

Университетские субботы-2021. Информатика.

Юлия Борисовна Буркатовская, ОИТ

Алгоритмы и исполнители



**Андрей Андреевич Марков
(1903–1979)**



Алонзо Чёрч (1903–1995)



Алан Тьюринг (1912–1954)

- **Алонзо Черч (1936г.). Лямбда-исчисление.**
- **Алан Тьюринг (1936г.). Машина Тьюринга.** Абстрактная вычислительная машина с бесконечной лентой памяти, конечными входным алфавитом, множеством состояний и заданной функцией перехода.
- **А.А.Марков (1940-1950-е гг.). Нормальные алгоритмы Маркова.** Система подстановок для строк символов.

Алгоритм — конечная совокупность точно заданных правил решения некоторого класса задач или набор инструкций, описывающих порядок действий исполнителя для решения определённой задачи.

Задания с сайта <https://kpolyakov.spb.ru/school/ege.htm>

Задание 5. Анализ простых алгоритмов

Преобразование десятичных чисел

Пример 1. Автомат получает на вход четырёхзначное число. По этому числу строится новое число по следующим правилам.

1. Складываются первая и вторая, а также третья и четвёртая цифры исходного числа.

2. Полученные два числа записываются друг за другом в порядке убывания (без разделителей).

Пример. Исходное число: 3165. Суммы: $3 + 1 = 4$; $6 + 5 = 11$. Результат: 114.

Укажите минимальное число, в результате обработки которого, автомат выдаст число 1412.

Решение.

Разберемся, что же делает автомат. Четырёхзначное число выглядит в общем случае так:

$$abcd$$

При этом число не начинается с нуля, поэтому верны следующие условия

$$1 \leq a \leq 9$$

$$0 \leq b \leq 9$$

$$0 \leq c \leq 9$$

$$0 \leq d \leq 9$$

В ходе работы автомата находятся суммы $a + b$ и $c + d$, при этом верно

$$1 \leq a + b \leq 18$$

$$0 \leq c + d \leq 18$$

Это означает, что, если в результате работы автомата получено число 1412, то одна из этих сумм равна 14, а другая – 12. Так как мы ищем **наименьшее** число, то первая и вторая цифры должны быть **как можно меньше**, а значит, и первая сумма. Получаем уравнения

$$a + b = 12$$

$$c + d = 14$$

Чтобы a было минимально, надо максимизировать b , имеем

$$a + b = 12: a = 3, b = 9.$$

Во второй сумме надо минимизировать c и максимизировать d , то есть

$$c + d = 14: c = 5, d = 9.$$

Ответ: 3959.

Пример 2. Автомат получает на вход четырёхзначное число. По этому числу строится новое число по следующим правилам.

1. Складываются первая и третья, а также вторая и четвёртая цифры исходного числа.

2. Полученные два числа записываются друг за другом в порядке возрастания (без разделителей).

Пример. Исходное число: 3165. Суммы: $3 + 6 = 9$; $1 + 5 = 6$. Результат: 69.

Укажите максимальное число, в результате обработки которого, автомат выдаст число 1315.

Решение.

Четырёхзначное число выглядит в общем случае так (условия на цифры те же, что и в предыдущем примере):

$$abcd$$

В ходе работы автомата находятся суммы $a + c$ и $b + d$, при этом верно

$$1 \leq a + c \leq 18$$

$$0 \leq b + d \leq 18$$

Это означает, что, если в результате работы автомата получено число 1315, то одна из этих сумм равна 13, а другая – 15. Так как мы ищем наибольшее число, то большие цифры должны стоять в начале. Первая и вторая цифры входят в разные суммы, обе суммы больше 9, значит, мы можем получить

$$a = 9, b = 9.$$

Далее, чем больше третья цифра, тем больше число. Поэтому, поскольку $a = b$, нам требуется $a + c > b + d$:

$$a + c = 9 + c = 15, c = 6;$$

$$b + d = 9 + d = 13, d = 4.$$

Ответ: 9964.

