INTRODUCTION TO PROGRAMMING AND DATA STRUCTURES –

ASSIGNMENT 1

BStat I year 2023–2024

Deadline: 15 October, 2023

Total: 5 marks

SUBMISSION INSTRUCTIONS

1. Naming convention for your programs: `BS23xx-progy.py` (assuming `BS23xx` denotes your roll number and `progy` denotes the program number).
2. To submit the solution files (`.py` only), ensure that they are not password protected and upload them (within the stipulated time) through the link: `https://forms.gle/nhBBeuuPQv1yWuao8`.

NOTE: The programs are to be written in Python and should be well commented. All programs should take the required inputs from standard input file and print the desired outputs to standard output file, until otherwise stated.

Q1. Let us define the `folding` of a k-digit natural number, say $N = \sum_{i=0}^{k-1} n_i * 10^i$ ($n_{k-1} \neq 0$), as adding each pair of its digits at a time starting from the two ends going up to the middle. Hence, the folding of N will result into the following number:

$$\sum_{i=0}^{\lfloor k/2 \rfloor} (n_{\lfloor (k-1)/2 \rfloor - i} + n_{\lfloor k/2 \rfloor + i}) * 10^i.$$

A number is `foldable` if none of the digits formed through the aforementioned `folding` process, i.e. $(n_{\lfloor (k-1)/2 \rfloor - i} + n_{\lfloor k/2 \rfloor + i})$ for all applicable i surpasses 9, after the `folding` is attempted. Write a program to take an input number and print all the `foldable` numbers, including the given one, that can be obtained after repeatedly folding it. [3 marks]

Input Format
The input (to be taken from the user) is a number N.

Output Format
The output shows all the `foldable` numbers, including the one given as user input, which can be obtained after repeatedly folding it. If the number is not `foldable` print `NOT APPLICABLE`. If N is an invalid input it shows `INVALID`.

Sample Input 1:

```
10310
```

Sample Output 1:

```
10310
116
72
9
```

Sample Input 2:

```
152734
```

Sample Output 2:

```
152734
589
```

Sample Input 3:

```
555
```

Sample Output 3:

```
NOT APPLICABLE
```

Sample Input 4:

```
-121
```

Sample Output 4:

```
INVALID
```

Sample Input 5:

```
00123
```

Sample Output 5:

```
INVALID
```

Q2. Write a program to take a filename as user input and insert the character comma (i.e., ',') after every i^{th} character within it such that $i = 1, 2, \ldots$ as applicable. [2 marks]

Input Format
The input (to be taken from the user) is a filename, say "input.txt", taken as a string.

Output Format
Output is to be printed within the input file by inserting the character comma (i.e., ',') after every i^{th} character within i is in increasing order starting from 1. If the input file (as taken by name from the user) is not present in the specified path, print `NOT FOUND` on the standard output.

Sample Input 1:

```
input.txt
```

Content of "input.txt" before the execution of the program:

```
I am fine
```

Sample Output 1:
Content of "input.txt" after the execution of the program:

```
I, a,m f,ine
```

Sample Input 2:

```
data.txt
```

Content of "data.txt" before the execution of the program:

```
,,,,,,
```

Sample Output 2:
Content of "data.txt" after the execution of the program:

```
,,,,,,,,,
```

Sample Input 3:

```
fun.txt
```

Content of "fun.txt" before the execution of the program:

```
L        O
```

Sample Output 3:
Content of "fun.txt" after the execution of the program:

```
L,   ,    ,   O,
```

Sample Input 4:

```
p.data
```

Content of "p.data" before the execution of the program:

Sample Output 4:
Content of "p.data" after the execution of the program:

Sample Input 5:

`input.txt`

Content of "input.txt" before the execution of the program:

```
;
```

Sample Output 5:
Content of "input.txt" after the execution of the program:

```
; ,
```