовать эту возможность, нужно подключить заголовочный файл **fstream**:

```
#include <fstream>
```

- 7) для того чтобы читать всю строку целиком с помощью функции **getline**, нужно подключить заголовочный файл **string**:

```
#include <string>
```

- 8) теперь можно открыть файловый поток, связать его с нужным файлом и прочитать из потока строку в переменную типа **string**:

```

ifstream F("k7.txt");
string s;
getline( F, s );

```

- 9) алгоритм обработки строки тот же, что и в программах на языках Python и Pascal, рассмотренных выше
- 10) поскольку в программе используется много объектов из пространства имён **std**, имеет смысл подключить его в начале программы:

```
using namespace std;
```

- 11) полная программа на языке C++:

```

#include <iostream>
#include <fstream>
#include <string>
using namespace std;
int main()
{
  ifstream F("k7.txt");
  string s;
  getline( F, s );
  int maxLen = 0, cLen = 0;
  for( int i = 0; i < s.length(); i++ )
    if( s[i] == 'C' ) {
      cLen ++;
      if( cLen > maxLen )
        maxLen = cLen;
    }
  else cLen = 0;
}

```



```
cout << maxLen;
}
```

#### Решение методом грубой силы (Б.С. Михлин):

- 1) если решить красиво не получается, можно применить метод грубой силы, использующий встроенную функцию поиска подстроки: ищем строку из одного символа C, потом – из двух символов, из трёх и т.д.; в какой-то момент поиск не даст результата, значит ответ – это длина предыдущей цепочки, которая короче текущей на единицу

- 2) вот решение на Python:

```
with open( "k7.txt", "r" ) as F:
    s = F.readline()
    cc = 'C'
    while cc in s:
        cc += 'C'
    print( len(cc)-1 )
```

- 3) решение на Паскале:

```
var cc, s: string;
begin
    assign(input, 'k7.txt');
    readln(s);
    cc := 'C';
    while Pos(cc, s) > 0 do
        cc := cc + 'C';
    writeln( Length(cc)-1 );
end.
```

- 4) решение на C++:

```
#include <iostream>
#include <fstream>
#include <string>
using namespace std;
int main()
{
    ifstream F("k7.txt");
    string s, cc;
    getline( F, s );
    cc = 'C';
    while( s.find(cc) != string::npos )
        cc += 'C';
    cout << cc.length()-1;
}
```

- 5) эту задачу можно решать вообще без программирования, используя функцию поиска в любом текстовом редакторе или процессоре; для ускорения можно сначала удваивать длину искомой цепочки, а после того, как поиск закончится неудачно, применять двоичный поиск в интервале
- 6) конечно, нужно понимать, что эффективность (скорость работы) этого алгоритма крайне низкая в сравнении с описанным выше однопроходным поиском, но на небольших файлах и этот метод вполне может сработать.

#### Решение в электронных таблицах (Б.С. Михлин):

- 1) можно применить тот же метод грубой силы, использующий электронные таблицы. Сначала откроем файл в текстовом редакторе и скопируем все его содержимое в буфер обмена. Затем

откроем новую электронную таблицу и вставим строку из буфера обмена в какую-нибудь ячейку (в примере ниже это ячейка A2). Затем в окне «Найти» вбиваем один символ «С» и нажимаем кнопку «Найти все», потом два символа «С», три и т.д., пока не появится сообщение «...не удастся найти искомые данные». Значит максимальная длина подцепочки из символов «С», входящая в заданную цепочку, на единицу меньше. При большой длине максимальной подцепочки при подсчете в ней количества символов легко ошибиться.

- 2) можно также использовать встроенные текстовые функции электронных таблиц: FIND (НАЙТИ) или SEARCH (ПОИСК) и REPT (ПОВТОР). Меняя в функции ПОВТОР коэффициент повторения символа "С" мы повторяем идею п. 1. Для ускорения поиска можно коэффициент повторения менять сперва с крупным шагом, а затем с более мелким. Также можно обойтись только одной ячейкой с формулой.

Функции НАЙТИ и ПОИСК выводят положение начала искомой подцепочки в заданной цепочке символов или сообщение #ЗНАЧ!, если подцепочка не найдена. Если поиск надо осуществлять с начала цепочки, то третий параметр функций НАЙТИ и ПОИСК можно не указывать. Функция НАЙТИ учитывает регистр символов. Функция ПОИСК не учитывает регистр символов и в ней можно использовать подстановочные символы (\* и ?).

Вот решение задачи в OpenOffice Calc:

	А	В	С
1			
2	CCCCBACCBCCCCACCABCAACCACCCCAABAAABBBCCCCBCCCCBVBCCBACCACCABCAACCBVBBCACBCCCCACCCCBAAAC		
3	<b>Результат формулы</b>	<b>Формула</b>	<b>Комментарии</b> В ячейке A2 находится заданная цепочка из файла K7-53.txt. (Ее длина не должна превышать 32767)
4		<b>FIND</b>	
5	1	FIND(REPT("C";1);A2)	Подцепочка из одного символа "С" начинается с позиции 1
6	354	FIND(REPT("C";10);A2)	Подцепочка из десяти символов "С" начинается с позиции 354
7	#ЗНАЧЕН!	FIND(REPT("C";20);A2)	Подцепочка из двадцати символов "С" не найдена.
8	#ЗНАЧЕН!	FIND(REPT("C";15);A2)	Подцепочка из пятнадцати символов "С" тоже не найдена.
9	763	FIND(REPT("C";14);A2)	Подцепочка из четырнадцати символов "С" начинается с позиции 763. Т.е. она входит в заданную цепочку.
10	<b>Ответ:</b>	<b>14</b>	Максимальная длина подцепочки из символов "С"
11		<b>SEARCH</b>	
12	1	SEARCH(REPT("C";1);A2)	Подцепочка из одного символа "С" начинается с позиции 1
13	354	SEARCH(REPT("C";10);A2)	Подцепочка из десяти символов "С" начинается с позиции 354
14	#ЗНАЧЕН!	SEARCH(REPT("C";20);A2)	Подцепочка из двадцати символов "С" не найдена.
15	#ЗНАЧЕН!	SEARCH(REPT("C";15);A2)	Подцепочка из пятнадцати символов "С" тоже не найдена.
16	763	SEARCH(REPT("C";14);A2)	Подцепочка из четырнадцати символов "С" начинается с позиции 763. Т.е. она входит в заданную цепочку.
17	<b>Ответ:</b>	<b>14</b>	Максимальная длина подцепочки из символов "С"

