Sum of Digits of a Five-Digit Number

The modulo operator `%` returns the remainder of a division. For example, `4 % 3 = 1` and `12 % 10 = 2`. The division operator `/` returns the integer quotient when dividing integers, truncating the decimal part. For example, `5 / 3 = 1`.

To extract the last digit of a number, use modulo 10. The task is to sum the digits of a given five-digit integer.

Solution in C:

```c
#include <stdio.h>

int main() {
    int n, sum = 0;
    scanf("%d", &n);

    while (n > 0) {
        sum += n % 10;  // Add last digit to sum
        n /= 10;        // Remove last digit
    }

    printf("%d\n", sum);
    return 0;
}
```

Listing 1: Sum of digits of a five-digit integer

Explanation:

- Use `n % 10` to get the last digit of `n`.
- Use `n /= 10` to remove the last digit.
- Repeat until `n` becomes zero.
- Accumulate the sum of the digits in `sum`.

Common Problem Ideas and Extensions

Digit extraction using modulo and division is a common technique that can be adapted to many related problems:

- **Counting specific digits:** Count how many times a particular digit appears in a number.
- **Checking properties of digits:** For example, determine if a number is *palindromic* by comparing digits.
- **Digit manipulation:** Reverse the digits of a number or rotate digits.

- **Digit-based arithmetic:** Calculate the product of digits, find the maximum/minimum digit, or calculate digital roots.
- **Validity checks:** Verify if the sum of digits is divisible by a certain number (useful in divisibility rules).

How to Make It Harder

- **Larger inputs:** Process very large numbers that don't fit in standard integer types, requiring the number to be processed as a string.
- **Negative numbers:** Handle negative input correctly, summing digits ignoring the sign.
- **Variable-length numbers:** Instead of fixed five-digit numbers, accept any length and process accordingly.
- **Multiple operations:** Combine digit extraction with other operations, such as sorting digits or finding the nth digit from the right/left.
- **Digit frequency histograms:** Count frequency of each digit and analyze patterns.
- **Digit-based dynamic programming:** For example, count numbers within a range that satisfy certain digit sum or digit pattern properties.

These problems encourage deeper understanding of numeric manipulation and can be tackled by combining simple digit operations with loops, conditionals, arrays, and sometimes recursion or DP.