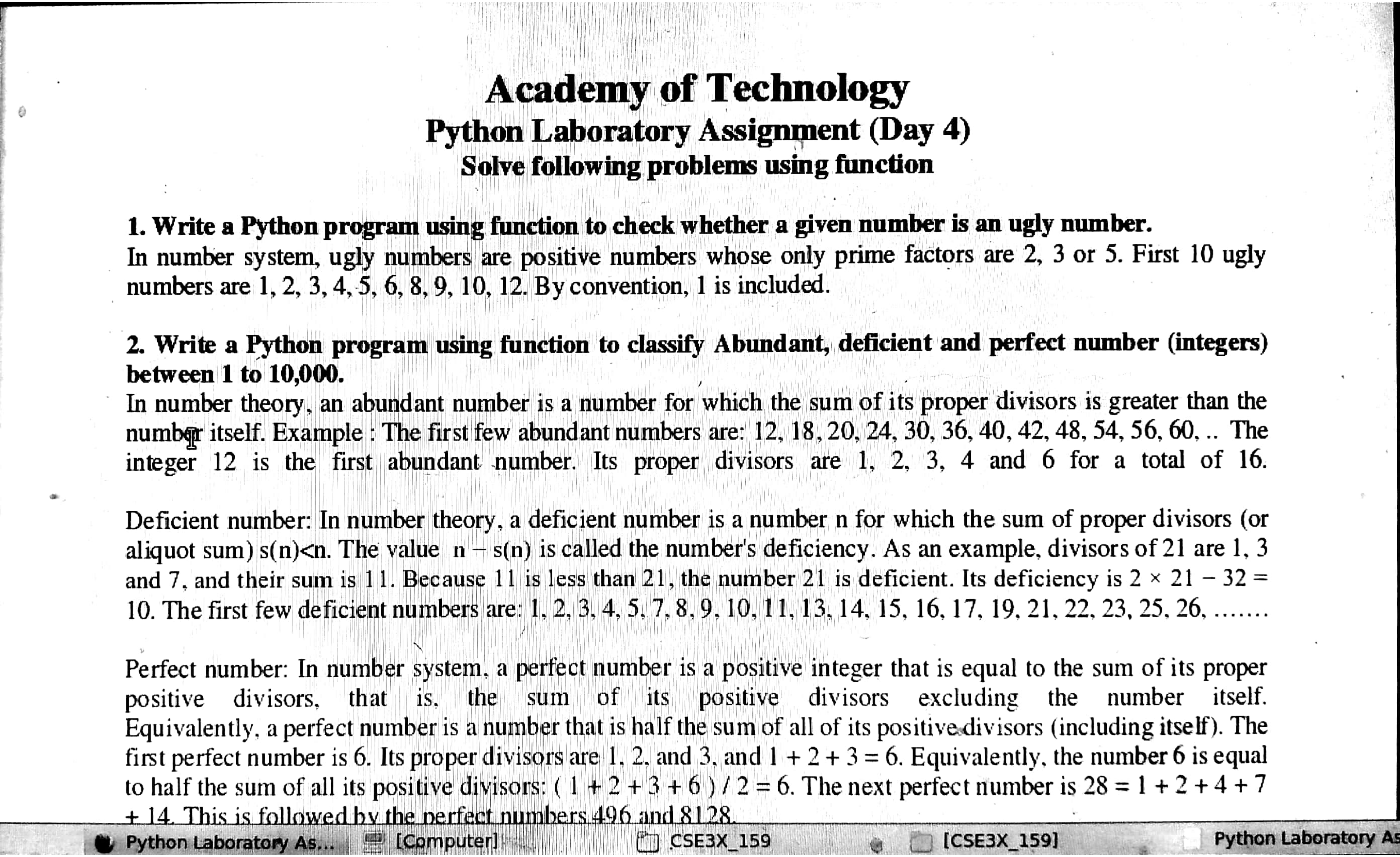


Academy of Technology

Python Laboratory Assignment (Day 4)

Solve following problems using function

1. Write a Python program using function to check whether a given number is an ugly number.
In number system, ugly numbers are positive numbers whose only prime factors are 2, 3 or 5. First 10 ugly numbers are 1, 2, 3, 4, 5, 6, 8, 9, 10, 12. By convention, 1 is included.

