1. A procedure that calls itself is said to as recursive. This can be used to tackle issues where a major issue is broken down into smaller and smaller issues until they are straightforward enough to be solved directly.

2. Repetition of a set of instructions until the intended result is attained is an approach known as iteration. A loop, a control structure that enables code to be executed repeatedly, can be used to accomplish this.

3. Recursion and iteration can both be used to solve the mathematical functions factor and Fibonacci.

The product of all positive integers smaller than or equal to a specific number is known as a factorial. The factorial of 5, for instance, is 120 since 120 = 1 * 2 * 3 * 4 * 5.

Recursive: To solve the problem of factorial, define the factorial of a given number, n, as the sum of the factorial of n and one. The factorial of 5, for instance, can be calculated as follows:

```
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n - 1)
```

Iterative: The iterative solution to factorial is to use a loop to calculate the product of all the positive integers less than or equal to n. For example, the factorial of 5 can be calculated as follows:

```
def factorial(n):
    result = 1
    for i in range(1, n + 1):
        result *= i
    return result
```

Fibonacci: is a sequence of numbers where each number is the sum of the two numbers before it. The first two numbers in the Fibonacci sequence are 0 and 1, and the sequence continues as follows: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

Recursive: The recursive solution to Fibonacci is to define the Fibonacci number for a number n as the sum of the Fibonacci numbers for n - 1 and n - 2. For example, the Fibonacci number for 5 can be calculated as follows:

```
def fibonacci(n):
    if n == 0 or n == 1:
        return n
    else:
        return fibonacci(n - 1) + fibonacci(n - 2)
```

Iterative: The iterative solution to Fibonacci is to use a loop to calculate the sum of the Fibonacci numbers for n - 1 and n - 2. For example, the Fibonacci number for 5 can be calculated as follows:

```
def fibonacci(n):
    f0 = 0
    f1 = 1
    for i in range(2, n + 1):
        f2 = f0 + f1
        f0 = f1
        f1 = f2
    return f1
```