Пример 3. Автомат получает на вход четырехзначное натуральное число и строит новое число по следующему алгоритму:

1. вычисляются суммы первой и второй, второй и третьей и третьей и четвертой цифр;
2. из полученных сумм отбрасывается наименьшая;
3. остальные суммы записываются в порядке невозрастания.

Пример. Исходное число: 1284. Суммы: $1 + 2 = 3$; $2 + 8 = 10$; $8 + 4 = 12$. Отбрасывается наименьшая сумма 3. Результат: 1210.

Укажите наименьшее число, при вводе которого автомат выдаёт значение 126.

Решение.

Четырехзначное число выглядит в общем случае так (условия на цифры те же, что и раньше):

$$abcd$$

В ходе работы автомата находятся суммы $a + b$, $b + c$ и $c + d$, при этом верно

$$1 \leq a + b \leq 18$$

$$0 \leq b + c \leq 18$$

$$0 \leq c + d \leq 18$$

Важно: эта задача отличается от предыдущих тем, что суммы содержат одинаковые слагаемые, то есть, **значения сумм зависят друг от друга**. Из трех сумм отбрасывается наименьшая, то есть в результате работы автомата (126) участвуют две суммы, это могут быть только 12 и 6, учитывая условия. Помним, что третья сумма, которую отбросили, **не больше 6** (отбрасывается наименьшая сумма).

Мы ищем наименьшее число, поэтому самые большие цифры должны стоять в конце, то есть

$$c + d = 12$$

Ясно, что, так как $d \leq 9$, то $c \geq 3$. Нам нужна маленькая первая цифра, попробуем $a = 1$. У нас осталось две суммы, причем

$$a + b = 1 + b$$

$$b + c \geq b + 3$$

Очевидно, первая сумма меньше, значит, она и была отброшена. Постараемся теперь минимизировать вторую цифру, она может быть и нулем. Если $b = 0$, получаем

$$a + b = 1 + 0 = 1$$

$$b + c = 0 + c = c = 6$$

$$c + d = 6 + d = 12, d = 6$$

Ответ. 1066.

Пример 4. Автомат обрабатывает трёхзначное натуральное число N по следующему алгоритму.

1. Из цифр, образующих десятичную запись N , строятся наибольшее и наименьшее возможные двузначные числа (числа не могут начинаться с нуля).
2. На экран выводится разность полученных двузначных чисел.

Пример. Дано число $N = 351$. Алгоритм работает следующим образом.

1. Наибольшее двузначное число из заданных цифр – 53, наименьшее – 13.
2. На экран выводится разность $53 - 13 = 40$.

Чему равно количество чисел N на отрезке $[300; 400]$, в результате обработки которых на экране автомата появится число 20?

Решение. Ясно, что двузначные числа, получаемые в ходе алгоритма, не зависят от порядка цифр в исходном числе. Рассмотрим, как вообще они могут получаться. Обозначим цифры исходного числа a, b, c , причем $a \leq b \leq c$. возможны следующие случаи. Здесь числа записаны сначала в виде последовательности двух цифр, а потом вычислены через цифры.

Условие	Большее число	Меньшее число	Разность
$0 < a \leq b \leq c \leq 9$	$cb = 10c + b$	$ab = 10a + b$	$10(c - a)$
$0 = a < b \leq c \leq 9$	$cb = 10c + b$	$b0 = 10b$	$10(c - b) + b$
$0 = a = b \leq c \leq 9$	$c0 = 10c$	$c0 = 10c$	0

Итак, разность 20 можно получить только в первом случае, при этом $c - a = 2$. Перечислим все возможные значения цифр. Числа ищутся на отрезке $[300; 400]$, но его границы не подходят, все цифры должны быть больше нуля. Значит, как минимум одна цифра равна 3, это еще ограничивает возможные варианты. Из цифр составляем числа, начинающиеся с 3.

a	b	c	Числа
1	1	3	311
1	2	3	312, 321
1	3	3	313, 331
2	3	4	324, 342
3	3	5	335, 353
3	4	5	345, 354
3	5	5	355

Ответ. 12.

