

### Задача 1.

Данная задача отлично решается с помощью reduce.

Сначала перевернем строку. Потом будем добавлять | к каждому третьему символу.

Удалим лишний символ, если требуется.

```
from functools import reduce
s = input()[::-1]
ans = reduce(lambda a, x: a+x[1]+"|" if x[0] % 3 == 2 else a + x[1], enumerate(s), "")[::-1]
if len(s) % 3 == 0:
    print(ans[1:])
else:
    print(ans)
```

### Задача 2.

Для хранения таких больших чисел рационально использовать строковый тип.

Заметим, что если строка — палиндром, то вторая половина симметрична первой. Таким образом, разделим строку пополам и сделаем правую симметричной левой. Далее смотрим на получившуюся строку и на исходную. Если новое число больше, чем старое, значит оно является ответом. В противном случае, необходимо прибавить к левой части 1 и сделать правую половину опять симметричной левой.

Также, если на вход подана только одна 9, то необходимо вывести 11. Это единственный, не охватываемый данным алгоритмом.

```
S = str(input())
if S == '9':
    print(11 - int(S))
else:
    n = len(S)
    S1 = S[:-(n // 2)] + S[:n // 2][::-1]
    if S1 > S:
        print(int(S1)-int(S))
    else:
        n1 = (n + 1) // 2
        n2 = n - n1
        N = str(int(S[:n1]) + 1)
        S2 = N + N[:n2][::-1]
        print(int(S2)-int(S))
```