1. Find the kth smallest digit in a number:

```
def kth_smallest_digit(number, k):
    unique_digits = sorted(set(int(d) for d in str(abs(number))))
    if k < 1 or k > len(unique_digits):
        return "Invalid value of k"
    return unique_digits[k - 1]

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

Expected Output: 3

2. Rotate the digits in a number:

```
def rotate_number(number, k):
    number_str = str(abs(number))
    k = k % len(number_str)  # Handle cases where k > length of number
    rotated_number_str = number_str[-k:] + number_str[:-k]
    return int(rotated_number_str) if number >= 0 else -int(rotated_number_str)

# Example usage
print(rotate_number(12345, 2))
```

Expected Output: 45123

3. Dot Product of Two Vectors:

```
public class VectorOperations {

    public static int dotProduct(int[] a, int[] b) {
        if (a.length != b.length) {
            throw new IllegalArgumentException("Vectors must be of the same dimension");
```

```
        }
        int dotProduct = 0;
        for (int i = 0; i < a.length; i++) {
            dotProduct += a[i] * b[i];
        }
        return dotProduct;
    }

    public static void main(String[] args) {
        int[] vectorA = {1, 2, 3};
        int[] vectorB = {4, 5, 6};
        System.out.println("Dot Product: " + dotProduct(vectorA, vectorB));  // Output: 32
    }
}
```

Expected Output:

Dot Product: 32

4. Find distance between 2 points:

```
public class Distance {

    // Method to calculate 2D distance
    public static double distance2D(double x1, double y1, double x2, double y2) {
        return Math.sqrt(Math.pow(x2 - x1, 2) + Math.pow(y2 - y1, 2));
    }

    // Method to calculate 3D distance
    public static double distance3D(double x1, double y1, double z1, double x2, double y2, double z2)
{
        return Math.sqrt(Math.pow(x2 - x1, 2) + Math.pow(y2 - y1, 2) + Math.pow(z2 - z1, 2));
    }
```

```
    public static void main(String[] args) {
        // Example usage for 2D points (x1, y1) and (x2, y2)
        double x1 = 1.0, y1 = 2.0, x2 = 4.0, y2 = 6.0;
        System.out.println("2D Distance: " + distance2D(x1, y1, x2, y2));  // Output: 5.0

        // Example usage for 3D points (x1, y1, z1) and (x2, y2, z2)
        double z1 = 3.0, z2 = 7.0;
        System.out.println("3D Distance: " + distance3D(x1, y1, z1, x2, y2, z2));  // Output:
5.385164807134504
    }
}
```

Expected Output:

2D Distance: 5.0

3D Distance: 6.4031242374328485

5. Find the Number of Integers Divisible by 5:

```
#include <stdio.h>

int main() {
    // Default value for n
    int n = 50;
    int count = 0;

    // Counting numbers divisible by 5
    for (int i = 1; i <= n; i++) {
        if (i % 5 == 0) {
            count++;
        }
    }

    // Output the count of numbers divisible by 5
```

```
    printf("Number of integers divisible by 5 from 1 to %d: %d\n", n, count);

    return 0;
}
```

Expected Output:

Number of integers divisible by 5 from 1 to 50: 10

6. Sum of Digits:

```
#include <stdio.h>

int main(void) {
    // Default value for num
    int num = 12345;
    int sum = 0, rem;

    // Keep dividing until the number is not zero
    while (num != 0) {
        rem = num % 10;
        sum = sum + rem;
        num = num / 10;
    }

    printf("Sum of digits of the number is %d\n", sum);
    return 0;
}
```

Expected Output:

Sum of digits of the number is 15