и в русской версии Excel:

	А	В	С
1			
2	CCCCBACCBCCCCACCABCAACCACCCCAABAAABBBCCCCBCCCCBVBCCBACCACCABCAACCBVBBCACBCCCCACCCCBAAAC		
3	<b>Результат формулы</b>	<b>Формула</b>	<b>Комментарии</b> В ячейке A2 находится заданная цепочка из файла K7-53.txt. (Ее длина не должна превышать 32767)
4		<b>НАЙТИ</b>	
5	1	НАЙТИ(ПОВТОР("С";1);A2)	Подцепочка из одного символа "С" начинается с позиции 1
6	354	НАЙТИ(ПОВТОР("С";10);A2)	Подцепочка из десяти символов "С" начинается с позиции 354
7	#ЗНАЧ!	НАЙТИ(ПОВТОР("С";20);A2)	Подцепочка из двадцати символов "С" не найдена.
8	#ЗНАЧ!	НАЙТИ(ПОВТОР("С";15);A2)	Подцепочка из пятнадцати символов "С" тоже не найдена.
9	763	НАЙТИ(ПОВТОР("С";14);A2)	Подцепочка из четырнадцати символов "С" начинается с позиции 763. Т.е. она входит в заданную цепочку.
10	<b>Ответ:</b>	<b>14</b>	Максимальная длина подцепочки из символов "С"
11		<b>ПОИСК</b>	
12	1	ПОИСК(ПОВТОР("С";1);A2)	Подцепочка из одного символа "С" начинается с позиции 1
13	354	ПОИСК(ПОВТОР("С";10);A2)	Подцепочка из десяти символов "С" начинается с позиции 354
14	#ЗНАЧ!	ПОИСК(ПОВТОР("С";20);A2)	Подцепочка из двадцати символов "С" не найдена.
15	#ЗНАЧ!	ПОИСК(ПОВТОР("С";15);A2)	Подцепочка из пятнадцати символов "С" тоже не найдена.
16	763	ПОИСК(ПОВТОР("С";14);A2)	Подцепочка из четырнадцати символов "С" начинается с позиции 763. Т.е. она входит в заданную цепочку.
17	<b>Ответ:</b>	<b>14</b>	Максимальная длина подцепочки из символов "С"

### ***Задачи для тренировки:***

- 22) **(А.М. Кабанов)** В текстовом файле **k7a-2.txt** находится цепочка из символов латинского алфавита A, B, C, D, E, F. Найдите длину самой длинной подцепочки, состоящей из символов A, C, D (в произвольном порядке).
- 23) **(А.М. Кабанов)** В текстовом файле **k7a-3.txt** находится цепочка из символов латинского алфавита A, B, C, D, E, F. Найдите длину самой длинной подцепочки, состоящей из символов A, B, E, F (в произвольном порядке).
- 24) **(А.М. Кабанов)** В текстовом файле **k7a-4.txt** находится цепочка из символов латинского алфавита A, B, C, D, E, F. Найдите длину самой длинной подцепочки, не содержащей символа D.
- 25) **(А.М. Кабанов)** В текстовом файле **k7a-5.txt** находится цепочка из символов латинского алфавита A, B, C, D, E, F. Найдите длину самой длинной подцепочки, не содержащей символов C и F.
- 26) **(А.М. Кабанов)** В текстовом файле **k7a-6.txt** находится цепочка из символов латинского алфавита A, B, C, D, E, F. Найдите длину самой длинной подцепочки, не содержащей гласных букв.
- 27) **(А.М. Кабанов)** В текстовом файле **k7b-1.txt** находится цепочка из символов латинского алфавита A, B, C, D, E. Найдите максимальную длину цепочки вида EABEABEABE... (состоящей из фрагментов EAB, последний фрагмент может быть неполным).
- 28) **(А.М. Кабанов)** В текстовом файле **k7b-2.txt** находится цепочка из символов латинского алфавита A, B, C, D, E, F. Найдите максимальную длину цепочки вида DBACDBACDBAC.... (состоящей из фрагментов DBAC, последний фрагмент может быть неполным).
- 29) **(А.М. Кабанов)** В текстовом файле **k7b-3.txt** находится цепочка из символов латинского алфавита A, B, C, D, E, F. Найдите максимальную длину цепочки вида BAFEBAFEBAFE... (состоящей из фрагментов BAFE, последний фрагмент может быть неполным).
- 30) **(А.М. Кабанов)** В текстовом файле **k7b-4.txt** находится цепочка из символов латинского алфавита A, B, C, D, E, F. Найдите максимальную длину цепочки вида EBCFEBCFEBCF.... (состоящей из фрагментов EBCF, последний фрагмент может быть неполным).
- 31) **(А.М. Кабанов)** В текстовом файле **k7b-5.txt** находится цепочка из символов латинского алфавита A, B, C, D, E, F. Найдите максимальную длину цепочки вида CACACA.... (состоящей из фрагментов CA, последний фрагмент может быть неполным).
- 32) **(А.М. Кабанов)** В текстовом файле **k7b-6.txt** находится цепочка из символов латинского алфавита A, B, C, D, E, F. Найдите максимальную длину цепочки вида DAFDAFDAF.... (состоящей из фрагментов DAF, последний фрагмент может быть неполным).
- 33) **(А.М. Кабанов)** В текстовом файле **k7c-1.txt** находится цепочка из символов латинского алфавита A, B, C, D, E. Найдите количество цепочек длины 3, удовлетворяющих следующим условиям:
- 1-й символ – один из символов B, C или D;
  - 2-й символ – один из символов B, D, E, который не совпадает с первым;
  - 3-й символ – один из символов B, C, E, который не совпадает со вторым.
- 34) **(А.М. Кабанов)** В текстовом файле **k7c-2.txt** находится цепочка из символов латинского алфавита A, B, C, D, E, F. Найдите количество цепочек длины 3, удовлетворяющих следующим условиям:
- 1-й символ – один из A, C, E;
  - 2-й символ – один из A, D, F, который не совпадает с первым;
  - 3-й символ – один из A, B, F, который не совпадает со вторым.
- 35) **(А.М. Кабанов)** В текстовом файле **k7c-3.txt** находится цепочка из символов латинского алфавита A, B, C, D, E, F. Найдите количество цепочек длины 3, удовлетворяющих следующим условиям:
- 2-й символ – один из B, D, E;

