```python
# Code by Rumaisa Abdulhai

N = 0
P = 0

############
# METHODS #
###########

def primes(N):
    num = 2
    l = []

    x = N
    while (num < x):
        if (x % num == 0):
            l.append(num);
            x /= num
        else:
            num +=1

    l.append(num)
    return len(set(l))

def new_left(middle, lefts):
    return int(''.join([str(int(num) + middle) for num in lefts]))

def new_right(middle, rights):
    return int(''.join([str( abs(int(num) - middle) ) for num in rights]))

def new_center(N):
    return primes(N)

def make_new_number(left, center, right):
    return int(str(left) + str(center) + str(right))

###############
# MAIN METHOD #
###############

def transformNum(N,P):

    right_digits = ''
    left_digits = ''
    center_digits = ''

    n_left = ''
    n_right = ''
    n_center = ''

    if P == 1:
        middle_digit = int(str(N)[len(str(N))- 1])
        left_digits = str(N)[:len(str(N))-1]
```

```
54
55          n_left = new_left(middle_digit,left_digits)
56
57      elif P == len(str(N)):
58          middle_digit = int(str(N)[0])
59          right_digits = str(N)[1:]
60
61          n_right = new_right(middle_digit,right_digits)
62
63      else:
64          right_digits = str(N)[-P+1:]
65          left_digits = str(N)[:len(str(N))-P]
66          middle_digit = int(str(N)[len(str(N)) - P])
67
68          n_left = new_left(middle_digit,left_digits)
69          n_right = new_right(middle_digit,right_digits)
70
71      n_center = new_center(N)
72
73      return make_new_number(n_left, n_center, n_right)
74
75  ###################
76  # RUNNING INPUTS #
77  ###################
78
79  print(transformNum(N=102438,P=3))
80  print(transformNum(N=4329,P=1))
81  print(transformNum(N=6710,P=2))
82  print(transformNum(N=16807,P=1))
83  print(transformNum(N=60098065452,P=7))
```