В целом при анализе данных алгоритмов обращайтесь внимание на следующее:

- 1) Работаем с цифрами, их значения от 0 до 9 (кроме первой), сумма двух цифр не превышает 18.
- 2) Первая цифра не может быть нулем, но остальные могут.
- 3) В минимальном числе минимизируем первые цифры, в максимальном максимизируем первые цифры. Другими словами, если мы рассматриваем две цифры и ищем минимальное число, та цифра, которая стоит левее, должна быть как можно меньше.
- 4) Какие цифры используются в суммах? Используется ли одна цифра в двух суммах? Если да, то ее значение одно и то же в двух суммах.
- 5) Внимательно читаем вопрос в задании. Иногда спрашивается, каково исходное число, иногда – каков результат.
- 6) Если это кажется очевидным, не обижайтесь, я перечислила ошибки, которые обычно делают люди при решении этих задач. Я их не придумала.

Преобразование двоичных чисел

Пример 1. На вход алгоритма подаётся натуральное число N . Алгоритм строит по нему новое число R следующим образом.

1) Строится двоичная запись числа N .

2) К этой записи дописываются справа ещё два разряда по следующему правилу:

а) в конец числа (справа) дописывается 1, если число единиц в двоичной записи числа чётно, и 0, если число единиц в двоичной записи числа нечётно.

б) к этой записи справа дописывается 1, если остаток от деления количества единиц на 2 равен 0, и 0, если остаток от деления количества единиц на 2 равен 1.

Полученная таким образом запись (в ней на два разряда больше, чем в записи исходного числа N) является двоичной записью искомого числа R . Укажите минимальное число R , которое превышает 54 и может являться результатом работы алгоритма. В ответе это число запишите в десятичной системе.

Решение.

Разберемся, что делает алгоритм. Пункт 1 понятен, перейдем к 2а.

В конец числа (справа) дописывается 1, если число единиц в двоичной записи числа чётно, и 0, если число единиц в двоичной записи числа нечётно.

Несколько примеров.

$$100011 \rightarrow 1000110$$

$$111100 \rightarrow 1111001$$

$$111111 \rightarrow 1111111$$

$$111 \rightarrow 1110$$

Каким в результате будет полученное число? Если число единиц было **чётным**, то мы дописываем 1, и число единиц становится **нечётным**. Если же число единиц было **нечётным**, в конец дописывается 0, и число единиц **не меняется**.

Таким образом, **в результате шага 2а число единиц нечётно**.

Рассмотрим шаг 2б. Он сформулирован иначе, но если вчитаться и понять смысл формулировки, то ясно, что они абсолютно одинаковы.

а) в конец числа (справа) дописывается 1, если **число единиц в двоичной записи числа чётно**, и 0, если **число единиц в двоичной записи числа нечётно**.

б) к этой записи справа дописывается 1, если **остаток от деления количества единиц на 2 равен 0**, и 0, если **остаток от деления количества единиц на 2 равен 1**.

То, что выделено одинаковым цветом, имеет одинаковый смысл. Поэтому, вспоминая, что после шага 2а число единиц нечётно, приходим к выводу, что на шаге 2б в конец числа просто дописывается 0.

Итак, мы получаем **числа с нечётным числом единиц и с нулем в конце**. Вспоминаем, что если двоичная запись числа оканчивается на 0, то число четное. То есть результатом работы алгоритма является **чётное число, в двоичной записи которого нечётное число единиц**.

Дальше можно просто подобрать подходящее число, большее 54. Рассматриваем только четные числа, переводим их в двоичную систему и проверяем.

$$56 = 32 + 16 + 8 = 111000.$$

Нам сразу повезло, 56 подходит.

Ответ. 56.

Пример 2. На вход алгоритма подаётся натуральное число N . Алгоритм строит по нему новое число R следующим образом.