- 3-й символ – один из А, С, D, который не совпадает со вторым;
  - 1-й символ – совпадает с третьим.
- 36) (А.М. Кабанов) В текстовом файле **k7c-4.txt** находится цепочка из символов латинского алфавита А, В, С, D, Е, F. Найдите количество цепочек длины 3, удовлетворяющих следующим условиям:
- 3-й символ – один из С, D, F;
  - 1-й символ – один из А, D, F, который не совпадает с третьим;
  - 2-й символ – один из С, D, F, который не совпадает с третьим.
- 37) (А.М. Кабанов) В текстовом файле **k7c-5.txt** находится цепочка из символов латинского алфавита А, В, С, D, Е, F. Найдите количество цепочек длины 5, в которых соседние символы не совпадают.
- 38) (А.М. Кабанов) В текстовом файле **k7c-6.txt** находится цепочка из символов латинского алфавита А, В, С, D, Е, F. Найдите количество цепочек длины 3, в которых символы не совпадают.
- 39) (Б.С. Михлин) В текстовом файле **k7-m1.txt** находится цепочка из прописных (заглавных) символов латинского алфавита А, В, С. Найдите длину самой короткой подцепочки, состоящей из символов С (С-подцепочки). В ответе через пробел укажите: длину найденной подцепочки (если С-подцепочек нет, то 0), количество С-подцепочек и длину исходной цепочки.
- 40) (Б.С. Михлин) В текстовом файле **k7-m2.txt** находится цепочка из прописных (заглавных) символов латинского алфавита А, В, С. Найдите длину самой длинной подцепочки, состоящей из символов С (С-подцепочки). В ответе через пробел укажите: длину найденной подцепочки (если С-подцепочек нет, то 0), количество С-подцепочек и длину исходной цепочки.
- 41) (Б.С. Михлин) В текстовом файле **k7-m3.txt** находится цепочка из прописных (заглавных) символов латинского алфавита А, В, С. Найдите все подцепочки, состоящие из символов С (С-подцепочки) длиной не более четырех. В ответе через пробел укажите: порядковый номер найденной подцепочки (начиная с единицы) при проходе по исходной цепочке слева направо, длину подцепочки и саму подцепочку, заменив в ней, начиная со второго символа «С», большие «С» на «с» строчные (маленькие). Гарантируется, что в исходной цепочке есть С-подцепочки.
- 42) (Б.С. Михлин) В текстовом файле **k7-m4.txt** находится цепочка из прописных (заглавных) символов латинского алфавита А, В, С. Найдите все подцепочки, состоящие из символов С (С-подцепочки) длиной не менее шести. В ответе через пробел укажите: порядковый номер найденной подцепочки (начиная с единицы) при проходе по исходной цепочке СПРАВА НАЛЕВО, ее длину и саму подцепочку, заменив в ней все символы «С» слева от правого символа «С» на «с» строчное (маленькое). Гарантируется, что в исходной цепочке есть С-подцепочки.
- 43) (Б.С. Михлин) В текстовом файле **k7-m5.txt** находится цепочка из прописных (заглавных) символов латинского алфавита А, В, С. В исходной цепочке замените все найденные С-подцепочки на подцепочки, содержащие длину текущей С-подцепочки с последующей текущей С-подцепочкой с замененными символами «С» большими на «с» маленькие. В ответе в трех строчках выведите:
- 1) количество С-подцепочек;
  - 2) левые 15 символов, пробел и правые 15 символов исходной цепочки;
  - 3) левые 15 символов, пробел и правые 15 символов преобразованной цепочки.
- 44) (Б.С. Михлин) В текстовом файле **k7-m6.txt** находится цепочка из прописных (заглавных) символов латинского алфавита А, В, С. В исходной цепочке замените все найденные С-подцепочки на подцепочки, содержащие порядковый номер (начиная с единицы) текущей С-подцепочки с последующей текущей С-подцепочкой в которой символы «С», начиная со второго, заменены на восклицательные знаки («!»). В ответе в трех строчках выведите:
- 1) количество С-подцепочек;



- 2) левые 15 символов, пробел и правые 15 символов исходной цепочки;
- 3) левые 15 символов, пробел и правые 15 символов преобразованной цепочки.
- 45) **(Б.С. Михлин)** В текстовом файле **k7-m7.txt** находится цепочка из прописных (заглавных) символов латинского алфавита A, B, C. В исходной цепочке все найденные C-подцепочки переместите в начало исходной цепочки и перед ними поставьте суммарную длину C-подцепочек, а после произведение длин C-подцепочек. Гарантируется, что в исходной цепочке есть C-подцепочки. В ответе в трех строчках выведите:
- 1) количество C-подцепочек;
  - 2) левые 35 символов исходной цепочки;
  - 3) левые 35 символов преобразованной цепочки.
- 46) **(Б.С. Михлин)** В текстовом файле **k7-m21.txt** находится цепочка из прописных (заглавных) символов латинского алфавита A, B, C, D, E, F. Найдите количество подцепочек из трёх различных символов, в которых символы идут в алфавитном порядке, и индекс первой буквы последней найденной подцепочки (первый символ исходной цепочки имеет индекс 0).  
Например, у цепочки ABCDF таких подцепочек три: ABC, BCD и CDF, а индекс первой буквы последней найденной подцепочки (CDF) два и, следовательно, ответ: 3 2.
- 47) **(Б.С. Михлин)** В текстовом файле **k7-m22.txt** находится цепочка из прописных (заглавных) символов латинского алфавита A, B, C, D, E, F. Найдите количество подцепочек из трех символов, в которых символы идут в обратном алфавитном порядке и индекс первой буквы последней найденной подцепочки (первый символ исходной цепочки имеет индекс 0).  
Например, у цепочки FDCBA таких подцепочек три: FDC, DCB и CBA, а индекс первой буквы последней найденной подцепочки (CBA) два и, следовательно, ответ: 3 2.
- 48) **(Б.С. Михлин)** В текстовом файле **k7-m23.txt** находится цепочка из прописных (заглавных) символов латинского алфавита A, B, C, D, E, F. Найдите количество подцепочек из трех символов, в которых символы идут не в убывающем алфавитном порядке и индекс первой буквы последней найденной подцепочки (первый символ исходной цепочки имеет индекс 0).  
Например, у цепочки ABCFF таких подцепочек три: ABC, BCF и CFF, а индекс первой буквы последней найденной подцепочки (CFF) два и, следовательно, ответ: 3 2.
- 49) **(Б.С. Михлин)** В текстовом файле **k7-m24.txt** находится цепочка из прописных (заглавных) символов латинского алфавита A, B, C, D, E, F. Найдите количество подцепочек из трех символов, в которых символы идут не в возрастающем алфавитном порядке и индекс первой буквы последней найденной подцепочки (первый символ исходной цепочки имеет индекс 0).  
Например, у цепочки FFCBA таких подцепочек три: FFC, FCB и CBA, а индекс первой буквы последней найденной подцепочки (CBA) два и, следовательно, ответ: 3 2.
- 50) **(Б.С. Михлин)** В текстовом файле **k7-m25.txt** находится цепочка из прописных (заглавных) символов латинского алфавита A, B, C, D, E, F. Найдите количество подцепочек из трех символов, в которых средний символ ближе к концу алфавита, чем символ слева и справа от него, а также найдите индекс первой буквы последней найденной подцепочки (первый символ исходной цепочки имеет индекс 0). Например, у цепочки ACBFAED таких подцепочек три: ACB, BFA и AED, а индекс первой буквы последней найденной подцепочки (AED) четыре и, следовательно, ответ: 3 4.
- 51) **(Б.С. Михлин)** В текстовом файле **k7-m26.txt** находится цепочка из прописных (заглавных) символов латинского алфавита A, B, C, D, E, F. Найдите количество подцепочек из трех символов, в которых средний символ ближе к началу алфавита, чем символ слева и справа от него, а также найдите индекс первой буквы последней найденной подцепочки (первый символ исходной цепочки имеет индекс 0).  
Например, у цепочки FABACAE таких подцепочек три: FAB, BAC и CAE, а индекс первой буквы последней найденной подцепочки (CAE) четыре и, следовательно, ответ: 3 4.