2. Write a Python program using function to classify Abundant, deficient and perfect number (integers) between 1 to 10,000.
In number theory, an abundant number is a number for which the sum of its proper divisors is greater than the number itself. Example : The first few abundant numbers are: 12, 18, 20, 24, 30, 36, 40, 42, 48, 54, 56, 60, .. The integer 12 is the first abundant number. Its proper divisors are 1, 2, 3, 4 and 6 for a total of 16.

Deficient number: In number theory, a deficient number is a number n for which the sum of proper divisors (or aliquot sum) s(n)<n. The value n − s(n) is called the number's deficiency. As an example, divisors of 21 are 1, 3 and 7, and their sum is 11. Because 11 is less than 21, the number 21 is deficient. Its deficiency is 2 × 21 − 32 = 10. The first few deficient numbers are: 1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, 19, 21, 22, 23, 25, 26, ……

Perfect number: In number system, a perfect number is a positive integer that is equal to the sum of its proper positive divisors, that is, the sum of its positive divisors excluding the number itself. Equivalently, a perfect number is a number that is half the sum of all of its positive divisors (including itself). The first perfect number is 6. Its proper divisors are 1, 2, and 3, and 1 + 2 + 3 = 6. Equivalently, the number 6 is equal to half the sum of all its positive divisors: (1 + 2 + 3 + 6) / 2 = 6. The next perfect number is 28 = 1 + 2 + 4 + 7 + 14. This is followed by the perfect numbers 496 and 8128.

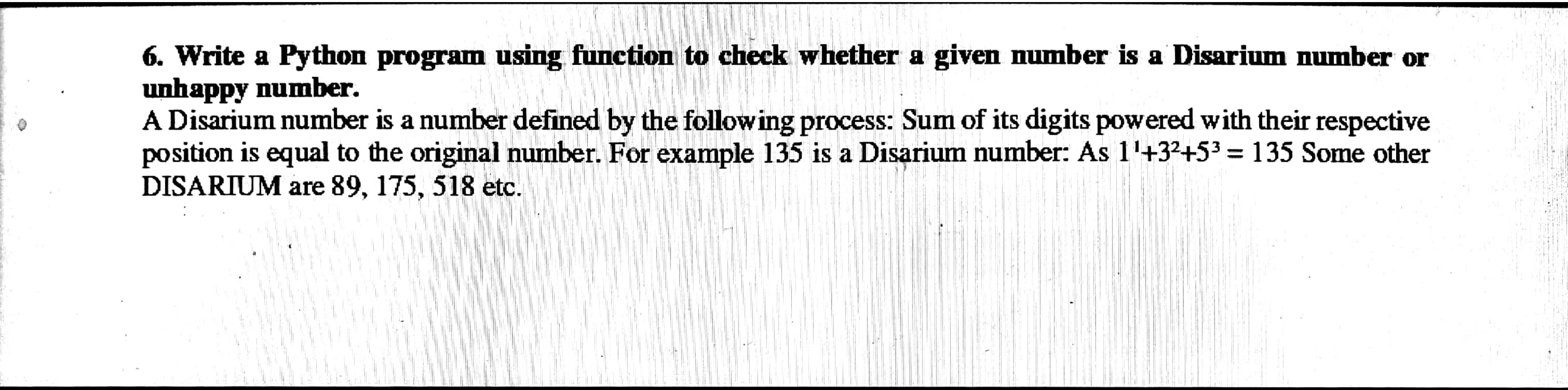
first perfect number is 6. Its proper divisors are 1, 2, and 3, and 1 + 2 + 3 = 6. Equivalently, the number 6 is equal to half the sum of all its positive divisors: (1 + 2 + 3 + 6) / 2 = 6. The next perfect number is 28 = 1 + 2 + 4 + 7 + 14. This is followed by the perfect numbers 496 and 8128.

