АСОУ
АКАДЕМИЯ СОЦИАЛЬНОГО УПРАВЛЕНИЯ

ЦНППМ
ЦЕНТР НЕПРЕРЫВНОГО ПОВЫШЕНИЯ ПЕДАГОГИЧЕСКОГО МАСТЕРСТВА

Вопросы методики подготовки к выполнению задания № 24 КЕГЭ по информатике и ИКТ

Перевозникова Маргарита Николаевна
учитель математики и информатики МОУ Евсеевской СОШ
г.о. Павловский Посад

АКАДЕМИЯ СОЦИАЛЬНОГО УПРАВЛЕНИЯ

ЦЕНТР НЕПРЕРЫВНОГО ПОВЫШЕНИЯ ПЕДАГОГИЧЕСКОГО МАСТЕРСТВА

Пример задачи №24 (в общем виде)

В текстовом файле *.txt находится цепочка из символов латинского алфавита A, B, C,… (могут быть другие, обозначим символ для общего случая N). Найдите длину самой длинной подцепочки, состоящей из символов N.

Чтение из файла (Python)

```
# однострочный файл
        f=open('*.txt')  #открываем файл для чтения
        s=f.read()       #считываем все символы из файла в символьную строку
          …
        f.close ()       #закрываем файл
# многострочный файл
        f=open('*.txt')  #открываем файл для чтения
        for s in f:      # циклом проходим по всем строкам этого файла, на каждом новом шаге этого цикла
                           будет считываться новая строка
          …
        f.close ()       #закрываем файл
```

Чтение из файла (PascalABC.NET в Windows)

```
# однострочный файл
    var s:string;
    begin
        assign(input, '*.txt');          // перенаправление потока ввода, программа будет «думать», что читает
                                            данные, введённые с клавиатуры (с консоли), а на самом деле эти данные
                                            будут прочитаны из текстового файла
        reset(input);                    // открываем файл для чтения
        readln(input,s);                    // считываем символы  из файла в символьную строку s
        …
       close(input);
   end.
# многострочный файл
         var s:string;
    begin
        assign(input, '*.txt');          // перенаправление потока ввода…
        reset(input);                    // открываем файл для чтения
        while not eof (input) do         // пока не достигли окончания файла,  выполняем чтение построчно
            begin
               readln(input,s);          // считываем символы  из файла в символьную строку s
               …                         // обработка в теле цикла
            end;
       close(input);
   end.
```

Используемые переменные

Для решения задачи используем следующие переменные:

curLen – длина текущей цепочки букв N;

maxLen – максимальная длина цепочки букв N на данный момент.

Переменным **curLen и maxLen** нужно присвоить начальные значения, зависящие от условия задачи.

```
curLen:=0;
maxLen:=0;
```

(в *PascalABC.NET)*

```
curLen = 0
maxLen= 0
```

(в Python)

Обработка строк

Встроенные функции

- определение длины строки

Length(s) (в *PascalABC.NET)*

len(s) *(в Python)*

- нахождение максимального значения

max (curLen, maxLen) *(в Python)*

Организуем цикл, используя который, *просматриваем поочередно все символы строки s*

for i:=1 to Length(s) (в PascalABC.NET)

```
for c in s:
```

или (в Python)

```
for i in range(len(s))
```

Обработка строк

Делаем проверку условия:

выясняем удовлетворяет ли очередной символ строки требуемому условию:

```
if (s[i] =‘N')  then
if c==‘N':
```

(в *PascalABC.NET)*

(в Python)

Обработка строк.

Если условие выполняется, например, это нужная нам буква N, то увеличиваем переменную **curLen** на 1 и запоминаем *новую максимальную длину* ***maxLen***:

```
curLen  := curLen+1;
if curLen>maxLen then
        maxLen := curLen;
```

(в *PascalABC.NET)*

```
curLen  += 1
maxLen = max(curLen, maxLen)
```

(в Python)

Если условие не выполняется, то переменная **curLen** становится вновь равна 0, то есть цепочка букв закончилась:

```
curLen  := 0;
```

(в *PascalABC.NET)*

```
curLen  = 0
```

(в Python)

Вывод результата

```
writeln(maxLen);
```
(в *PascalABC.NET)*

```
print(maxLen)
```
(в Python)

