

## Description

The reverse and add function starts with a number, reverses its digits, and adds the reverse to the original. If the sum is not a palindrome (meaning it does not give the same number read from left to right and right to left), we repeat this procedure until it does.

For example, if we start with 195 as the initial number, we get 9,339 as the resulting palindrome after the fourth addition:

```
195 + 591 = 786
786 + 687 = 1,473
1,473 + 3,741 = 5,214
5,214 + 4,125 = 9,339
```

This method leads to palindromes in a few steps for almost all of the integers. But there are interesting exceptions. 196 is the first number for which no palindrome has been found. It has never been proven, however, that no such palindrome exists. You must write a program that takes a given number and gives the resulting palindrome (if one exists) and the number of iterations/additions it took to find it. You may assume that all the numbers used as test data will terminate in an answer with less than 1,000 iterations (additions), and yield a palindrome that is not greater than 4,294,967,295.

## Input

The first line will contain an integer  $N$  ( $0 < N \leq 100$ ), giving the number of test cases, while the next  $N$  lines each contain a single integer  $P$  whose palindrome you are to compute.

## Output

For each of the  $N$  integers, print a line giving the minimum number of iterations to find the palindrome, a single space, and then the resulting palindrome itself.

## ReversibleInteger Class

Your program must implement the following “ReversibleInteger” class for this problem. You need to implement a constructor function, a `reverse()` function that reverses its digits, an `add()` function that adds two ReversibleInteger objects.

```
#define MAXDIGITS 10

class ReversibleInteger {
private:
    char digits[MAXDIGITS];    /* represent the number */
    int lastdigit;             /* index of high-order digit */
public:
    ReversibleInteger(int n=0);
    ReversibleInteger reverse();
    ReversibleInteger add(ReversibleInteger arg);
    int getValue();
};
```

## Sample Input

```
3
195
265
750
```

## Sample Output

```
4 9339
5 45254
3 6666
```

## Submitting Your Code and Report

Your file name must be "assign3.cpp", and it is the only file you should submit. Please do not submit the executable file of your program. Turn in your project using the "oops submit" command as follows:

```
$ oops submit assign3 assign3.cpp
```

Late submissions will lose 20 points per day.

If you do not follow this submission guideline, you will lose 10 points out of 100.

You should also submit a hard copy of your code to TA. Your report must have a cover page with your student ID and name. In the report, your code must be well commented to explain your algorithm. Also, the sample input and the output of your program must be included in the report.