3. Write a Python program using function to generate and show all Kaprekar numbers less than 1000.
In number theory, a Kaprekar number for a given base is a non-negative integer, the representation of whose square in that base can be split into two parts that add up to the original number again. For instance, 45 is a Kaprekar number, because $45^2 = 2025$ and $20 + 25 = 45$. The first few Kaprekar numbers in base 10 are: 1, 9, 45, 55, 99, 297, 703, 999, 2223, 2728, 4879, …

4. Write a Python program using function to display first 10 lucus numbers.
The Lucas numbers or series are an integer sequence named after the mathematician François Édouard Anatole Lucas, who studied both that sequence and the closely related Fibonacci numbers. Lucas numbers and Fibonacci numbers form complementary instances of Lucas sequences. The sequence of Lucas numbers is: 2, 1, 3, 4, 7, 11, 18, 29, ….

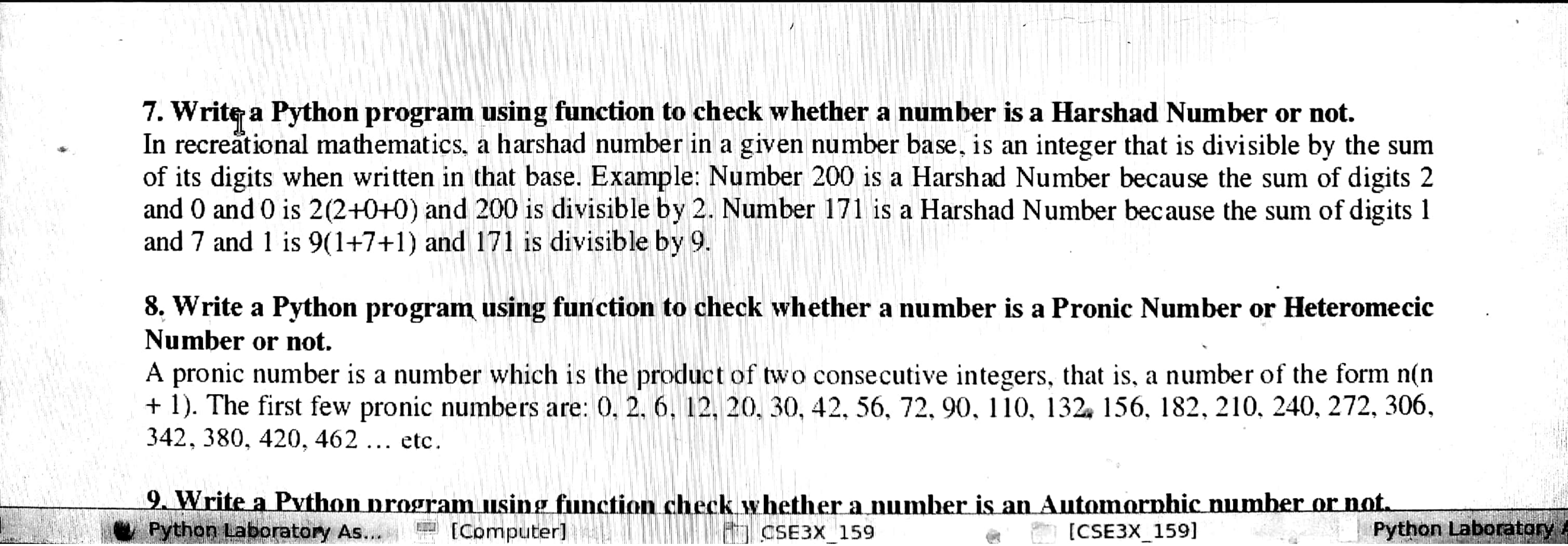
5. Write a Python program using function to find and print the first 10 happy numbers.
Happy number: Starting with any positive integer, replace the number by the sum of the squares of its digits, and repeat the process until the number equals 1, or it loops endlessly in a cycle which does not include 1.
Example: 19 is a happy number
$1^2 + 9^2=82$
$8^2 + 2^2=68$
$6^2 + 8^2=100$
$1^2 + 0^2 + 0^2=1$

6. Write a Python program using function to check whether a given number is a Disarium number or unhappy number.
A Disarium number is a number defined by the following process: Sum of its digits powered with their respective



6. Write a Python program using function to check whether a given number is a Disarium number or unhappy number.

A Disarium number is a number defined by the following process: Sum of its digits powered with their respective position is equal to the original number. For example 135 is a Disarium number: As $1^1+3^2+5^3 = 135$ Some other DISARIUM are 89, 175, 518 etc.



