TG ACADEMIC RESOURCE SERIES

PYTHON PROGRAMMING

Automorphic Numbers

Mathematical Logic, Algorithm & Worked Dry Run

SUBJECT	Python Programming / Number Logic
LEVEL	Class 11–12 / BTech / BCA / MCA / BSc
RESOURCE TYPE	Sample learning note
PUBLICATION ID	PY-001 | Version 1.0

Prepared by Tanmoy Gupta

Computer Science Educator & Industry Professional

This sample uses arithmetic operations to determine whether the square of a number ends with the original number. The program uses digit counting, integer division, modulo and a loop—without string conversion or a user-defined function.

A focused preview of a structured programming-learning approach, not a complete collection of number programs.

1. Learning Objectives

After studying this note, you should be able to:

- Define an Automorphic Number.
- Explain the mathematical condition using powers of 10 and the modulo operator.
- Count the digits of a positive integer using a loop.
- Translate the mathematical test into a Python program.
- Trace the program using a tabular dry run.

2. What Is an Automorphic Number?

A positive integer is called an **Automorphic Number** if its square ends with the number itself.

Examples

25: $25^2 = 625$. The last two digits are 25, so 25 is Automorphic.
76: $76^2 = 5776$. The last two digits are 76, so 76 is Automorphic.
13: $13^2 = 169$. The last two digits are 69, not 13, so 13 is not Automorphic.

3. Mathematical Condition

Let n be a positive integer containing d digits. The last d digits of n^2 are obtained by calculating the remainder when n^2 is divided by 10^d.

Therefore, n is Automorphic exactly when:

$$n^2 \bmod 10^d = n$$

For n = 25, the number of digits d = 2. Therefore $10^d = 100$.
$25^2 \bmod 100 = 625 \bmod 100 = 25$.
Since the remainder equals the original number, 25 is Automorphic.

Why does modulo work?

For any integer, dividing by 10^d leaves a remainder equal to its last d decimal digits. For example, 625 mod 100 = 25. This is precisely the suffix test required by the definition.

4. Algorithm

1. Read a positive integer n.
2. Store n in a temporary variable.
3. Count its digits by repeatedly dividing the temporary value by 10 using integer division.
4. Calculate p = 10■, where d is the digit count.
5. Calculate square = n × n.
6. Calculate remainder = square % p.
7. If remainder equals n, the number is Automorphic; otherwise, it is not.

5. Python Program (Arithmetic Approach)

This version uses no string conversion and defines no user-defined function. It is suitable for understanding the mathematical process and loop-based digit extraction.

```
n = int(input("Enter a positive integer: ")) if n <= 0: print("Please enter a positive integer.") else: temp = n digits = 0 while temp > 0: digits = digits + 1 temp = temp // 10 divisor = 10 ** digits square = n * n last_part = square % divisor print("Square =", square) print("Last", digits, "digit(s) =", last_part) if last_part == n: print(n, "is an Automorphic Number.") else: print(n, "is not an Automorphic Number.")
```

Sample Output

```
Enter a positive integer: 25
Square = 625
Last 2 digit(s) = 25
25 is an Automorphic Number.
```

Program note

The loop counts digits without converting the number to text. The expression temp // 10 removes the rightmost digit at each iteration. The expression 10 ** digits constructs the place-value divisor needed for the modulo test.

6. Complete Dry Run: Check 9376

We will trace the program for n = 9376.

Step A — Count the digits

Iteration	temp before	digits after increment	temp after // 10
1	9376	1	937
2	937	2	93
3	93	3	9
4	9	4	0

The loop executes four times. Therefore d = 4 and divisor = 10■ = 10000.

Step B — Apply the Automorphic test

$9376^2 = 87,909,376$
$87,909,376 \% 10,000 = 9,376$
Remainder = original number = 9,376

Step C — Decision table

Variable / expression	Value
n	9376
digits	4
divisor = 10 ** digits	10000
square = n * n	87909376
last_part = square % divisor	9376
last_part == n	True

Conclusion: 9376 is an Automorphic Number because its square ends with the original four-digit number.

A quick non-example

For n = 13, d = 2 and $13^2 \% 100 = 169 \% 100 = 69$. Since $69 \neq 13$, the number is not Automorphic.

7. Check Your Understanding

Try these questions before checking the suggested answers.

Practice Questions

Q1. Explain why 76 is an Automorphic Number using the modulo condition.

Q2. For n = 13, determine d, 10■ and n^2 mod 10■. State the result.

Q3. What is the purpose of the statement **temp = temp // 10** inside the digit-counting loop?

Q4. Write the mathematical condition for a positive d-digit integer n to be Automorphic.

Suggested Answers (brief)

A1. $76^2 = 5776$, d = 2 and 5776 mod 100 = 76. The remainder equals 76, so it is Automorphic.

A2. d = 2, 10■ = 100 and 13^2 mod 100 = 169 mod 100 = 69. Since $69 \neq 13$, it is not Automorphic.

A3. Integer division by 10 removes the last decimal digit, allowing the loop to count how many digits the original number contains.

A4. n^2 mod 10■ = n, where d is the number of decimal digits in n.

8. Important Observations

• This sample program accepts positive integers. Zero and negative inputs are rejected.
• The arithmetic approach avoids converting the number to a string.
• Python integers can represent very large values, but the size of the number still affects the practical amount of computation.
• The test uses the number of decimal digits of the original positive integer.

Where to Go Next

A broader Python number-programming module may include Palindrome, Armstrong, Perfect, Strong, Harshad, Prime and other number-property problems. Those topics can be taught individually through mathematical derivation, algorithm design, code tracing and progressively harder exercises.

In guided classes: students practise deriving the condition first, translating it into code, tracing each loop iteration, testing boundary cases and explaining why the algorithm works.

About this resource
TG Academic Resource Series | PY-001 | Version 1.0
Prepared by Tanmoy Gupta — Computer Science Educator & Industry Professional

Original educational material prepared for student learning and revision. Redistribution or commercial reproduction without permission is not authorised.