одинаковой длины, нужно взять первую из них. Выведите сначала символ, из которого строится эта подцепочка, а затем через пробел – длину этой подцепочки.

- 71) В текстовом файле **k8-100.txt** находится цепочка из символов, в которую могут входить заглавные буквы латинского алфавита A...Z и десятичные цифры. Найдите длину самой длинной подцепочки, состоящей из одинаковых символов. Если в файле несколько подходящих цепочек одинаковой длины, нужно взять первую из них. Выведите сначала символ, из которого строится эта подцепочка, а затем через пробел – длину этой подцепочки.
- 72) В текстовом файле **k8-0.txt** находится цепочка из символов, в которую могут входить заглавные буквы латинского алфавита A...Z и десятичные цифры. Найдите длину самой длинной подцепочки, состоящей из одинаковых символов. **Для каждой цепочки максимальной длины** выведите в отдельной строке сначала символ, из которого строится эта цепочка, а затем через пробел – длину этой цепочки.
- 73) В текстовом файле **k8-4.txt** находится цепочка из символов, в которую могут входить заглавные буквы латинского алфавита A...Z и десятичные цифры. Найдите длину самой длинной подцепочки, состоящей из одинаковых символов. **Для каждой цепочки максимальной длины** выведите в отдельной строке сначала символ, из которого строится эта цепочка, а затем через пробел – длину этой цепочки.
- 74) В текстовом файле **k8-6.txt** находится цепочка из символов, в которую могут входить заглавные буквы латинского алфавита A...Z и десятичные цифры. Если в файле несколько подходящих цепочек одинаковой длины, нужно взять первую из них. **Для каждой цепочки максимальной длины** выведите в отдельной строке сначала символ, из которого строится эта цепочка, а затем через пробел – длину этой цепочки.
- 75) В текстовом файле **k8-12.txt** находится цепочка из символов, в которую могут входить заглавные буквы латинского алфавита A...Z и десятичные цифры. Найдите длину самой длинной подцепочки, состоящей из одинаковых символов. **Для каждой цепочки максимальной длины** выведите в отдельной строке сначала символ, из которого строится эта цепочка, а затем через пробел – длину этой цепочки.
- 76) В текстовом файле **k8-18.txt** находится цепочка из символов, в которую могут входить заглавные буквы латинского алфавита A...Z и десятичные цифры. Найдите длину самой длинной подцепочки, состоящей из одинаковых символов. **Для каждой цепочки максимальной длины** выведите в отдельной строке сначала символ, из которого строится эта цепочка, а затем через пробел – длину этой цепочки.
- 77) Текстовый файл **k8-1.txt** состоит не более чем из  $10^6$  символов. Определите максимальное количество идущих подряд символов, среди которых каждые два соседних различны.
- 78) Текстовый файл **k8-2.txt** состоит не более чем из  $10^6$  символов. Определите максимальное количество идущих подряд символов, среди которых каждые два соседних различны.
- 79) Текстовый файл **k8-3.txt** состоит не более чем из  $10^6$  символов. Определите максимальное количество идущих подряд символов, среди которых каждые два соседних различны.
- 80) Текстовый файл **k8-4.txt** состоит не более чем из  $10^6$  символов. Определите максимальное количество идущих подряд символов, среди которых каждые два соседних различны.
- 81) Текстовый файл **k8-5.txt** состоит не более чем из  $10^6$  символов. Определите максимальное количество идущих подряд символов, среди которых каждые два соседних различны.
- 82) Текстовый файл **k8-6.txt** состоит не более чем из  $10^6$  символов. Определите максимальное количество идущих подряд символов, среди которых каждые два соседних различны.
- 83) Текстовый файл **k8-7.txt** состоит не более чем из  $10^6$  символов. Определите максимальное количество идущих подряд символов, среди которых каждые два соседних различны.

- 84) Текстовый файл **k8-8 .txt** состоит не более чем из  $10^6$  символов. Определите максимальное количество идущих подряд символов, среди которых каждые два соседних различны.
- 85) Текстовый файл **k8-9 .txt** состоит не более чем из  $10^6$  символов. Определите максимальное количество идущих подряд символов, среди которых каждые два соседних различны.
- 86) Текстовый файл **k8-10 .txt** состоит не более чем из  $10^6$  символов. Определите максимальное количество идущих подряд символов, среди которых каждые два соседних различны.

