

ЕГЭ по информатике 2022: задание 24

Настоящая статья содержит разбор задания, представленного в проекте демонстрационного варианта Единого государственного экзамена по информатике и ИКТ 2022 года [1], и является второй в цикле статей, описывающих методы решения заданий ЕГЭ 2022, предусматривающих самостоятельное написание программы на одной из доступных на экзамене систем программирования.

В первой статье [2] мы подробно рассмотрели решения заданий 16 и 17, представленных в проекте демонстрационного варианта Единого государственного экзамена по информатике и ИКТ 2022 года [1]. Однако самостоятельное написание программы для решения задачи не ограничивается только заданиями 16 и 17. В настоящей, второй по счету статье мы продолжим рассмотрение такого типа заданий и подробно разберём следующее подобное задание под номером 24.

Ранее, до 2020 года, то есть до перевода ЕГЭ по информатике в компьютерную форму, в демонстрационном варианте Единого государственного экзамена по информатике и ИКТ [3] содержался подробный разбор заданий 24 и 25, причём на нескольких языках программирования. Перевод ЕГЭ по информатике в компьютерную форму повлёк не только изменение самих заданий (в сторону усложнения), но и внёс изменения в сам демонстрационный вариант: сейчас в таком варианте содержатся только краткие ответы для самопроверки. Настоящей статьёй мы восполним этот недостаток.

Задание 24

Условие

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 R.

Для выполнения этого задания следует написать программу.

Решение

Кому-то может показаться, что задача требует обязательной обработки строк и использования специальных функций работы со строками и/или регулярных выражений, но, как и в рассмотренном в предыдущей статье [2] задании 17, это совсем не обязательно.

На самом деле, оба эти задания (17 и 24) очень похожи, только вместо числа, как в задании 17, здесь используется символ. В обеих задачах используются два подряд идущих числа/символа, поэтому и структуру алгоритма их последова-

тельного считывания, обработки и поиска подходящих пар можно взять ту же. Отличия будут только в производимых действиях над самой уже найденной парой.

Итак, как и в предыдущей задаче, мы можем считывать файл посимвольно, храня только текущий (b) и предыдущий (a) символы, сразу же их обрабатывая. Нам также потребуется переменная для хранения максимального значения длины подходящей подпоследовательности (m) и переменная для подсчета текущей длины (k) подпоследовательности БЕЗ идущих подряд символов «Р». Из условия задания следует, что у нас возможны две ситуации:

1. Текущая пара не содержит двух символов «Р». В этом случае мы просто увеличиваем длину подпоследовательности (k) на 1.

2. Текущая пара состоит из двух букв «Р». Это означает, во-первых, что подходящая подпоследовательность завершилась на предыдущем символе (пока ещё не появился второй символ «Р»), и её длину (k «старую») обязательно нужно сравнить с искомым максимумом (m).

Во-вторых, это означает также и то, что на текущем символе началась новая подпоследовательность (в которой нет двух букв «Р», т.к. предыдущий символ, тоже содержащий символ «Р», мы в неё уже не включаем), и её длина (k «новая») теперь равна 1.

После завершения ввода всех символов с одновременной их обработкой, мы получим результат проверки всех подпоследовательностей, завершившихся двойным «Р» в процессе ввода. Но не стоит забывать, что последнюю последовательность мы могли в процессе ввода не обработать (т.к. проверка производилась только при возникновении двойного «Р»). Заключительная последовательность может и не завершаться не двойным «Р», поэтому, после завершения цикла, мы эту последовательность обязательно обрабатываем, дополнительно сравнивая длину последней подпоследовательности (k) с искомым максимумом (m). В итоге, в переменной для хранения максимума (m) мы получаем ответ: 188.

Ниже приведена реализующая описанный алгоритм программа, написанная на четырех языках программирования (рис. 1-4).

C++ (№24)

```
#include <iostream>
#include <fstream>
using namespace std;

int main()
{
    ifstream f("24.txt");
    char a, b;
    long k, m;
    k = 1;
    m = 0;
    f >> a;
    while (f >> b)
    {
        if ((a == 'P') && (b == 'P'))
        {
            if (k > m)
                m = k;
            k = 1;
        }
        else
            k++;
        a = b;
    }
    if (k > m)
        m = k;
    cout << m;
    f.close();
    return 0;
}
```

Рис. 1

Алгоритмический язык (№24)

```
алг
нач
  файл f
  сим a, b
  цел k, m
  f := открыть на чтение("24.txt")
  k := 1
  m := 0
  ввод f, a
  нц пока не конец файла(f)
    ввод f, b
    если a = "P" и b = "P" то
      если k > m то
        m := k
      все
        k := 1
    иначе
      k := k + 1
    все
      a := b
  кц
  если k > m то
    m := k
  все
  вывод m
  закрыть(f)
кон
```

Рис. 2

Паскаль (№24)

```
var
  f: text;
  a, b: char;
  k, m: longint;
begin
  assign(f, '24.txt');
  reset(f);
  k := 1;
  m := 0;
  read(f, a);
  while not eof(f) do
    begin
      read(f, b);
      if (a = 'P') and (b = 'P') then
        begin
          if k > m then
            m := k;
          k := 1
        end
      else
        k := k + 1;
      a := b
    end;
  if k > m then
    m := k;
  write(m);
  close(f)
end.
```

Рис. 3

Python (№24)

```
f = open("24.txt")
k = 1
m = 0
a = f.read(1)
for b in f.read():
    if a == 'P' and b == 'P':
        if k > m:
            m = k
        k = 1
    else:
        k += 1
    a = b
if k > m:
    m = k
print(m)
f.close()
```

Рис. 4

Примечание: на языке Питон может оказаться привычнее сразу считать всю последовательность символов из файла (если нам хватает оперативной памяти) одной строкой в строковую переменную (т.к. в файле все символы P, Q, R и S идут подряд без разделителей в одной строке), и обработать символы уже сохраненной строки по такому же алгоритму.

Программа на языке Питон, реализующая этот алгоритм, представлена ниже (рис. 5).

Python (№24 «строки»)
<pre>f = open("24.txt") k = 1 m = 0 s = f.readline() for i in range(len(s)-1): if s[i] == 'P' and s[i+1] == 'P': if k > m: m = k k = 1 else: k += 1 if k > m: m = k print(m) f.close()</pre>

Рис. 5

Литература

1. https://doc.fipi.ru/ege/demoversii-specifikacii-kodifikatory/2022/inf_11_2022.zip
2. Муржин Д.В. "ЕГЭ по информатике 2022: задачи 16 и 17" // Потенциал (Математика. Физика. Информатика). – 2021. – № 8.
3. https://doc.fipi.ru/ege/demoversii-specifikacii-kodifikatory/2020/inf_ege_2020.zip
4. https://doc.fipi.ru/ege/demoversii-specifikacii-kodifikatory/2022/izm_ege_2022.pdf
5. <https://doc.fipi.ru/ege/demoversii-specifikacii-kodifikatory/2021/inf-ege-2021.zip>

Статью к публикации подготовил

Муржин Дмитрий Викторович

*Преподаватель подготовительных курсов
факультета ВМК МГУ им. М. В. Ломоносова*