# Data Structure and Programming, Fall 2018
# Programming Assignment #2

Due 2 p.m., November 5, 2018

## 1 Problem Statement

When Prof. Wu was recording the scores of the midterm exam, he forgot to add a comma between two adjacent scores. Therefore, all the scores were concatenated into a single digit string. At that time, Prof. Wu came up an interesting question: how many possible ways could a given digit string be separated into the original scores by adding commas?

This programming assignment asks you to compute how many ways a string composed of digits can be divided into integers (“$n_1, n_2, ..., n_k$”) ranging **from 0 to 100** ($0 \leq n_i \leq 100, i = 1, 2, ..., k$). For example, the digit string “21005” can be divided into “2,1,0,0,5”, “21,0,0,5”, “2,10,0,5”, or “2,100,5”, and thus there are **4** ways to divide the string. Note that 0-leading numbers (except for 0 itself) are not allowed. For example, “2,10,05” is not allowed due to the leading zero in “05”.

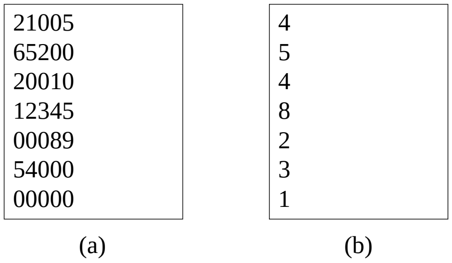
## 2 Input/Output Specification

### 2.1 Input Format

The input consists of several digit strings separated by a newline. Figure 1(a) shows an input example.

### 2.2 Output Format

The output contains the answers of the input in the order of the input. The answers should be separated by a newline. Figure 1(b) shows an output example.



```
21005
65200
20010
12345
00089
54000
00000
```

(a)

```
4
5
4
8
2
3
1
```

(b)

Figure 1: (a) An input example of 7 digit strings. (b) The output example of (a).

## 3 Command-line Arguments

Please follow the command-line arguments as below.
Usage:

```
python3 programming_hw2.py [input file] [output file]
```

Example:

```
python3 programming_hw2.py input.txt output.txt
```

# 4 Evaluation

To check the correctness of your implementation, 50 public cases are provided for you. In the final evaluation, we will use the 50 public cases and 50 hidden cases (1% for each case) to evaluate your code. The sizes of hidden cases are similar to those of the public cases. The runtime limit for each case is 1 second.

# 5 Submission

- Please compress your programming_hw2.py (.zip or .tar) and upload it to CEIBA.
- Please submit your code before **2 p.m. on November 5, 2018**.

# 6 Hints

- Use dynamic programming. You may need a lookup table to store the results you have computed.
- Be careful of the cases with consecutive 0s (e.g., 10010, 00100, and 11000).