Программы для решения задачи 24 (поиск максимальной подстроки) в общем виде

АКАДЕМИЯ СОЦИАЛЬНОГО УПРАВЛЕНИЯ

ЦЕНТР НЕПРЕРЫВНОГО ПОВЫШЕНИЯ ПЕДАГОГИЧЕСКОГО МАСТЕРСТВА

Python

```
f=open('*.txt')
s=f.read ()
curLen = 0
maxLen= 0
for  i  in range(len (s)):
      if   s[i] =='N':
           curLen+= 1
           maxLen = max(curLen, maxLen)
     else:
           curLen=0
print(maxLen)
```

PascalABC.NET

```
var s: string;
var  curLen, maxLen, i: integer;
begin
assign(input,'*.txt');
readln(s);
curLen:=0;
maxLen:=0;
for  i:=1 to Length(s) do
     if (s[i]='N')  then
          begin
          curLen := curLen+1;
          if curLen > maxLen then
              maxLen = curLen
          end
     else
           curLen := 0;
writeln(maxLen);
end.
```

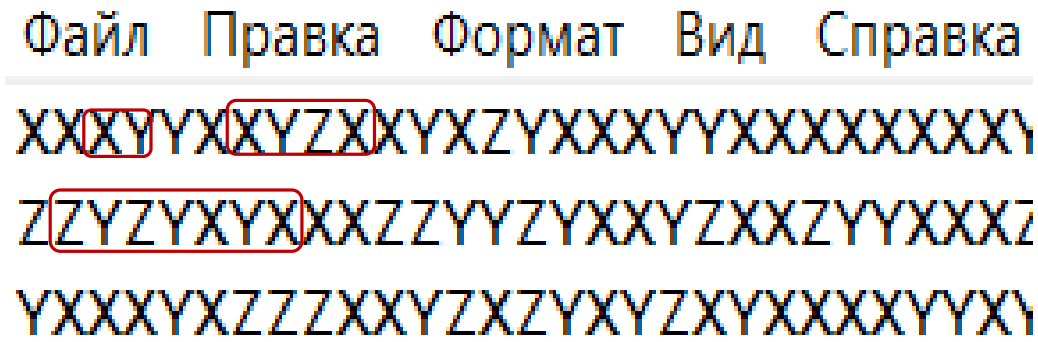
Демо версии задания № 24

ЕГЭ-2021

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**.

Для выполнения этого задания следует написать программу.

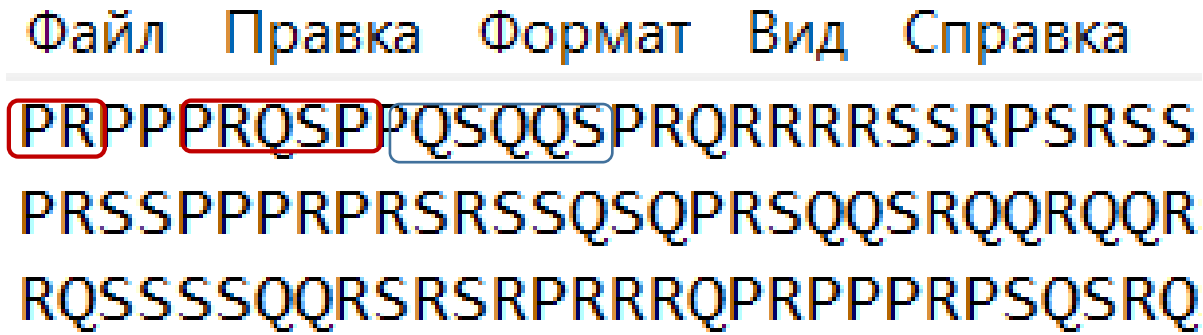


ЕГЭ-2022

Текстовый файл состоит из символов P, Q, R и S.

Определите **максимальное количество идущих подряд символов** в прилагаемом файле, среди **которых нет идущих подряд символов P.**

Для выполнения этого задания следует написать программу.



Описание решения задачи 24 демо версии ЕГЭ-2021

•Для решения задачи применяем *однопроходный алгоритм без вложенного цикла*.
Используемые следующие *переменные*:
curLen – длина текущей цепочки символов, среди которых нет двух подряд букв Р
maxLen – максимальная на данный момент длина цепочки символов, среди которых нет двух подряд букв Р.
•В начальный момент, когда рассматриваем один первый символ (цепочка длины 1 есть всегда!):

(PascalABC.NET) **maxLen := 1**
curLen := 1

(Python) **maxLen = 1**
curLen = 1

- Перебираем в цикле все символы, начиная с **s[1]** (со второго по счёту) до конца строки, сравнивая текущий и предыдущий символы:

PascalABC.NET) for i:=2 to **Length(s)**-1 do

(Python) **for i in range(1,len(s)):**

- Обрабатываем пару символов s[i] и s[i-1], сравниваем текущий и предыдущий символы между собой:

(PascalABC.NET) **if (s[i]<>s[i-1])**

(Python) **if s[i] != s[i-1]**

Если символы не равны, то увеличиваем длину текущей цепочки символов

(PascalABC.NET) **curLen := curLen+1**

(Python) **curlen+=1**

Если символы равны, то обновляем максимум и длину текущей цепочки вновь делаем равной 1

(PascalABC.NET) **if curLen>maxLen then**
maxLen:=curLen;
curLen :=1

(Python) **maxLen=max(curLen, maxLen)**
curLen=1

- Выводим результат:

(PascalABC.NET) **writeln(maxLen)**

(Python) **print(maxlen)**

Программы для решения задачи 24 в демо версии ЕГЭ-2021

АКАДЕМИЯ СОЦИАЛЬНОГО УПРАВЛЕНИЯ

ЦЕНТР НЕПРЕРЫВНОГО ПОВЫШЕНИЯ ПЕДАГОГИЧЕСКОГО МАСТЕРСТВА

Python

```
f=open('24_demo_21.txt')
s=f.read ()
maxLen=1
curLen=1
for i in range(1, len(s)):
        if s[i]!=s[i-1]:
            curLen+=1
        else:
            maxLen=max(curLen, maxLen)
            curLen=1
print(maxLen)
```

PascalABC.NET

```
var    s: string;
       curLen, maxLen,i: integer;
begin
 assign(input,'24_2021.txt');
 reset(input);
 readln(s);
 close(input);
 curLen:=1;
 maxLen:=1;
 for  i:=2 to Length(s)-1 do
     if  (s[i]<>s[i-1])  then
        curLen := curLen+1
     else
        begin
          if   curLen>maxLen then
               maxLen:=curLen;
          curLen :=1;
        end;
 writeln(maxLen);
end.
```

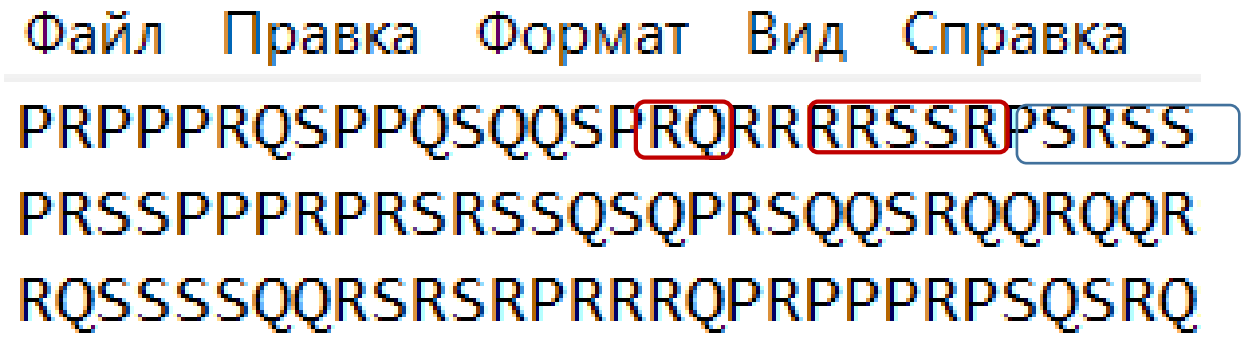
Задача № 24

Демо версия ЕГЭ-2022

Текстовый файл состоит из символов P, Q, R и S.

Определите **максимальное количество идущих подряд символов** в прилагаемом файле, среди **которых нет идущих подряд символов P.**

Для выполнения этого задания следует написать программу.



Описание решения задачи 24
демо версии ЕГЭ-2022

•Для решения задачи применяем *однопроходный алгоритм без вложенного цикла*.
Используемые следующие *переменные*:
curLen – длина текущей цепочки символов, среди которых нет двух подряд букв Р
maxLen – максимальная на данный момент длина цепочки символов, среди которых нет двух подряд букв Р.
•В начальный момент, когда рассматриваем один первый символ (цепочка длины 1 есть всегда!):
maxLen = 1
curLen = 1
• Перебираем в цикле все символы, начиная с **s[1]** (со второго по счёту) до конца строки, сравнивая текущий и предыдущий символы:
for i in range(1,len(s)):
• Обрабатываем пару символов s[i-1] и s[i], сравниваем текущий и предыдущий символы с буквой Р :
if s[i] == 'P' and s[i-1] == 'P
Если оба символа равны Р, то увеличение длины текущей цепочки символов, среди которых нет идущих подряд символов Р, прерывается, и текущая длина вновь становится равна 1:
curlen=1
Если оба символа не равны Р одновременно, то обновляем максимум и увеличиваем длину текущей цепочки символов, среди которых нет идущих подряд символов Р, на 1
maxLen=max(curLen, maxLen)
curLen=1
• Выводим результат: **print(maxlen)**