1) Строится двоичная запись числа N .

2) Затем справа дописываются символы 01, если число N чётное, и 10, если нечётное.

Полученная таким образом запись (в ней на два разряда больше, чем в записи исходного числа N) является двоичной записью искомого числа R . Укажите минимальное число N , после обработки которого автомат получает число, большее 138. Ответ запишите в десятичной системе.

Решение.

И снова разберемся, что делает автомат. Если число **чётное**, то в двоичной записи оно **оканчивается на 0**, потом к нему дописываются цифры 01, получаем, что число оканчивается на 001. Если число **нечётное**, то в двоичной записи оно **оканчивается на 1**, потом к нему дописываются цифры 10, получаем, что число оканчивается на 110. Получается, что нам результатом работы алгоритма будет число, которое **оканчивается на 001 или 110**. Поищем такие числа среди чисел, больших 138.

$$139 = 128 + 8 + 2 + 1 = 10001011$$

Число заканчивается на 011, если поменять эту последовательность на 001, оно станет меньше и нам не подойдет, чтобы придется добавить единичный разряд вместо нулевого, получим

$$10010001.$$

Тут мы постарались найти наименьшее число, большее данного, и заканчивающееся на 001. Но есть и другая возможность, и можно было сразу начать с нее – поменять в конце числа 011 на 110, таким образом число увеличится, но меньше, чем в предыдущем случае.

$$10001110 = 128 + 8 + 4 + 2 = 142$$

К тому же результату мы могли прийти и перебором.

$$140 = 128 + 8 + 4 = 10001100$$

$$141 = 128 + 8 + 4 + 1 = 10001101$$

$$142 = 128 + 8 + 4 + 2 = 10001110$$

В ответе нужно указать число, из которого получается 142. Чтобы найти его, отбросим две последние цифры, добавленные в алгоритме.

$$10001110 \rightarrow 100011 = 32 + 2 + 1 = 35.$$

Либо, понимая, что отбрасывание двух цифр двоичного числа равнозначно его целочисленному делению на 4, просто получаем 35 как частное 142 и 4.

Альтернативное решение.

Если вы понимаете, как устроена позиционная система счисления, то вы понимаете следующее:

- Дописывание в конец двоичного числа 01 означает, что число умножили на 4 и затем прибавили к нему 1;
- Дописывание в конец двоичного числа 10 означает, что число умножили на 4 и затем прибавили к нему 2.

Примеры.

До преобразования		После преобразования	
Двоичное	Десятичное	Двоичное	Десятичное
1100	12	110001	$49 = 12 \times 4 + 1$
1010	10	101001	$41 = 10 \times 4 + 1$
1101	11	110110	$54 = 13 \times 4 + 2$
1001	9	100110	$38 = 9 \times 4 + 2$

Итак, можно не переходить к двоичной записи и не выполнять шаги алгоритма, а вывести формулу для результирующего числа.

$$N = 2k: R = N \times 4 + 1 = 2k \times 4 + 1 = 8k + 1$$

$$N = 2k + 1: R = N \times 4 + 2 = (2k + 1) \times 4 + 2 = 8k + 6$$

То есть результат имеет вид $R = 8k + 1$ или $R = 8k + 6$, ищем число в такой форме, большее 138:

$$139 = 8 \times 17 + 3$$

...

$$142 = 8 \times 17 + 6$$

Итак,

$$142 = R = 8k + 6 = N \times 4 + 2, N = 35$$

Ответ. 35

Пример 3. Автомат обрабатывает целое число N ($0 \leq N \leq 255$) по следующему алгоритму:

- 1) Строится восьмьбитная двоичная запись числа N .
- 2) Все цифры двоичной записи заменяются на противоположные (0 на 1, 1 на 0).
- 3) Полученное число переводится в десятичную запись.
- 4) Из нового числа вычитается исходное, полученная разность выводится на экран.

Какое число нужно ввести в автомат, чтобы в результате получилось 45?

