

60-141 Winter 2018

Assignment 1: Reverse Number Magic Sequence

Due: Friday February 2, 2018 11:59pm EST

A reverse number is a number written in arabic numerals, but where the order of digits is reversed. The first digit becomes the last and vice versa. For example, the number 1245 when its digits are reversed it would become 5421. Note that all the leading zeros are omitted. That means if the number ends with a zero, the zero is lost by reversing (e.g. 1200 gives 21). Also note that the reversed number never has any trailing zeros. Finally, every single digit number (i.e. 0-9) is its own reverse number.

In order to generate a magic number, we reverse a given original number and store the absolute value of the difference between the original number and its reversed version. For example, given the number 476, we will generate the reverse number 674 and then compute the absolute value of the difference between 476 and 674 to be 198. We then reverse 198 to display the number 891; we call that the magic number!

We need your help to compute the magic numbers of a given sequence. Your task is to calculate the difference between a given number and its reverse version, and output the reverse of the difference. Of course, the result is not unique because any particular number is a reversed form of several numbers (e.g. 21 could be 12, 120 or 1200 before reversing). Thus we must assume that no zeros were lost by reversing (e.g. assume that the original number was 12).

Input

The input consists of N numbers, where N is an arbitrary positive integer. The first line of the input contains only a positive integer N. Then follows one or more lines with the N numbers; these numbers should all be non-negative and may be single or multiple digits. These are the original numbers you need to generate their N corresponding magic numbers.

Output

For each original number in the sequence, print exactly one integer – its magic number. Omit any leading zeros in the output. On a separate line, output the largest absolute difference encountered in the sequence.

Sample Input

```
6
24 1 4358 754 305 794
```

Sample Output

```
81 0 6714 792 891 792
4176
```

Specific Requirements: [15 pts]

- [3 pts] Write a function called **reverseInteger**, that takes as input an unsigned integer and returns its reversed digits version as an unsigned integer.
- [3 pts] Write a function called **generateMagicNumber**, that takes as input an unsigned integer and return its magic number as described in the problem.
- [3 pts] Display the sequence of magic numbers correctly. (shown in the script file)
- [2 pts] Display the largest absolute difference (shown in the script file)
- [3 pts] Demonstrate the complete program using a main function capable of processing the input of any sequence and producing its corresponding output.
- [1 pt] Compilation on the CS server gcc compiler without errors and warnings.

Failure to properly document your entire code will receive a mark of zero.

You are to submit the following:

- Source code file: **assign1.c**
- Script file demonstrating the compilation and execution : **assign1.txt**
To generate the script file use the following command from the CS server:
`cp assign1.c assign1.backup`
`typescript assign1.txt`
`cc assign1.c`
`a.out`
[test your code here with at least 3 different input test cases in addition to the example given]
`exit`
[These steps will create a file called assign1.txt. Do not edit its contents - just submit it on

Blackboard!]

Hint:

This table explains the work done in this example:

Original number	Reverse	Absolute difference	Reverse (Magic number)
X	X_r	$ X - X_r $	$ X - X_r _r$
24	42	18	81
1	1	0	0
4358	8534	4176	6714
754	457	297	792
305	503	198	891
794	497	297	792

Note that your program should not use arrays and should be able to read a sequence of N size, for any value of N (a 32 bit integer). Of course, memory space optimization should be considered since there is no need to store all the N numbers in memory all at once at any given time.