АКАДЕМИЯ СОЦИАЛЬНОГО УПРАВЛЕНИЯ

ЦЕНТР НЕПРЕРЫВНОГО ПОВЫШЕНИЯ ПЕДАГОГИЧЕСКОГО МАСТЕРСТВА

Программа для решения задачи 24
демо версии ЕГЭ-2022 (Python)

```
f=open('24_2022.txt')              # открываем файл для чтения
s=f.read ()                        # считываем символы в строку s
maxlen=1                           #максимальная длина на данный момент цепочки  символов, среди которых нет двух  P подряд
curlen=1                           #длина текущей цепочки, среди которых нет двух P подряд на данный момент
for i in range(1, len(s)):           #перебираем в цикле все символы, начиная с s[1]( второго по счёту) до конца строки,
   if s[i] == 'P' and s[i-1] == 'P':   #сравниваем текущий и предыдущий символы с буквой P
       maxlen=max(curlen,maxlen) # если оба символа равны P, то  обновляем максимум
       curlen=1                    # длина текущей цепочки символов, среди которых нет
                                     идущих подряд символов P, вновь становится равна 1
   else:
      curlen+=1                 # если же два символа подряд не равны P, то длина текущей цепочки символов увеличивается на 1
print(maxlen)                      # выводим результат
```

Программы для решения задач. ЕГЭ-2022

АКАДЕМИЯ СОЦИАЛЬНОГО УПРАВЛЕНИЯ

ЦЕНТР НЕПРЕРЫВНОГО ПОВЫШЕНИЯ ПЕДАГОГИЧЕСКОГО МАСТЕРСТВА

Python

```
f=open('24_2022.txt')
s=f.read ()
maxlen=1
curlen=1
for i in range(1, len(s)):
    if s[i] == 'P' and s[i-1] == 'P':
        curlen=1
    else:
        curlen+=1
        maxlen=max(curlen,maxlen)
print(maxlen)
```

PascalABC.NET

```
var
  s: string;
  curLen, maxLen,i: integer;
begin
  assign(input, '24_2022.txt');
  readln(s);
  curLen := 1;
  maxLen := 1;
  for  i := 2 to Length(s) do
   if  (s[i] = 'P') and (s[i-1] = 'P') then
     curLen := 1
   else
     begin
       curLen := curlen+1;
       if curLen > maxLen then
         maxLen := curLen;
     end;
  writeln(maxLen);
end.
```

АКАДЕМИЯ СОЦИАЛЬНОГО УПРАВЛЕНИЯ

Задача № 24 (Е. Джобс)

ЦЕНТР НЕПРЕРЫВНОГО ПОВЫШЕНИЯ ПЕДАГОГИЧЕСКОГО МАСТЕРСТВА

Текстовый файл состоит не более чем из 10^6 символов S, T ? O, K.
Определите сколько раз встречается в файле комбинация «КОТ».
Для выполнения этого задания следует написать программу.

Описание решения № 24 (Е. Джобс).

Введем пеменнную count - количество комбинаций КОТ, начальное значение count=0

Циклом проходим все цепочки длины 3. Пусть переменная i хранит номер первого элемента в тройке, то есть будем рассматривать тройки s[i], s[i+1], s[i+2].

Таким образом перебираем в цикле все символы строки кроме двух последних, для них троек нет.

Если тройка подходит под условие: s[i]+ s[i+1]+s[i+0]== 'КОТ‘, то переменную count увеличиваем на 1.

Выводим результат – последнее значение count

Программа для решения № 24 (Е. Джобс).

```
f=open('24_ED.txt')                  # открываем файл для чтения
s=f.read ()                          # считываем символы в строку
count=0                              #количество комбинаций КОТ
for i in range(1, len(s)-2):         #перебираем в цикле все символы строки кроме двух последних
    if s[i]+ s[i+1]+s[i+0]== 'КОТ':  #сравниваем комбинацию трех символов с комбинацией КОТ
            count+=1                 #увеличиваем количество комбинацией на 1,
                                     если условие выполнилось, то есть получилась комбинация КОТ
print(count)                         #выводим результат - значение переменной count
```

Задача № 24

СтатГрад от 02.02.2021

Текстовый файл содержит только заглавные буквы латинского алфавита (ABC…Z). Определите символ, который чаще всего встречается в файле между двумя одинаковыми символами. Например, в тексте CBCABABACCC есть комбинации CBC, ABA (два раза), BAB и CCC. Чаще всего – 3 раза – между двумя одинаковыми символами стоит B, в ответе для этого случая надо написать B.