В следующих задачах под **числом** подразумевается последовательность цифр, ограниченная другими символами (не цифрами).

- 87) (П.Е. Финкель, г. Тимашевск) Текстовый файл **24-1 .txt** состоит не более чем из  $10^6$  символов. Определите максимальное нечётное число, записанное в этом файле.
- 88) (П.Е. Финкель, г. Тимашевск) Текстовый файл **24-1 .txt** состоит не более чем из  $10^6$  символов. Определите минимальное нечётное число, записанное в этом файле.
- 89) (П.Е. Финкель, г. Тимашевск) Текстовый файл **24-1 .txt** состоит не более чем из  $10^6$  символов. Определите максимальное чётное число, записанное в этом файле.
- 90) (П.Е. Финкель, г. Тимашевск) Текстовый файл **24-1 .txt** состоит не более чем из  $10^6$  символов. Определите минимальное чётное число, записанное в этом файле.
- 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.
- 92) (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 чётных цифр.

**Возрастающей подпоследовательностью** будем называть последовательность символов, расположенных в порядке увеличения их номера в кодовой таблице символов ASCII.

**Убывающей подпоследовательностью** будем называть последовательность символов, расположенных в порядке уменьшения их номера в кодовой таблице символов ASCII.

- 93) (В.Н. Шубинкин, г. Казань) Текстовый файл **24 .txt** содержит последовательность из строчных и заглавных букв английского алфавита и цифр, всего не более  $10^6$  символов. Определите длину наибольшей возрастающей подпоследовательности.
- 94) (В.Н. Шубинкин, г. Казань) Текстовый файл **24-1 .txt** содержит последовательность из строчных и заглавных букв английского алфавита и цифр, всего не более  $10^6$  символов. Определите длину наибольшей возрастающей подпоследовательности.
- 95) (В.Н. Шубинкин, г. Казань) Текстовый файл **24-2 .txt** содержит последовательность из строчных и заглавных букв английского алфавита и цифр, всего не более  $10^6$  символов. Определите длину наибольшей возрастающей подпоследовательности.
- 96) (В.Н. Шубинкин, г. Казань) Текстовый файл **24-3 .txt** содержит последовательность из строчных и заглавных букв английского алфавита и цифр, всего не более  $10^6$  символов. Определите длину наибольшей возрастающей подпоследовательности.
- 97) (В.Н. Шубинкин, г. Казань) Текстовый файл **24-4 .txt** содержит последовательность из строчных и заглавных букв английского алфавита и цифр, всего не более  $10^6$  символов. Определите длину наибольшей возрастающей подпоследовательности.
- 98) (В.Н. Шубинкин, г. Казань) Текстовый файл **24 .txt** содержит последовательность из строчных и заглавных букв английского алфавита и цифр, всего не более  $10^6$  символов. Запишите в ответе наибольшую возрастающую подпоследовательность. Если таких последовательностей несколько, запишите первую из них.
- 99) (В.Н. Шубинкин, г. Казань) Текстовый файл **24-1 .txt** содержит последовательность из строчных и заглавных букв английского алфавита и цифр, всего не более  $10^6$  символов. Запишите в ответе





Нумерация символов начинается с 1. Если таких последовательностей несколько, используйте первую из них.

- 122) (В.Н. Шубинкин, г. Казань) Текстовый файл **24-4.txt** содержит последовательность из строчных и заглавных букв английского алфавита и цифр, всего не более  $10^6$  символов. Запишите в ответе номер символа, с которого начинается наибольшая убывающая подпоследовательность. Нумерация символов начинается с 1. Если таких последовательностей несколько, используйте первую из них.
- 123) (В.Н. Шубинкин, г. Казань) Текстовый файл содержит последовательность из строчных и заглавных букв английского алфавита и цифр. Всего не более  $10^6$  символов. Назовём локальным максимумом символ, номер которого в кодовой таблице больше номеров предыдущего и последующего символов. Самый первый и самый последний символ не являются локальными максимумами. Определить наибольшее расстояние между двумя соседними локальными максимумами. Расстоянием между элементами будем считать разность их индексов. Исходные данные записаны в файле **24-1.txt**.
- 124) (В.Н. Шубинкин, г. Казань) Текстовый файл содержит последовательность из строчных и заглавных букв английского алфавита и цифр. Всего не более  $10^6$  символов. Назовём локальным минимумом символ, номер которого в кодовой таблице меньше номеров предыдущего и последующего символов. Самый первый и самый последний символ не являются локальными минимумами. Определить наибольшее расстояние между двумя соседними локальными минимумами. Расстоянием между элементами будем считать разность их индексов. Исходные данные записаны в файле **24-1.txt**.
- 125) (К. Амеличев) Текстовый файл **24-5.txt** содержит последовательность из символов « ( » и « ) », всего не более  $10^6$  символов. Определить количество пар скобок « ( ) » в этом файле.
- 126) (К. Амеличев) Текстовый файл **24-5.txt** содержит последовательность из символов « ( » и « ) », всего не более  $10^6$  символов. Определить максимальное количество подряд идущих пар скобок « ( ) » в этом файле.
- 127) (К. Амеличев) Текстовый файл **24-5.txt** содержит последовательность из символов « ( » и « ) », всего не более  $10^6$  символов. Определить максимальное количество подряд идущих открывающих скобок « ( » в этом файле.
- 128) (К. Амеличев) Текстовый файл **24-5.txt** содержит последовательность из символов « ( » и « ) », всего не более  $10^6$  символов. Определить максимальное количество подряд идущих закрывающих скобок « ) » в этом файле.
- 129) (К. Амеличев) Текстовый файл **24-5.txt** содержит последовательность из символов « ( » и « ) », всего не более  $10^6$  символов. Определить, каким по счёту символом от начала файла окажется 10000-я открывающая скобка « ( ».
- 130) (К. Амеличев) Текстовый файл **24-5.txt** содержит последовательность из символов « ( » и « ) », всего не более  $10^6$  символов. Определить, каким по счёту символом от начала файла окажется 10000-я закрывающая скобка « ) ».
- 131) (К. Амеличев) Текстовый файл **24-5.txt** содержит последовательность из символов « ( » и « ) », всего не более  $10^6$  символов. Определить, с какого по счёту символа от начала файла начинается 10000-я пара скобок « ( ) ».
- 132) (Е. Джобс) Текстовый файл **24-J1.txt** состоит не более чем из  $10^6$  кириллических символов К, О, Т. Определите максимальное количество подряд идущих комбинаций КОТ.
- 133) (Е. Джобс) Текстовый файл **24-J2.txt** состоит не более чем из  $10^6$  символов F, A, I, L. Определите максимальное количество подряд идущих одинаковых букв.
- 134) (Е. Джобс) Текстовый файл **24-J3.txt** состоит не более чем из  $10^6$  символов I, K, O, T. Сколько раз встречаются комбинации «ТІК» и «ТОК».