Решение. Разберемся, что происходит с числом на втором шаге. Вспоминаем математическую логику – на втором шаге число инвертируется покомпонентно. Приведем несколько примеров.

$$00100110 \rightarrow 11011001$$

$$10011111 \rightarrow 01100000$$

$$00000001 \rightarrow 11111110$$

Что будет, если сложить число и его инверсию? Сложение «в столбик» работает одинаково во всех позиционных системах счисления.

$$00100110$$

$$\underline{11011001}$$

$$11111111$$

Нетрудно понять, что во всех разрядах складываются 0 и 1, в результате получается 1, то есть результат сложения будет 11111111 (в двоичной системе). Это самое большое число, которое можно записать 8 битами, если к нему прибавить 1, то получим число 100000000(2), в десятичной системе это 256. Таким образом, $11111111(2)=255$. Итак, если N – исходное число, \bar{N} – инвертированное число, полученное на шаге 2, то

$$N + \bar{N} = 255, \bar{N} = 255 - N.$$

Третий шаг пропускаем, поскольку мы уже представили \bar{N} в десятичном виде. На четвертом шаге происходит вычитание

$$\bar{N} - N = (255 - N) - N = 255 - 2N.$$

Это результат работы алгоритма. Зная его, легко получить N .

$$R = 255 - 2N, N = \frac{255 - R}{2}.$$

В нашей задаче:

$$R = 45, N = \frac{255 - 45}{2} = 105.$$

Ответ. 105.

В целом при анализе данных алгоритмов обращайтесь внимание на следующее:

- 1) Какие числа мы можем получить в алгоритме, каковы их свойства?
- 2) Если в конец двоичного числа дописали 0, это значит, число умножили на 2. Если дописали 1, это значит, число умножили на 2 и затем прибавили 1.
- 3) Если от двоичного числа отбросили последнюю цифру, это значит, его поделили на 2.
- 4) Вообще думаем, что происходит с десятичным числом при действиях с его двоичной записью. В последнем случае это позволило нам не переходить к двоичной записи, и решить все в десятичных числах.

Арифмометры

Пример 1. Исполнитель КВАДРАТОР имеет только две команды, которым присвоены номера:

1. возведи в квадрат

2. прибавь 1

Выполняя команду номер 1, КВАДРАТОР возводит число на экране в квадрат, а выполняя команду номер 2, прибавляет к этому числу 1. Напишите программу, содержащую не более 4 команд, которая из числа 1 получает число 17. Укажите лишь номера команд.

Решение. Тут удобно идти от результата, который мы получили. Понятно, что возведением в квадрат мы можем получить 2, 4, 9, 16, но никак не 17. Поэтому 17 получается только прибавлением единицы. А вот 16 можно получить двумя способами, но, так как программа должна быть короткой, логично попробовать получить 16 возведением в квадрат. Рассуждения оформим в виде таблицы.

Число	Как получено	Команда
17	$16 + 1$	2
16	4^2	1
4	2^2	1
2	$1 + 1$	2

Теперь выпишем команды снизу вверх.

Ответ. 2112.

Пример 2. У исполнителя Троечник две команды, которым присвоены номера:

1. прибавь 3,

2. умножь на 3.

Первая из этих команд увеличивает число на экране на 3, вторая умножает его на 3. Программа для исполнителя Троечник — это последовательность номеров команд. Например, **121** — это программа **прибавь 3, умножь на 3, прибавь 3**. Эта программа преобразует число 1 в число 15.

Запишите программу, которая преобразует число 6 в число 69 и содержит не более 5 команд. Если таких программ более одной, то запишите любую из них.

Решение. Так как требуется не более пяти команд, понятно, что будет два умножения на 3. При этом 6 кратно 3, поэтому при любой операции мы должны получать число, кратное 3. Поэтому программу строим так – как и раньше, начинаем с конца, используем команду умножь на три, только если частное от деления текущего числа на 3 также кратно 3.

