

Условие

Исполнитель Редактор получает на вход строку символов и преобразовывает её. Редактор может выполнять две команды, в обеих командах v и w обозначают цепочки символов.

А) заменить (v, w).

Эта команда заменяет в строке первое слева вхождение цепочки v на цепочку w . Например, выполнение команды заменить (111, 27) преобразует строку 05111150 в строку 0527150.

Если в строке нет вхождений цепочки v , то выполнение команды заменить (v, w) не меняет эту строку.

Б) нашлось (v).

Эта команда проверяет, встречается ли цепочка v в строке исполнителя Редактор. Если она встречается, то команда возвращает логическое значение «истина», в противном случае возвращает значение «ложь». Строка исполнителя при этом не изменяется.

Цикл

ПОКА условие

последовательность команд

КОНЕЦ ПОКА

выполняется, пока условие истинно.

В конструкции

ЕСЛИ условие

ТО команда1

ИНАЧЕ команда2

КОНЕЦ ЕСЛИ

выполняется команда1 (если условие истинно) или команда2 (если условие ложно).

Дана программа для Редактора:

НАЧАЛО

ПОКА нашлось (>1) ИЛИ нашлось (>2) ИЛИ нашлось (>0)

ЕСЛИ нашлось (>1)

ТО заменить ($>1, 22>$)

КОНЕЦ ЕСЛИ

```
ЕСЛИ нашлось (>2)
ТО заменить (>2, 2>)
КОНЕЦ ЕСЛИ
ЕСЛИ нашлось (>0)
ТО заменить (>0, 1>)
КОНЕЦ ЕСЛИ
КОНЕЦ ПОКА
КОНЕЦ
```

На вход приведённой выше программе поступает строка, начинающаяся с символа «>», а затем содержащая 39 цифр «0», n цифр «1» и 39 цифр «2», расположенных в произвольном порядке.

Определите наименьшее значение n , при котором сумма числовых значений цифр строки, получившейся в результате выполнения программы, является простым числом.



Обратите внимание на фразу «расположенных в произвольном порядке»

В преобладающем большинстве случаев это означает, что данный алгоритм работает для строки с произвольным расположением 0 1 и 2

Решение

Нам задана функция трансформации строки, реализуем ее на python, обзовем эту функцию `transform(s)`.

Далее нам понадобится функция, которая будет складывать цифры из нашей строки `sum_digits(s)`. Логика работы этой функции следующая: просматриваем посимвольно нашу строку, если очередной символ '0'...'9' - вычитаем из его кода код символа '0'. Это даст нам числовое значение символа. Такой трюк возможен, так как в ASCII таблице символы '0'..'9' идут друг за другом.

И наконец, функция проверки числа на простое `is_prime(n)`. Она считает количество делителей числа n , если их 2 - то число простое.

```
def is_prime(n):
    dividers = 0
    for i in range(1, n+1):
```

```
        if n%i==0:
            dividers+=1

    return dividers == 2

def sum_digits(s):

    ss = 0

    for ch in s:

        if ch>='0' and ch<='9':
            ss = ss + (ord(ch) - ord('0'))

    return ss

def transform(a):

    while ('>1' in a) or ('>2' in a ) or ('>0' in a):

        if '>1' in a:
            a = a.replace('>1', '22>', 1)

        if '>2' in a:
            a = a.replace('>2', '2>', 1)

        if '>0' in a:
            a =a.replace('>0', '1>', 1)

    return a

flag = False
n = 1
while not flag:
    s = '>'+39*'0'+n*'1'+39*'2'

    s = transform(s)

    val = sum_digits(s)

    if is_prime(val):
        print(n)
        flag = True
    n=n+1
```

```
$python ex.py
```

```
5
```

From:

<http://lidarbackup.dvo.ru/dokuwiki/> - **Записки репетитора**

Permanent link:

http://lidarbackup.dvo.ru/dokuwiki/doku.php/ege_inf:task12:ex1



Last update: **2023/03/02 13:24**