- 135) (Е. Джобс) Текстовый файл **24-J4.txt** состоит не более чем из  $10^6$  символов J, O, B, S. Сколько раз встречаются комбинации «BOSS» при этом до и после этого слова нет символа «J». Например, комбинации «JBOSS», «BOSSJ» и «JBOSSJ» не должны учитываться.
- 136) (Е. Джобс) Текстовый файл **24-J5.txt** состоит не более чем из  $10^6$  символов S, T, O, C, K. Сколько раз встречается комбинация «OCK», не являющаяся при этом частью комбинации «STOCK».
- 137) Текстовый файл **24-s1.txt** 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.
- 138) Текстовый файл **24-s1.txt** состоит не более чем из  $10^6$  заглавных латинских букв (A..Z). Текст разбит на строки различной длины. Определите количество строк, в которых буква K встречается чаще, чем буква U.
- 139) Текстовый файл **24-s1.txt** состоит не более чем из  $10^6$  заглавных латинских букв (A..Z). Текст разбит на строки различной длины. Определите количество строк, в которых буква S встречается столько же раз, сколько и буква X.
- 140) Текстовый файл **24-s1.txt** состоит не более чем из  $10^6$  заглавных латинских букв (A..Z). Текст разбит на строки различной длины. Определите количество строк, в которых комбинация YZ встречается больше одного раза.
- 141) Текстовый файл **24-s1.txt** 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.
- 142) Текстовый файл **24-s1.txt** 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 A\*R, где звёздочка обозначает любой символ.
- 143) Текстовый файл **24-s1.txt** 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 Z\*RO, где звёздочка обозначает любой символ.
- 144) (Е. Джобс) Текстовый файл **24-j6.txt** состоит не более чем из  $10^6$  десятичных цифр. Восходящей последовательностью называется последовательность цифр, в которой каждая цифра меньше следующей за ней. Например, в последовательности 7238903278 три таких последовательности – 2389, 03 и 278. Длиной последовательности называется количество входящих в нее цифр. Определите сколько в файле восходящих последовательностей длиной 5, не входящих в восходящие последовательности большей длины.
- 145) (Е. Джобс) Текстовый файл **24-j7.txt** состоит не более чем из  $10^6$  десятичных цифр. Найдите максимальную длину последовательности, которая состоит из цифр одинаковой четности. Например, в последовательности 1533244622185452354, 5 последовательностей с нечетными цифрами – 1533, 1, 5, 5, 35 – и 5 с четными – 244622, 8, 4, 2, 4. Следовательно, искомая последовательность – 244622. В качестве ответа укажите максимальную длину найденной последовательности.
- 146) (Е. Джобс) Текстовый файл **24-j8.txt** состоит не более чем из  $10^6$  десятичных цифр. Найдите максимальную длину последовательности, каждые две соседние цифры в которой в сумме дают значение не меньшее 10. Например, в последовательности 1567543853 есть две такие последовательности: 5675 и 385. В качестве ответа укажите максимальную длину найденной последовательности.
- 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 А. Если таких символов несколько, нужно вывести тот, который стоит раньше в алфавите. Например, в тексте ABCAABADDD после буквы А два раза стоит В, по одному разу – А и D. Для этого текста ответом будет В2.
- 148) Текстовый файл **24-s2.txt** содержит только заглавные буквы латинского алфавита (ABC...Z). 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. Если таких символов несколько, нужно вывести тот, который стоит раньше в алфавите. Например, в тексте XBCXXBXDDDD после буквы X два раза стоит В, по одному разу – X и D. Для этого текста ответом будет В2.
- 149) Текстовый файл **24-s2.txt** содержит только заглавные буквы латинского алфавита (ABC...Z). Определите символ, который чаще всего встречается в файле между буквами X и Z, так что X стоит слева от него, а Z – справа. В ответе запишите сначала этот символ, а потом сразу (без разделителя) сколько раз он встретился между буквами X и Z. Если таких символов несколько, нужно вывести тот, который стоит раньше в алфавите. Например, в тексте XBZCXXZXBZXDZDD между буквами X и Z два раза стоит В, по одному разу – X и D. Для этого текста ответом будет В2.
- 150) Текстовый файл **24-s2.txt** содержит только заглавные буквы латинского алфавита (ABC...Z). Определите символ, который чаще всего встречается в файле между буквами А и С, так что А стоит слева от него, а С – справа. В ответе запишите сначала этот символ, а потом сразу (без разделителя) сколько раз он встретился между буквами А и С. Если таких символов несколько, нужно вывести тот, который стоит раньше в алфавите. Например, в тексте ABCCAACZABCADCDD между буквами А и С два раза стоит В, по одному разу – А и D. Для этого текста ответом будет В2.
- 151) (**Е. Джобс**) Текстовый файл **24-j9.txt** состоит не более чем из  $10^6$  символов английского алфавита. Определите количество палиндромов (последовательностей, которые читаются в обе стороны одинаково) длиной 5 символов.
- 152) (**Е. Джобс**) Текстовый файл **24-j9.txt** состоит не более чем из  $10^6$  символов. Симметричной парой называют два одинаковых символа, которые расположены на одинаковом удалении от концов строки. Сколько пар символов в строке являются симметричными? Например, в строке **adcdeefcba** три симметричных пары – **aa**, **cc** и **ee**.
- 153) (**А. Кабанов**) Текстовый файл **24-153.txt** содержит строку из заглавных букв А, В, С, D, Е, F, всего не более  $10^6$  символов. D-подстроками назовём последовательности идущих подряд символов D, ограниченные иными символами и/или границами строки. Определите минимальную длину D-подстроки.
- 154) (**А. Кабанов**) Текстовый файл **24-153.txt** содержит строку из заглавных букв А, В, С, D, Е, F, всего не более  $10^6$  символов. DD-подстроками назовём последовательности символов А, В, С, Е, F, ограниченные символами D (граничные символы входят в подстроку). Определите минимальную длину DD-подстроки. Подстроки, состоящие из двух символов, не учитывать.
- 155) (**А. Кабанов**) Текстовый файл **24-153.txt** содержит строку из заглавных букв А, В, С, D, Е, F, всего не более  $10^6$  символов. AF-подстроками назовём последовательности символов А, В, С, D, Е, F, ограниченные в начале символом А, а в конце символом F (граничные символы входят в подстроку). Определите минимальную длину AF-подстроки. Подстроки, состоящие из двух символов, не учитывать.
- 156) (**А. Кабанов**) Текстовый файл **24-153.txt** содержит строку из заглавных букв А, В, С, D, Е, F, всего не более  $10^6$  символов. AF-подстроками назовём непустые последовательности идущих подряд символов А, В, С, D, Е, F, ограниченные в начале символом А, а в конце символом F (граничные символы входят в подстроку). Определите количество AF-подстрок длиной от 7 до 10 символов.