Число	Как получено	Команда	Примечание
69	$66 + 3$	1	$69/3 = 23$ не кратно 3
66	$63 + 3$	1	$66/3 = 22$ не кратно 3
63	21×3	2	
21	$18 + 3$	1	$21/3 = 7$ не кратно 3
18	6×3	2	

Ответ. 21221

Пример 3. У исполнителя Аккорд две команды, которым присвоены номера:

1. отними 1

2. умножь на 5

Выполняя первую из них, Аккорд отнимает от числа на экране 1, а выполняя вторую, умножает это число на 5. Запишите порядок команд в программе, которая содержит не более 5 команд и переводит число 5 в число 98.

Решение. Умножением на 5 мы, естественно, получим число, кратное 5. Поэтому действуем так – если число не кратно 5, то оно получено вычитанием 1.

Число	Как получено	Команда	Примечание
98	$99 - 1$	1	98 не кратно 5
99	$100 - 1$	1	99 не кратно 5
100	20×5	2	
20	4×5	2	
4	$5 - 1$	1	

Ответ. 12211

В целом при анализе данных алгоритмов обращайтесь внимание на следующее:

- 1) Какие числа мы можем получить в алгоритме?
- 2) Начинаем с конца, это уменьшает число перебираемых вариантов.
- 3) Возможно, какое-то решение мы примем неправильно, тогда возвращаемся на шаг назад и выбираем другую команду..

Задание 12 Исполнитель «Редактор»

Пример 1. Исполнитель Редактор получает на вход строку цифр и преобразовывает её. Редактор может выполнять две команды, в обеих командах v и w обозначают цепочки цифр.

заменить (v, w)

нашлось (v)

Дана программа для исполнителя Редактор:

НАЧАЛО

ПОКА нашлось (4444) ИЛИ нашлось (777)

ЕСЛИ нашлось (4444)

ТО заменить (4444, 77)

ИНАЧЕ заменить (777, 4)

КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ

Какая строка получится в результате применения приведённой выше программы к строке, состоящей из 197 идущих подряд цифр 4? В ответе запишите полученную строку.

Решение.

Воображаем себя редактором и действуем по алгоритму. В чем он состоит? В цикл заходим, если **нашлось (4444) ИЛИ нашлось (777)**. Затем проверяем условие **нашлось (4444)**. Если оно истинно, **заменить (4444, 77)**. То есть, пока в строке есть 4444, они будут заменяться на 77. Если же 4444 не найдется, то в цикл мы зайдем при наличии 777, затем **заменим 777 на 4**.

Для примера рассмотрим строку из меньшего количества четверок, скажем, 29.

Строка	Условие	Действие
4444 4444444444444444444444444444	нашлось (4444)	заменить (4444, 77)
77 4444 4444444444444444444444444444	нашлось (4444)	заменить (4444, 77)
7777 4444 444444444444444444444444	нашлось (4444)	заменить (4444, 77)
777777 4444 4444444444444444	нашлось (4444)	заменить (4444, 77)
	...	
7777777777777777774		

Таким образом, с начала с последовательностью четверок происходит следующее: каждые четыре четверки заменяются на две семерки. В нашей маленькой последовательности 28 четверок заменилось на 14 семерок.

Посмотрим, что произойдет в последовательности из 197 четверок. Поскольку

$$197 = 49 \times 4 + 1,$$

данная замена произойдет 49 раз, в результате получим последовательность из 98 семерок и одной четверки в конце.

Продолжим выполнение алгоритма для маленького примера.

