# Question 1: Getting the Data

Dr. Nam Bo Phan approached QHR with a problem to solve. She has recently learnt of a process *digitSquareSum()* that takes a number and returns the sum of the squares of each of its digits. She asked her assistant Ken Nothad to build a list of results for the numbers 1 to 500.

|  |  |
| --- | --- |
| X | digitSquareSum(x) |
| 1 | 1 |
| 2 | 4 |
| … | … |

Ken got started on the list, but got tired of punching the digits in his calculator. Dr. Phan would like us to produce this list.

## Requirements

* Implement *digitSquareSum()*
* Implement a function *questionOne()* that returns a Java File object for a CSV file that has the following data populated:

1,1

2,4

3,9

…

500,25

# Question 2: Forming the Link

Dr. Phan has found that by performing *digitSquareSum()* multiple times, two numbers get back to their original number:

* 1 -> 1
* 89 -> 145 -> 42 -> 20 -> 4 -> 16 -> 37 -> 58 -> 89

She asked Ken to identify which of the original numbers, between 1 and 500, will arrive at 1 after multiple iterations of performing *digitSquareSum()*. Ken is threatening to quit.

## Requirements

* Implement a function *questionTwo()* that returns the ascending numbers as a comma-separated String (e.g. 1,7,10,…)

# Question 3: Taking on the World

Ken has called QHR to say that Dr. Phan has been rocking in her chair and going on about world domination. Her latest request has Ken fearing for his sanity and his life.

## Requirements

* Implement a method *questionThree()* to determine how many starting numbers between 1 and 10 million inclusively take more than 7 executions of *digitSquareSum()* to reach an 89.
* *questionThree()* should be optimized for speed.