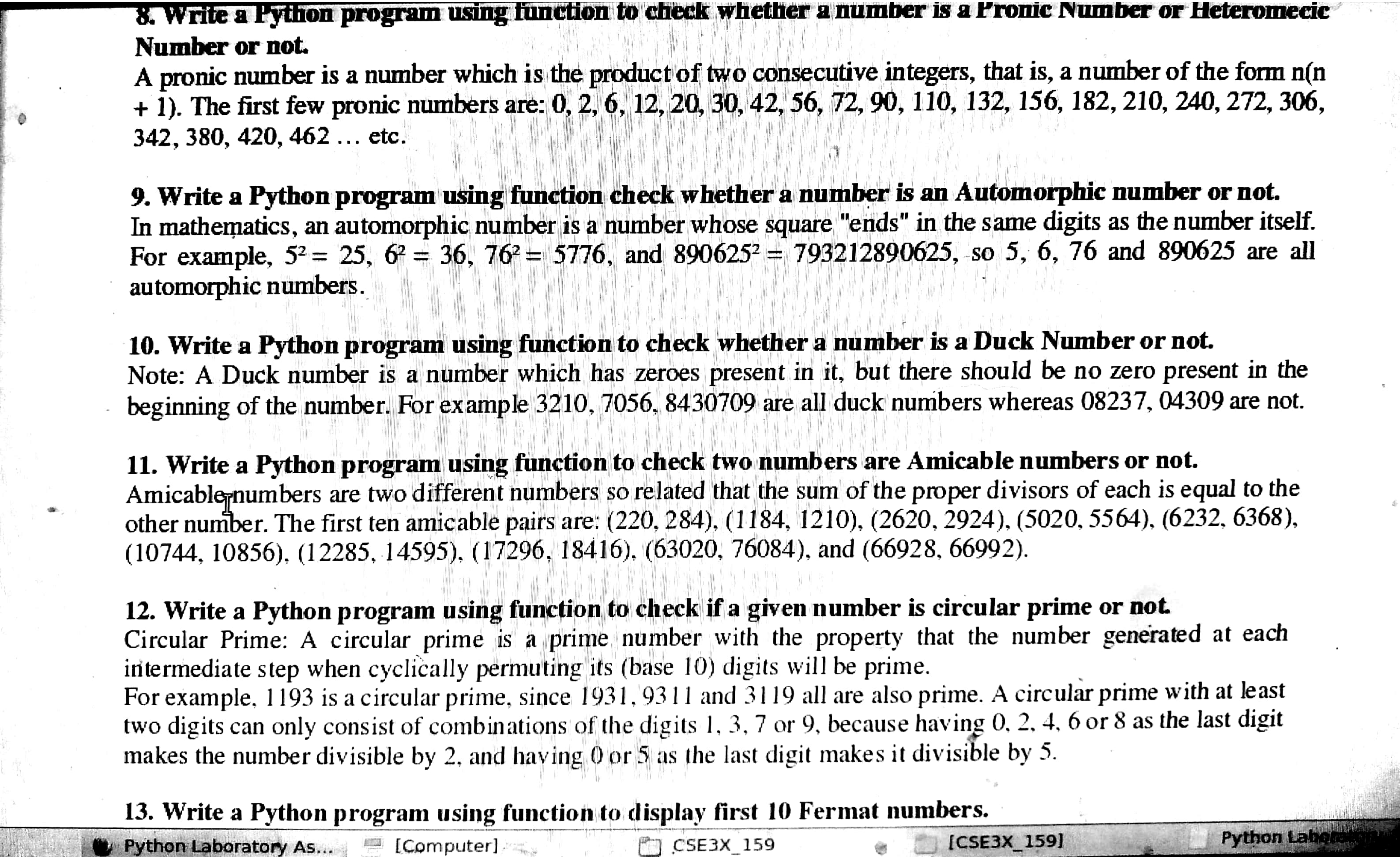
7. Write a Python program using function to check whether a number is a Harshad Number or not.

