# Lab 8 B

---

## Problem 1: LLM Data Cleaning

In the world of large language models (LLMs), data preprocessing is essential. To demonstrate your skills, you've been given a unique task involving a number that's been distorted with punctuation marks—similar to noisy data that LLMs often encounter.
Your objective is to preprocess this input by cleaning up the punctuation and reconstructing the number accurately, then square it and print the result.
Here's the task:

- Input File: The input number is in a file named number.txt, where each digit of the number has been separated by random punctuation marks.
- Output: Print the square of the number after cleaning up the punctuation marks.

Challenge Details: Since the number of digits is unknown, read the file character by character, ignoring punctuation, and piece together only the digits to form the original number. Treat the end of the file (EOF) as the stopping point.
In this LLM-inspired task, accurate preprocessing is key. Clean the input, compute the square, and output the result to complete the challenge.

### File Format:

- File contains only 1 line of input
- $x$ space seperated digits corresponding to the number
- $x$ will not be provided in the input

### Output Format:

Print 1 line containing the square of the number.

### Constraints:

- $1 \leq x \leq 9$ , $x$ corresponds to the number of digits in the number.
- Each digit ranges from 0 to 9. You are guarenteed that the first digit is not a 0 and the resulting number is a valid number.

### Example File:

```
1,2,0
```

### Example Output 1:

```
14400
```

### Example File:

```
1,1
```

### Example Output 2:

```
121
```

---

## Problem 2: Line and Character Analysis of a Text File

The bots are testing your ability to process text files efficiently. This time, they require a detailed analysis of a file's structure. Your task is to read a text file, count the total number of lines, and for each line, determine the character count.

The input has been provided in a text file named "number.txt". You are required to read this file and print your results to the terminal (stdout).

### Task Description:

- First, count and print the total number of lines in the file.
- Then, print the line number and character count for each line in the file.

### File Format:

- The file may contain multiple lines of text.
- Each line may vary in length and may contain spaces or newline characters.

### Output Format:

- Print the total number of lines as the first line of output.
- Then, for each line in the file, print the line number and the character count on a new line.
- Each line should follow the format: `` `line_number character_count` ``

### Constraints:

- The file contains at least one line.
- Each line can contain any alphanumeric characters and spaces.

### Example File:

```
Hello World!
This is a test.
Count each line.
```

### Example Output:

```
3
1 12
2 15
3 16
```

### Explanation:

- The file contains 3 lines in total, so `3` is printed as the first line.
- Each subsequent line prints the line number and the character count, excluding the newline character.
- For instance, "Hello World!" has 12 characters, and "This is a test." has 15 characters.

---

## Problem 3: First Non-Repeating Word in a File

You have been tasked with developing a program that identifies the first non-repeating word in a file. Each file will contain an arbitrary number of words (the exact number is not specified), and your program should return the first word that does not repeat.

### Formal Statement:

Your program should read from a file "words.txt". The file will contain an arbitrary number of words, you have to determine and output the first non-repeating word in the file. Words are NOT case-sensitive; for example, "mango" and "MaNgO" are considered same.

**Note:** Each file is guaranteed to contain at least one non-repeating word.

### Input Format:

The input will consist of a file with a series of words separated by spaces or newlines. The number of words in the file is not specified.

### Output Format:

Output the first word that does not repeat in the file.

### Explanation:

- The program should identify words that appear only once in the file and output the first such word encountered.

### Sample Test Case:

**Input:**

```
MaNgO grAPe CHeRRy GrApE melOn bAnaNA beRry BERry mAnGO
```

**Output:**

```
cherry
```

---

## Submission Guidelines

Do not rename any files given in the handout. Only write the code in the specified C files in the respective directories.