Part A: Recursion

1. Fibonacci Sequence

The Fibonacci Sequence is a sequence (a list of numbers) that never ends. The first few numbers are:

f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	...
0	1	1	2	3	5	8	13	21	34	...

The rule that makes the Fibonacci Sequence is “the next number is the sum of the previous two”. This kind of rule is sometimes called a recurrence relation. Mathematically, this is written as:

$$f_n = f_{n-1} + f_{n-2}$$

Provide two ways to write the Fibonacci sequence program in C:

- Fibonacci Series without recursion
- Fibonacci Series using recursion

2. Factorial

The factorial of n is the product of all positive descending integers. Factorial of n is denoted by n!, for example:

5! = 5*4*3*2*1 = 120

3! = 3*2*1 = 6

Provide two ways to calculate the factorial of a given number in C:

- without recursion
- using recursion

3. Invert numbers

Implement a function that reverses an integer. That is, if the input number is 12345, the result should be 54321. The function prototype can be as follow:

```
int reverseNumber(int num);
```

Provide two ways to calculate the reverse number in C:

- without recursion
- using recursion

4. Binary search

Given a sorted array of n integers and a target value, determine if the target exists in the array in logarithmic time using the binary search algorithm. If a target exists in the array, print the index of it.

Provide two ways to binary search for a given number in C:

- without recursion
- using recursion

5. Palindrome

A palindrome number is a number that remains the same when digits are reversed. For example, the number 12321 is a palindrome number, but 1451 is not a palindrome number.

Write a recursive function in C to check if a given number is a palindrome number.

6. Sum of natural numbers in a range

Write a recursive function in C programming to find sum of all natural numbers between 1 to n.

7. Merge Sort

a. Analyze the algorithm we've seen before: https://www.geeksforgeeks.org/merge-sort/
b. Analyze the code, try to understand it as now you know recursive functions.
c. Check the result (at the end of each phase) for the following array (12,2,16,30,8,28,4,10,20,6,18)