Строка	Условие	Действие
777 7777777777777777774	нашлось (777)	заменить (777, 4)
4 777 7777777777777777774	нашлось (777)	заменить (777, 4)
44 777 7777777777777777774	нашлось (777)	заменить (777, 4)
444 777 7777777777777777774	нашлось (777)	заменить (777, 4)
4444 7777777777777777774	нашлось (4444)	заменить (4444, 77)
7777777777777777774		

Обратите внимание – внутри цикла сначала проверяется условие **нашлось 4444**, только если оно неверно, 777 заменяется на 4. Таким образом, эта команда выполнялась 4 раза, и последовательность из 12 семерок в начале строки преобразовалась в **четыре четверки**. А потом выполнилось условие **нашлось 4444**, и последовательность 4444 заменилась на 77. В итоге **12 семерок в начале строки преобразовалось в две семерки**, и строка приобрела ту же структуру, с какой мы начали – семерки в начала, одна четверка в конце. Нам удалось выделить несколько итераций алгоритма (пять), которые фактически сокращают строку, не меняя ее вида, и это хорошо – следующие пять итераций сделают то же самое.

Важно: ясно, что в результате количество семерок стало меньше на 10, но это не совсем верно. Посмотрим, например, что произойдет с последовательностью из 11 семерок.

Строка	Условие	Действие
77777777777	нашлось (777)	заменить (777, 4)
477777777	нашлось (777)	заменить (777, 4)
4477777	нашлось (777)	заменить (777, 4)
44477		

Здесь уже не выполнено условие для входа в цикл, алгоритм работу заканчивает.

Таким образом, можем считать, что после данных шагов число семерок уменьшается на 10, **если изначально их было не меньше 12.**

Вспоминаем, что в основной задаче мы пришли к последовательности из 98 семерок и четверки в конце. Затем семерок станет 88, 78, ..., 18, 8, и далее опять рассмотрим работу алгоритма.

Строка	Условие	Действие
777777774	нашлось (777)	заменить (777, 4)
4777774	нашлось (777)	заменить (777, 4)
44774		

Условия входа в цикл не выполнены, алгоритм работу закончил.

Очень важно здесь действовать по инструкциям для исполнителя и не думать за него. Делаем ровно то, что написано в инструкции.

Ответ: 44774.

Программы для данной задачи

<https://kpolyakov.spb.ru/school/ege.htm>

Программа на языке Python

```
s = 197*'4'
while "4444" in s or "777" in s:
    if "4444" in s:
        s = s.replace("4444", "77", 1)
    else:
        s = s.replace("777", "4", 1)
print(s)
```

Программа на языке PascalABC.NET

```
begin
    var s: string := '4'*197;
    while (s.contains('4444')) or (s.contains('777')) do
        begin
            if (s.contains('4444')) then
                s := s.replace('4444', '77', 1)
            else
                s := s.replace('777', '4', 1);
            end;
        writeln(s);
    end.
```

Программа на языке C++

```
#include<iostream>
using namespace std;
int main()
{
    string s(197, '4');
    int p4 = s.find("4444");
    int p7 = s.find("777");
    while( p4 != string::npos or p7 != string::npos ) {
        if( p4 != string::npos )
            s.replace( p4, 4, "77" );
    }
```


Пример 3. Исполнитель Редактор получает на вход строку цифр и преобразовывает её. Редактор может выполнять две команды, в обеих командах v и w обозначают цепочки символов.

заменить (v, w)

нашлось (v)

Первая команда заменяет в строке первое слева вхождение цепочки v на цепочку w. Если цепочки v в строке нет, эта команда не изменяет строку. Вторая команда проверяет, встречается ли цепочка v в строке исполнителя Редактор.

Дана программа для Редактора:

ПОКА нашлось(11)

 заменить(112, 4)

 заменить(113, 2)

 заменить(42, 3)

 заменить(43, 1)

КОНЕЦ ПОКА

Какая строка получится в результате применения приведённой программы к строке вида 1...13...32...2, состоящей из 170 единиц, 100 троек и 7 двоек?

Решение. Здесь проверяется условие нашлось(11), затем выполняются сразу четыре действия. Рассмотрим более короткую строку.