157) Текстовый файл **24-157.txt** содержит только заглавные буквы латинского алфавита (ABC...Z).

Определите символ, который чаще всего встречается в файле после двух одинаковых символов. Например, в тексте CCCBBAABAABCC есть комбинации CCC, CCB, BBA и AAB. Чаще всего – 2 раза – после двух одинаковых символов стоит B, в ответе для этого случая надо написать B2 (без пробелов и других разделителей). Если таких символов несколько, выведите тот, который стоит раньше в алфавите.

158) Текстовый файл **24-157.txt** содержит только заглавные буквы латинского алфавита (ABC...Z).

Определите символ, который чаще всего встречается в файле перед двумя одинаковыми символами. Например, в тексте CCBAABABVCCCC есть комбинации BAA, ABB, VCC и CCC. Чаще всего – 2 раза – перед двумя одинаковыми символами стоит B, в ответе для этого случая надо написать B2 (без пробелов и других разделителей). Если таких символов несколько, выведите тот, который стоит раньше в алфавите.

159) Текстовый файл **24-157.txt** содержит только заглавные буквы латинского алфавита (ABC...Z).

Определите символ, который чаще всего встречается в файле между двумя одинаковыми символами. Например, в тексте CCBAABABVCBC есть комбинации ABA, BAB, VCB и SVC. Чаще всего – 2 раза – между двумя одинаковыми символами стоит B, в ответе для этого случая надо написать B2 (без пробелов и других разделителей). Если таких символов несколько, выведите тот, который стоит раньше в алфавите.

160) Текстовый файл **24-s1.txt** состоит не более чем из  $10^6$  заглавных латинских букв (A..Z). Текст разбит на строки различной длины. Необходимо найти строку, содержащую наименьшее количество букв A (если таких строк несколько, надо взять ту, которая в файле встретилась раньше). Определите, какая буква встречается в этой строке чаще всего. Если таких букв несколько, надо взять ту, которая стоит последней в алфавите. Запишите в ответе эту букву, а затем – сколько раз она встречается во всем файле.

Пример. Исходный файл:

**ZZQAQ**  
**ZAVLAB**  
**KRAKTU**

В этом примере в первой и третьей строках по одной букве A, во второй – две. Берём первую строку, т.к. она находится в файле раньше. В этой строке чаще других встречаются буквы Z и Q (по два раза), выбираем букву Z, т. к. она позже стоит в алфавите. В ответе для этого примера надо записать Z3, так как во всех строках файла буква Z встречается 3 раза.

161) Текстовый файл **24-s1.txt** состоит не более чем из  $10^6$  заглавных латинских букв (A..Z). Текст разбит на строки различной длины. Необходимо найти строку, содержащую наибольшее количество букв Q (если таких строк несколько, надо взять ту, которая в файле встретилась позже). Определите, какая буква встречается в этой строке реже всего (но присутствует!). Если таких букв несколько, надо взять ту, которая стоит раньше в алфавите. Запишите в ответе эту букву, а затем – сколько раз она встречается во всем файле.

Пример. Исходный файл:

**ZZQAQV**  
**QAVQAB**  
**BAQTUB**

В этом примере в первой и второй строках по две буквы Q, в третьей – одна. Берём вторую строку, т.к. она стоит в файле позже. В этой строке реже других встречаются буквы V и B (по одному разу), выбираем букву B, т. к. она раньше стоит в алфавите. В ответе для этого примера надо записать B4, так как во всех строках файла буква B встречается 4 раза.



- 162) Текстовый файл **24-s1.txt** состоит не более чем из  $10^6$  заглавных латинских букв (A..Z). Текст разбит на строки различной длины. Необходимо найти строку, содержащую наименьшее ненулевое количество пар соседних букв, которые стоят в таком же порядке и в алфавите (например, АВ, ВС, CD и т.д.). Если таких строк несколько, надо взять ту, которая в файле встретилась раньше. Определите, какая буква встречается в этой строке чаще всего. Если таких букв несколько, надо взять ту, которая стоит последней в алфавите. Запишите в ответе эту букву, а затем – сколько раз она встречается во всем файле.

Пример. Исходный файл:

**ZZQABA**  
**ZALMAC**  
**KRAKUT**

В этом примере в первой и второй строках по одной подходящей паре (АВ и LM), в третьей таких пар нет. Берём первую строку, т.к. она находится в файле раньше. В этой строке чаще других встречаются буквы Z и A (по два раза), выбираем букву Z, т. к. она позже стоит в алфавите. В ответе для этого примера надо записать Z3, так как во всех строках файла буква Z встречается 3 раза.

- 163) Текстовый файл **24-s1.txt** состоит не более чем из  $10^6$  заглавных латинских букв (A..Z). Текст разбит на строки различной длины. Необходимо найти строку, содержащую наибольшее количество пар соседних букв, которые стоят в таком же порядке и в алфавите (например, АВ, ВС, CD и т.д.; в цепочке ABC две таких пары). Если таких строк несколько, надо взять ту, которая в файле встретилась позже. Определите, какая буква встречается в этой строке реже всего (но присутствует!). Если таких букв несколько, надо взять ту, которая стоит раньше в алфавите. Запишите в ответе эту букву, а затем – сколько раз она встречается во всем файле.