In recreational mathematics, a harshad number in a given number base, is an integer that is divisible by the sum of its digits when written in that base. Example: Number 200 is a Harshad Number because the sum of digits 2 and 0 and 0 is 2(2+0+0) and 200 is divisible by 2. Number 171 is a Harshad Number because the sum of digits 1 and 7 and 1 is 9(1+7+1) and 171 is divisible by 9.

8. Write a Python program using function to check whether a number is a Pronic Number or Heteromecic Number or not.

A pronic number is a number which is the product of two consecutive integers, that is, a number of the form n(n + 1). The first few pronic numbers are: 0, 2, 6, 12, 20, 30, 42, 56, 72, 90, 110, 132, 156, 182, 210, 240, 272, 306, 342, 380, 420, 462 … etc.

9. Write a Python program using function check whether a number is an Automorphic number or not.



8. Write a Python program using function to check whether a number is a Pronic Number or Heteromecic Number or not.

A pronic number is a number which is the product of two consecutive integers, that is, a number of the form n(n + 1). The first few pronic numbers are: 0, 2, 6, 12, 20, 30, 42, 56, 72, 90, 110, 132, 156, 182, 210, 240, 272, 306, 342, 380, 420, 462 … etc.

9. Write a Python program using function check whether a number is an Automorphic number or not.

In mathematics, an automorphic number is a number whose square "ends" in the same digits as the number itself. For example, $5^2 = 25$, $6^2 = 36$, $76^2 = 5776$, and $890625^2 = 793212890625$, so 5, 6, 76 and 890625 are all automorphic numbers.

10. Write a Python program using function to check whether a number is a Duck Number or not.

Note: A Duck number is a number which has zeroes present in it, but there should be no zero present in the beginning of the number. For example 3210, 7056, 8430709 are all duck numbers whereas 08237, 04309 are not.