Описание идеи решения № 24.
СтатГрад от 02.02.2021

Циклом проходим все цепочки длины 3. Пусть переменная i-1 хранит номер первого элемента в тройке, то есть будем рассматривать тройки s[i-1], s[i], s[i+1].
Таким образом перебираем в цикле все символы строки кроме двух последних, для них троек нет.

s[i-1]	s[i]	s[i+1]
A	B	A
C	B	C
A	B	A
B	A	D
C	C	C

……

Вот такие тройки нас будут интересовать.
Пробежав по всей строке, мы узнаем, сколько встретилась буква А между двумя одинаковыми, сколько буква В, буква С и т.д .
Чтобы сохранить количество для каждой буквы создадим массив с именем a.
В него на i-ое месте будем накапливать количество вхождений для каждой латинской буквы от A до Z.
Поскольку латинских букв 26 , то массив будет состоять из 26 элементов.
Сначала этот массив заполним нулями.

Описание идеи решения № 24.
СтатГрад от 02.02.2021

Чтобы узнать, какой элемент массива a увеличивать, если встретилась буква, которая нам подходит, будем использовать кодовую таблицу ASCII
В этой таблице у каждой латинской буквы есть свой порядковый номер.

Описание идеи решения № 24.
СтатГрад от 02.02.2021

Количество букв А заносится на нулевое место в массив, букв В – на первое, С – на второе и т.д.
Как это организовать?

Будем использовать функцию ord () , которая для каждой заглавной латинской буквы вернет число из кодовой таблицы символов, представляющее ее позицию.
Например, **ord ('A')=65, ord ('B')=66, ord ('C') =67, …, ord ('Z')=90.**

a[0] → 0= **ord ('A') - ord ('A')**
a[1] 1= **ord ('B') - ord ('A')**
a[2] 2= **ord ('C') - ord ('A')**
…
a[25] 25= **ord ('Z') - ord ('A')**

После того, как массив а будет заполнен, нам нужно выбрать из него максимальный элемент и узнать его номер.
Затем к найденному номеру снова прибавляем порядковый номер буквы А и с помощью **функцию chr (),** которая преобразует число в символ, выводим найденный символ: **print(chr(i+ord('A')))**
В нашем примере такой буквой стала буква В, она встретилась между двумя другими три раза, значит a[1]=3.
Получаем max=3, nom=1. print(chr(1+65)), будет напечатана буква В, поскольку в таблице ее номер 66

АКАДЕМИЯ СОЦИАЛЬНОГО УПРАВЛЕНИЯ

Программа для решения № 24.

СтатГрад от 02.02.2021.

ЦЕНТР НЕПРЕРЫВНОГО ПОВЫШЕНИЯ ПЕДАГОГИЧЕСКОГО МАСТЕРСТВА

```
f = open('24_0020221.txt') # открываем файл для чтения
s = f.readline()           # считываем символы в строку s
f.close()                  # закрываем файл
a = [0] * 26               #создаем массив из 26 элементов, заполняем его нулями. В этот массив будем заносить количество вхождений
                             каждой буквы, которая нам подходит, то есть стоит между двумя одинаковыми в строке.
for i  in range(2, len(s)-1):  #циклом проходим по строке, начиная со второго и заканчивая вторым от конца строки
   if s[i-1] == s[i+1]:        #сравниваем предыдущий и последующий символы
      a[ord(s[i]) - ord('A')] += 1 # из порядкового номера символа, который нам подходит, вычитаем порядковый номер символа А.
                                     Так мы узнаем порядковый  номер элемента в массиве а.
max = 0                      # начальное значение максимального элемента
nom = 0                      # начальное значение номера максимального
for i in range (len(a)):     #циклом проходим по всей длине массива а
   if a[i] > max:            #если текущий элемент массива а больше максимального,
      max = a[i]             #то максимальный заменяем на текущий
      nom = I                #номер меняет значение на номер текущего элемента, который стал максимальным
print(chr(i+ord('A')))       # узнаем номер  наиболее часто встречаемого символа i, преобразуем его в символ, используя chr()
```