1. Find the 2nd smallest digit in a number

```
def second_smallest_digit(number):
    unique_digits = sorted(set(int(d) for d in str(abs(number))))
    if len(unique_digits) < 2:
        return "No second smallest digit"
    return unique_digits[1]

# Example usage
print(second_smallest_digit(12345))
```

Expected Output: 2

2. Find the kth largest digit in a number

```
def kth_largest_digit(number, k):
    unique_digits = sorted(set(int(d) for d in str(abs(number))), reverse=True)

    if k < 1 or k > len(unique_digits):
        return "Invalid value of k"
    return unique_digits[k - 1]

# Example usage
print(kth_largest_digit(12345, 3))
```

Expected Output: 3

3. check if a number is prime or not:

```
public class PrimeCheck {

    // Method to check if a number is prime
    public static boolean isPrime(int num) {
```

```
        // Handle edge cases
        if (num <= 1) {
            return false; // Numbers less than or equal to 1 are not prime
        }

        // Check divisibility from 2 to the square root of num
        for (int i = 2; i <= Math.sqrt(num); i++) {
            if (num % i == 0) {
                return false; // If divisible, it's not prime
            }
        }
        return true; // If no divisors found, the number is prime
    }

    public static void main(String[] args) {
        // Default number to check
        int number = 29;

        // Check if the number is prime
        if (isPrime(number)) {
            System.out.println(number + " is a prime number.");
        } else {
            System.out.println(number + " is not a prime number.");
        }
    }
}
```

Expected Output:

```
29 is a prime number.
```

4. Find all Prime Factors of a Number

```
import java.util.ArrayList;
import java.util.List;

public class PrimeFactors {

    public static List<Integer> primeFactors(int n) {
        List<Integer> factors = new ArrayList<>();

        for (int i = 2; i <= n / i; i++) {
            while (n % i == 0) {
                factors.add(i);
                n /= i;
            }
        }
        if (n > 1) factors.add(n); // If n is a prime number greater than 1

        return factors;
    }

    public static void main(String[] args) {
        int number = 56;
        System.out.println(primeFactors(number));  // Output: [2, 2, 2, 7]
    }
}
```

Expected Output: [2, 2, 2, 7]

5. Check Whether a Number is Positive or Negative

```
#include <stdio.h>

void main() {
    // Default number
```

```
    int number = 5; // You can change this value to any number you want to check

    if (number >= 0)
        printf("%d is a positive number \n", number);
    else
        printf("%d is a negative number \n", number);
}
```

Expected Output:

5 is a positive number

6. Find the largest of three numbers

```
#include <stdio.h>

int main() {
    // Default values for a, b, and c
    int a = 10, b = 20, c = 15; // You can change these values to any numbers you want

    if (a > b && a > c)
        printf("Biggest number is %d\n", a);
    if (b > a && b > c)
        printf("Biggest number is %d\n", b);
    if (c > a && c > b)
        printf("Biggest number is %d\n", c);

    return 0;
}
```

Expected Output:

Biggest number is 20