Recursion is often described as a problem-solving technique where a function calls itself in its own definition.

To gain a clearer understanding, let's delve into some examples.

Consider the task of writing code to determine the sum of the digits of a positive integer through recursion. Imagine the user inputs the positive integer 456. Mathematically, the sum of digits for this number (4+5+6) equals 15. Now, let's see how the Python code employs recursion to compute this sum: Let's start by creating a function:

(In Python)

```
def sum_of_digits(n): if n < 10:

return n else:

a=n % 10 + sum_of_digits(n // 10) return a

num = int(input('Enter the number:\n')) result = sum_of_digits(num)

print(f"The sum of digits in {num} is {result}")
```

In this Python code, there's a function named sum_of_digits that calculates the sum of the digits of a given number n. The sum_of_digits function takes an integer n as input and checks if it is less than 10. If n is a single-digit number, the function returns n itself.

If n is greater than or equal to 10, the function enters the else block. It calculates the variable a by adding the last digit of n (found using n % 10) to the result of calling sum_of_digits on the remaining part of the number (n // 10). This recursive process continues until n becomes a single-digit number.

Where n % 10 expression represents the remainder when n is divided by 10. In simpler terms, it gives you the last digit of the number. And n // 10 expression represents the integer division of n by 10. It essentially removes the last digit from the number.

So suppose n=345

Then n % 10=345 % 10=5 and n // 10= 345 // 10=34

Moving on let’s say if the user enters the number 456, the code will output the sum of its digits, which is calculated as 4 + 5 + 6 = 15.

Let's go through the recursive calls at each step for the number 456:

Initial Call:

n = 456

The condition n < 10 is False, so it goes to the else block.

a = 6 (last digit) + sum_of_digits(456 // 10)

First Recursive Call:

n = 45

The condition n < 10 is False, so it goes to the else block. a = 5 (last digit) + sum_of_digits(45 // 10)

Second Recursive Call:

n = 4

The condition n < 10 is True, so it returns n itself, which is 4.

Back to First Recursive Call:

Now, a = 5 + 4 = 9

This value is used in the original call.

Back to Initial Call:

Now, a = 6 + 9 = 15

The final result is 15, representing the sum of the digits in the original number 456. So the output is :

The sum of digits in 456 is 15

Let's explore another example of using recursion to print a countdown in Python. In this example, we'll create a function called countdown that takes a positive integer as input and prints a countdown from that number to 1. Here's the Python code:

```
def countdown(n): if n <= 0:

print("Happy New Year!") else:

print(n) countdown(n - 1)

# Get user input for countdown start
```

```
start_number = int(input("Enter the starting number for the countdown: "))

# Call the countdown function countdown(start_number)
```

In this code, it defines a function named countdown that takes an integer n as its parameter. Inside the function, there's an if statement that checks if the input n is less than or equal to 0.

If true, it means the countdown has reached or gone below 0, and the function prints "Happy New Year!" as a special message. If the base case is not met (i.e., n is greater than 0), the code proceeds to the ‘else’ block. It prints the current value of n using print(n).

Then, it makes a recursive call to the countdown function with the decremented value n - 1. The code prompts the user to enter a starting number for the countdown using the input function. Finally, the countdown function is called with the user-provided starting number as an argument.

Suppose the user enters 10, the output will be:

```
10

9

8

7

6

5

4

3

2

1

Happy New Year!
```

So the recursive nature of the function ensures that it keeps calling itself with a decremented value until the base case is met (when n is less than or equal to 0), leading to the countdown effect. The final message "Happy New Year!" is printed when the countdown reaches its conclusion.