11. Write a Python program using function to check two numbers are Amicable numbers or not.

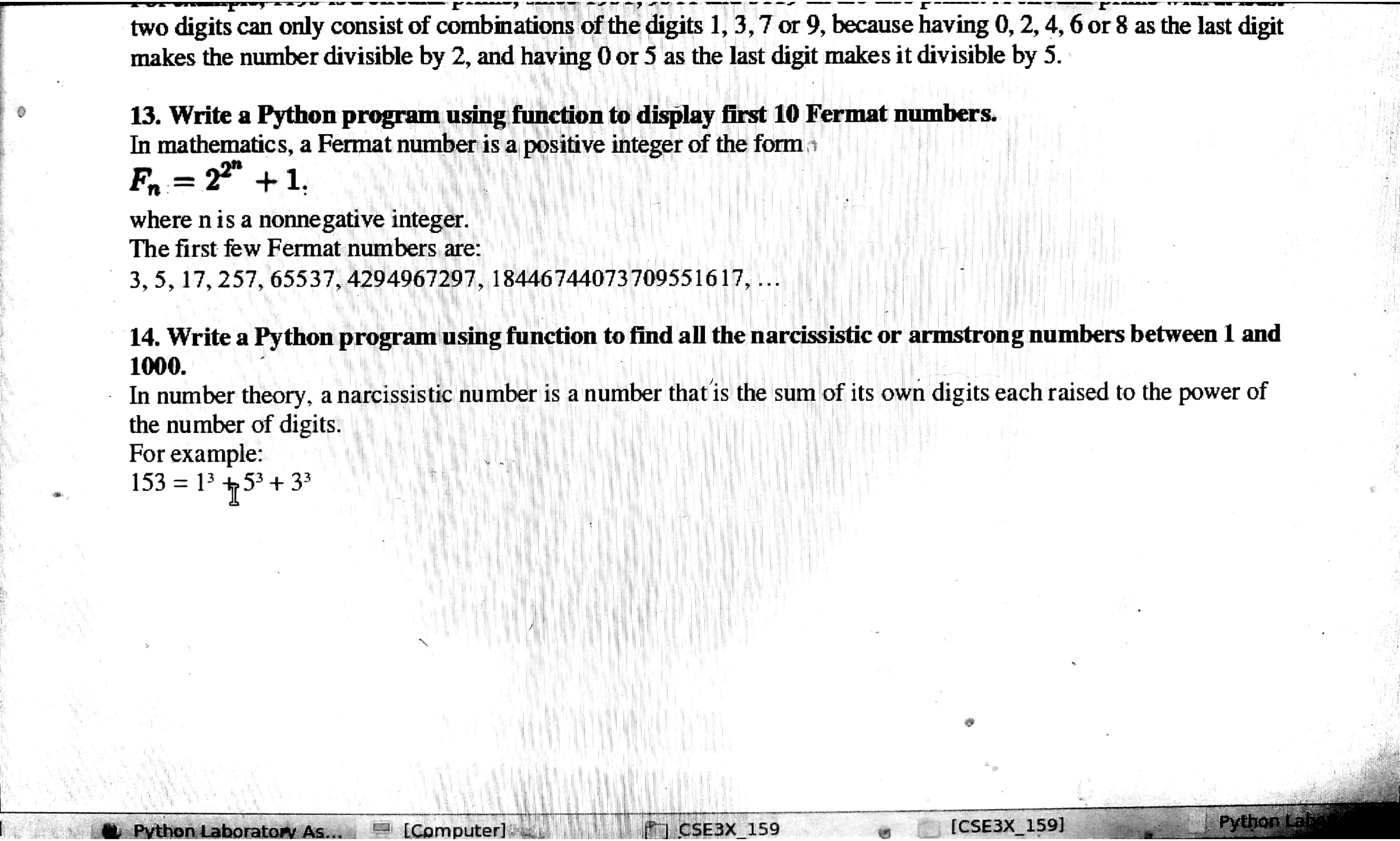
Amicable numbers are two different numbers so related that the sum of the proper divisors of each is equal to the other number. The first ten amicable pairs are: (220, 284), (1184, 1210), (2620, 2924), (5020, 5564), (6232, 6368), (10744, 10856), (12285, 14595), (17296, 18416), (63020, 76084), and (66928, 66992).

12. Write a Python program using function to check if a given number is circular prime or not.

Circular Prime: A circular prime is a prime number with the property that the number generated at each intermediate step when cyclically permuting its (base 10) digits will be prime.

For example, 1193 is a circular prime, since 1931, 9311 and 3119 all are also prime. A circular prime with at least two digits can only consist of combinations of the digits 1, 3, 7 or 9, because having 0, 2, 4, 6 or 8 as the last digit makes the number divisible by 2, and having 0 or 5 as the last digit makes it divisible by 5.

13. Write a Python program using function to display first 10 Fermat numbers.



two digits can only consist of combinations of the digits 1, 3, 7 or 9, because having 0, 2, 4, 6 or 8 as the last digit makes the number divisible by 2, and having 0 or 5 as the last digit makes it divisible by 5.

13. Write a Python program using function to display first 10 Fermat numbers.

In mathematics, a Fermat number is a positive integer of the form

$$F_n = 2^{2^n} + 1,$$

where n is a nonnegative integer.

The first few Fermat numbers are:

3, 5, 17, 257, 65537, 4294967297, 18446744073709551617, ...

14. Write a Python program using function to find all the narcissistic or armstrong numbers between 1 and 1000.

In number theory, a narcissistic number is a number that is the sum of its own digits each raised to the power of the number of digits.

For example:

$153 = 1^3 + 5^3 + 3^3$