Introduction to Algorithms

Sheet 1

Methods and Recursion: Revision

Exercise 1: Factorial Calculator

Write a recursive method to calculate the factorial of a given number. The factorial of a non-negative integer n, denoted as $n!$, is the product of all positive integers less than or equal to n. The recursive formula for factorial is given by:

$$n! = \begin{cases} 1 & \textit{if } n = 0, \\ n \times (n-1)! & \textit{if } n > 0 \end{cases} \tag{1}$$

```
Sample Input: 5
Expected Output: 120
```

Exercise 2: Fibonacci Series

Implement a recursive method to generate the nth Fibonacci number. The Fibonacci series is defined by the recurrence relation:

$$F(n) = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ F(n-1) + F(n-2) & \text{if } n > 1 \end{cases} \tag{2}$$

```
Sample Input: 6
Expected Output: 8 (0, 1, 1, 2, 3, 5, 8)
```

Exercise 3: Power Function

Create a recursive method to calculate the power of a number (baseˆexponent).

```
Sample Input:
base = 2
exponent = 3
Expected Output: 8
```

Exercise 4: Sum of Digits

Write a recursive method to find the sum of digits of a given number.

```
Sample Input: 12345
Expected Output: 15
```

Exercise 5: GCD (Greatest Common Divisor)

Implement a recursive method to find the GCD of two numbers. The greatest common divisor (GCD) of two integers a and b is calculated using Euclid's algorithm. The recursive rule for GCD is given by:

$$\gcd(a, b) = \begin{cases} a & \text{if } b = 0 \\ \gcd(b, a \mod b) & \text{if } b \neq 0 \end{cases} \tag{3}$$

```
Sample Input: 24, 36
Expected Output: 12
```

Exercise 6: Palindrome Check

Write a recursive method to check if a given string is a palindrome.

```
Sample Input: "level"
Expected Output: true
```

Exercise 7: Array Sum

Create a recursive method to find the sum of elements in an integer array.

```
Sample Input: [2, 4, 6, 8, 10]
Expected Output: 30
```

Exercise 8: Binary Search

Implement a recursive method for binary search on a sorted array.

```
Sample Input: [1, 2, 3, 4, 5, 6, 7, 8, 9],
target = 5
Expected Output: 4 (index of the target element)
```

Exercise 9: Reverse String

Write a recursive method to reverse a given string.

```
Sample Input: "hello"
Expected Output: "olleh"
```

Exercise 10: Tower of Hanoi

Implement the Tower of Hanoi problem using recursion.

```
Sample Input:
number of disks = 3
source = A
auxiliary = B
destination = C

Expected Output:
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 3 from A to C
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
```

Coin Changing Problem

You are given an array representing different coin denominations and a total amount of money. The goal is to find the minimum number of coins needed to make up that amount. Assume an unlimited supply of coins of each denomination.

Input

- An array `coins` representing the coin denominations, where each coin denomination is a positive integer.
- An integer `amount` representing the total amount of money to make up.

Output

- An integer representing the minimum number of coins needed to make up the amount.
- If it's not possible to make up the amount using the given coin denominations, return -1.

Example

```
Input:
coins = [1, 2, 5]
amount = 11

Output:3
```

Explanation: 11 = 5 + 5 + 1

Note

- You can assume that the input array `coins` does not contain duplicates.
- The order of coin denominations in the array is arbitrary.