It looks like you have the same set of functions from the previous JavaScript code, but this time it's organized into a single file named core.js. Each function serves a specific purpose, and they cover various topics in JavaScript, including basic arithmetic operations, recursion, bitwise operators, and functional programming concepts.

Here's a quick recap of the functions:

1. **isInteger(n):** Checks if a given number is an integer using bitwise operators.
2. **even():** Returns an array of even numbers from 2 to 20.
3. **sumTo(n):** Calculates the sum of numbers up to a given value using a loop.
4. **recSumTo(n):** Calculates the sum of numbers up to a given value using recursion.
5. **factorial(n):** Calculates the factorial of a given number using a loop.
6. **isBinary(n):** Checks if a given number is a power of two using bitwise operations.
7. **fibonacci(n):** Calculates the N-th Fibonacci number using recursion.
8. **getOperationFn(initialValue, operatorFn):** Returns a function that performs a given operation on its argument. If no operation is provided, it returns a function that always returns the initial value.
9. **sequence(start, step):** Returns a function that generates an arithmetic sequence with a specified starting value and step size.
10. **deepEqual(firstObject, secondObject):** Checks if two objects are deeply equal, considering nested properties.

The last part of the code checks if the module system is available (using typeof module === 'object') and exports the functions if it is, making the code suitable for use in a Node.js environment or with other module systems.