Пример. Исходный файл:

**ZCQABA**  
**ZALMAC**  
**CRACUT**

В этом примере в первой и второй строках по одной подходящей паре (АВ и LM), в третьей таких пар нет. Берём вторую строку, т.к. она позже встречается в файле. В этой строке реже других встречаются буквы Z и C (по одному разу), выбираем букву C, т. к. она раньше стоит в алфавите. В ответе для этого примера надо записать C4, так как во всех строках файла буква C встречается 4 раза.

- 164) Текстовый файл **24-164.txt** состоит не более чем из  $10^6$  заглавных латинских букв (A..Z). Текст разбит на строки различной длины. Необходимо найти строку, содержащую самую длинную цепочку стоящих подряд одинаковых букв. Если таких строк несколько, надо взять ту, которая в файле встретилась раньше. Определите, какая буква встречается в этой строке реже всего (но присутствует!). Если таких букв несколько, надо взять ту, которая стоит последней в алфавите. Запишите в ответе эту букву, а затем – сколько раз она встречается во всем файле.

Пример. Исходный файл:

**ZZQABA**  
**ZALAAC**  
**QRAQUT**

В этом примере в первой и второй строках наибольшая длина цепочек одинаковых букв равна 2 (ZZ в первой строке, AA во второй), в третьей – 1. Берём первую строку, т.к. она находится в файле раньше. В этой строке реже других встречаются буквы Q и В (по разу), выбираем букву Q, т. к. она позже стоит в алфавите. В ответе для этого примера надо записать Q3, так как во всех строках файла буква Q встречается 3 раза.

- 165) Текстовый файл **24-164.txt** состоит не более чем из  $10^6$  заглавных латинских букв (A..Z). Текст разбит на строки различной длины. Необходимо найти строку, содержащую самую длинную цепочку стоящих подряд одинаковых букв. Если таких строк несколько, надо взять ту, которая в файле встретилась раньше. Определите, какая буква встречается в этой строке чаще всего. Если таких букв несколько, надо взять ту, которая стоит раньше в алфавите. Запишите в ответе эту букву, а затем – сколько раз она встречается во всем файле.

Пример. Исходный файл:

**ZZQABA**

**ZALAAC**

**QRAQUT**

В этом примере в первой и второй строках наибольшая длина цепочек одинаковых буквы равна 2 (ZZ в первой строке, AA во второй), в третьей – 1. Берём первую строку, т.к. она находится в файле раньше. В этой строке чаще других встречаются буквы Z и A (по 2 раза), выбираем букву A, т.к. она стоит раньше в алфавите. В ответе для этого примера надо записать A6, так как во всех строках файла буква A встречается 6 раз.

- 166) Текстовый файл **24-164.txt** состоит не более чем из  $10^6$  заглавных латинских букв (A..Z). Файл разбит на строки различной длины. В строках, содержащих менее 15 букв G, нужно определить и вывести максимальное расстояние между одинаковыми буквами в одной строке.

Пример. Исходный файл:

**VOVA**

**ZAGALG**

**QRAGQT**

В этом примере во всех строках меньше 15 букв G. Самое большое расстояние между одинаковыми буквами – в третьей строке между буквами Q, расположенными в строке на 1-й и 5-й позициях. В ответе для данного примера нужно вывести число 4.

- 167) Текстовый файл **24-164.txt** состоит не более чем из  $10^6$  заглавных латинских букв (A..Z). Файл разбит на строки различной длины. В строках, содержащих менее 20 букв E, нужно определить и вывести максимальное расстояние между одинаковыми буквами в одной строке.

Пример. Исходный файл:

**VOVA**

**ZAFALE**

**QRAEQT**

В этом примере во всех строках меньше 20 букв E. Самое большое расстояние между одинаковыми буквами – в третьей строке между буквами Q, расположенными в строке на 1-й и 5-й позициях. В ответе для данного примера нужно вывести число 4.

- 168) (**Е. Джобс**) Текстовый файл **24-168.txt** состоит не более чем из  $10^6$  заглавных латинских букв (A..Z). Определите наибольшую длину последовательности из трех различных символов, расположенных в порядке неубывания. Например, для строки AABBAABCCDDDEFFGF искомая последовательность – BBCCDDDD, её длина – 7.

- 169) Текстовый файл **24-169.txt** состоит не более чем из  $10^6$  заглавных символов X, Y и Z.

Определите максимальную длину цепочки символов, состоящей из повторяющихся фрагментов XYZ. Цепочка должна начинаться с символа X и заканчиваться символом Z. Например, для строки ZZZXYZXYZXZZZ длина цепочки равна 6: --XYZ+XYZ--.

- 170) (**А. Богданов**) Текстовый файл **24-169.txt** состоит не более чем из  $10^6$  заглавных символов X, Y и Z. Определите максимальную длину цепочки символов, состоящей из повторяющихся фрагментов XYZ. Цепочка может начинаться и заканчиваться любым символом из XYZ, но внутри

цепочки порядок строго определен. Например, для строки ZZZXYZXYZXZZZ длина цепочки равна 8: --Z+XYZ+XYZ+X--, где цепочка начинается с Z и заканчивается X.

- 171) Текстовый файл **24-171.txt** состоит не более чем из  $10^6$  заглавных латинских букв. Файл разбит на строки различной длины. Определите максимальную длину цепочки символов, состоящей из повторяющихся фрагментов XYZ. Цепочка должна начинаться с символа X и заканчиваться символом Z. Например, для строки SAZZXYZXYZXZQW длина цепочки равна 6: --XYZ+XYZ--.
- 172) **(А. Богданов)** Текстовый файл **24-171.txt** состоит не более чем из  $10^6$  заглавных латинских букв. Файл разбит на строки различной длины. Определите максимальную длину цепочки символов, состоящей из повторяющихся фрагментов XYZ. Цепочка может начинаться и заканчиваться любым символом из XYZ, но внутри цепочки порядок строго определен. Например, для строки SAZXYZXYZXZQW длина цепочки равна 8: --Z+XYZ+XYZ+X--, где цепочка начинается с Z и заканчивается X.
- 173) **(Е. Джобс)** Текстовый файл **24-173.txt** состоит не более чем из  $10^6$  букв из набора A, B, C, D, E, F. Найдите максимальную длину подстроки, в которой ни одна тройка символов не записана два раза подряд. Например, в искомой подстроке не может быть фрагмента ABCABC.