Напишите программу, переводящую действительное неотрицательное число из шестеричной системы счисления в десятичную.

***Входные данные***

Дана запись целого шестеричного числа или шестеричной периодической дроби, которая включает в себя:

1. Необязательную целую часть.  
2. Обязательный символ точки, отделяющий целую часть от дробной.

3. Необязательную дробную непериодическую часть.   
4. Необязательную периодическую дробную часть, записываемую в круглых скобках.

*Выходные данные*

Необходимо определить значение этой дроби  и вывести на экран с точностью не менее 12 знаков. Общая длина входной строки не превосходит 30 символов.

***Примеры***

|  |  |
| --- | --- |
| ***Входные данные*** | ***Выходные данные*** |
| 10.2(23) | 6.4047619047619 |
| 510.0 | 186.0 |
| 0.223 | 0.402777777777 |

a = input()

if '.' not in a:

print(int(a, 6))

exit()

cel, drob = a.split('.')

cel = int(cel, 6)

if '(' not in a:

drob = int(drob, 6) / 6 \*\* len(drob)

print(cel + drob)

exit()

pp = drob[:drob.find('(')]

p = drob[drob.find('(') + 1: -1]

if pp == '':

drob = int(p, 6) / (6 \*\* len(p) - 1)

else:

drob = (int(pp + p, 6) - int(pp, 6)) / (6 \*\* len(pp + p) - 6 \*\* len(pp))

print(cel + drob)

Дано рациональное число. Запишите его в виде троичной периодической дроби.

*Входные данные*

На вход программа получает два натуральных числа *n* и *m*, каждое из которых не превосходит 1000.

*Выходные данные*

Программа должна вывести значение *n*/*m*, записанное в виде троичной периодической дроби, при этом длина непериодической дробной части и длина периода должны быть минимально возможными. Если данное число является конечной троичной дробью, периодическую часть выводить не надо.

*Примеры*

|  |  |
| --- | --- |
| ***Входные данные*** | ***Выходные данные*** |
| 62  5 | 110.(1012) |
| 3  10 | 0.(0220) |
| 39  1 | 1110 |

def NOD(n,m):

if m == 0: return n

return NOD(m, n % m)

n, m = map(int, input().split())

R = ''

if n \* m < 0 or n < 0: #проверка на минусы

if n \* m > 0:

n = -n

m = -m

else:

R = '-'

n = abs(n)

m = abs(m)

if m == 0:

print('Деление на ноль')

exit()

def pp(n):

s = ''

while n:

s = str(n % 3) + s

n //= 3

return s

if n % m == 0:

print(pp(n//m))

exit()

if n > m:

nd = NOD(n, m)

n //= nd

m //= nd

R += pp(n//m) + '.'

n = n % m

else:

R += '0.'

L = [n]

while n != 0:

n \*= 3

R += str(n // m)

n = n % m

if n in L:

R = R[: R.find('.') + L.index(n) + 1] + '(' + R[R.find('.') + L.index(n) + 1:] + ')'

break

else:

L.append(n)

print(R)