Problem

Someone just won the Code Jam lottery, and we owe them N jamcoins! However, when we tried to print out an over ized check, we encountered a problem. The value of N, which is an integer, includes at least one digit that is a 4... an the 4 key on the keyboard of our oversized check printer is broken.

Fortunately, we have a workaround: we will send our winner two checks for positive integer amounts A and B, such hat neither A nor B contains any digit that is a 4, and A + B = N. Please help us find any pair of values A and B that atisfy these conditions.

Input

The first line of the input gives the number of test cases, T. T test cases follow; each consists of one line with an inte er N.

Output

For each test case, output one line containing Case #x: A B, where x is the test case number (starting from 1), and A nd B are positive integers as described above.

It is guaranteed that at least one solution exists. If there are multiple solutions, you may output any one of them. (See "What if a test case has multiple correct solutions?" in the Competing section of the FAQ. This information about m ltiple solutions will not be explicitly stated in the remainder of the 2019 contest.)

Limits

$1 \leq T \leq 100$.
Time limit: 10 seconds per test set.
Memory limit: 1GB.
At least one of the digits of N is a 4.

Test set 1 (Visible)
$1 < N < 105$.

Test set 2 (Visible)
$1 < N < 109$.

Solving the first two test sets for this problem should get you a long way toward advancing. The third test set is wort only 1 extra point, for extra fun and bragging rights!

Test set 3 (Hidden)
$1 < N < 10100$.

Sample

Input

Output

3
4
940
4444

Case #1: 2 2

Case #2: 852 88
Case #3: 667 3777

In Sample Case #1, notice that A and B can be the same. The only other possible answers are 1 3 and 3 1.

Solution:

```
t = int(raw_input())

def solve():
    s = raw_input()
    a = ""
    for i in xrange(len(s)):
        if s[i] == '4':
            a += '1'
        else:
            a += '0'
    return "%s %s" % (int(a), int(s)-int(a))


for __ in xrange(t):
    print "Case #%d: %s" % (__+1, solve())
```