Строка	Условие	Действие
11111111113333333333332222222	нашлось (11)	заменить (112, 4)
111111111 113 3333333333332222222		заменить (113, 2)
11111111 2 3333333333332222222		заменить (42, 3)
1111111123333333333332222222		заменить (43, 1)
1111111 112 3333333333332222222	нашлось (11)	заменить (112, 4)
111111 4 3333333333332222222		заменить (113, 2)
11111143333333333332222222		заменить (42, 3)
111111 43 3333333333332222222		заменить (43, 1)
1111111 1 3333333333332222222		

Мы зашли в цикл два раза и получили строку, аналогичную исходной. Сначала 113 заменили на 2 (ушло две единицы и одна тройка), затем 112 на 4 (ушло еще две единицы и двойка, добавленная ранее), затем 43 на 1 (ушла добавленная четверка и еще одна тройка, добавилась одна единица. В итоге последовательность 111133 заменилась на 1. Можно это проверить – изначально было 10 единиц и 12 троек, в конце – 7 единиц и 10 троек. Таким образом, пока есть группа 111133, данным алгоритмом она заменяется на 1, то есть исчезает последовательность 11133.

У нас 170 единиц, 100 троек и 7 двоек, таким образом, уйдет 150 единиц и 100 троек, останется последовательность из 20 единиц и 7 двоек. Рассмотрим далее поведение алгоритма.

Строка	Условие	Действие
11111111111111111111 112 222222	нашлось (11)	заменить (112, 4)
11111111111111111111 4 222222		заменить (113, 2)
11111111111111111111 42 22222		заменить (42, 3)
11111111111111111111 3 22222		заменить (43, 1)
11111111111111111111322222	нашлось (11)	заменить (112, 4)
11111111111111111111 113 22222		заменить (113, 2)
11111111111111111111 2 22222		заменить (42, 3)
11111111111111111111222222		заменить (43, 1)
11111111111111111111222222		

Заменяется 112 на 4, уходят две единицы и двойка, появляется четверка. Затем заменяется 42 на 3, уходит еще одна двойка и добавленная четверка. Затем 113 на 2, уходят две единицы и добавленная тройка, появляется двойка. Итого: 111122 заменяется на 2, или уходит последовательность 11112, при условии, что в последовательности было две двойки. На данный момент осталось 16 единиц и 6 двоек, то есть, такое можно выполнить еще 4 раза. Останется 22.

Ответ. 22.

Общая стратегия. Выполняем действия строго по алгоритму, пока не придем к строке того же вида, что мы уже получали, но короче. Смотрим, что произошло, какая последовательность цифр исчезла или заменилась на другую последовательность. Почувствуйте разницу между «последовательность сократилась на 10 семерок» и «12 семерок заменилось на 10» (первый пример).

Либо пишем программу.

Исполнитель «Чертежник»

Пример 1. Исполнитель Чертежник перемещается на координатной плоскости, оставляя след в виде линии. Чертежник может выполнять команду Сместиться на (a, b) (где a, b – целые числа), перемещающую Чертежника из точки с координатами (x, y) в точку с координатами $(x + a, y + b)$. Чертежнику был дан для исполнения следующий алгоритм:

```
Сместиться на (16, -21)
Повтори N раз
    Сместиться на (a, b)
    Сместиться на (-1, -2)
конец
Сместиться на (-60, -12)
```

После выполнения этого алгоритма Чертежник возвращается в исходную точку. Какое наибольшее число повторов могло быть указано в конструкции «Повтори ... раз»?

Решение. Прежде всего заметим, что несколько подряд команд смещения равносильны одной команде, где смещение равно сумме всех смещений, то есть

```
Сместиться на (a, b)
Сместиться на (-1, -2)
```

Равносильно команде

```
Сместиться на (a-1, b-2)
```

Итого, весь алгоритм можно заменить одной командой

```
Сместиться на (16+N(a-1)-60, -21+N(b-2)-12)
```

Так как чертежник возвращается в исходную точку, смещение равно нулю. Имеем уравнения.

$$N(a-1) - 44 = 0$$

$$N(b-2) - 33 = 0$$

Поскольку все числа целые, N является делителем 44 и 33, это или 1, или 11. Мы ищем максимальное N , значит, берем 11.

Ответ. 11.