Here's a C program that calculates the sum of the digits of a number using recursion:

c

#include <stdio.h>

// Function to find the sum of digits using recursion

int sum\_of\_digits(int n) {

if (n == 0)

return 0;

return (n % 10) + sum\_of\_digits(n / 10);

}

int main() {

int number;

// Input

printf("Input number: ");

scanf("%d", &number);

// Output

printf("Sum of digits: %d\n", sum\_of\_digits(number));

return 0;

}

### Explanation:

- \*Base Case\*: The recursion stops when n becomes 0. At that point, the function returns 0.

- \*Recursive Step\*: In each recursive call, the last digit (n % 10) is added to the sum of the remaining digits (n / 10).

### Example:

For the input 1234:

- The recursion will work as follows:

- sum\_of\_digits(1234) = 4 + sum\_of\_digits(123)

- sum\_of\_digits(123) = 3 + sum\_of\_digits(12)

- sum\_of\_digits(12) = 2 + sum\_of\_digits(1)

- sum\_of\_digits(1) = 1 + sum\_of\_digits(0)

- sum\_of\_digits(0) = 0 (base case)

- The result is 4 + 3 + 2 + 1 = 10.

### Output:

Input number: 1234

Sum of digits: 10

This program efficiently calculates the sum of the digits of a number using recursion.