Practice Problem (Recursive Function)

1. Write a recursive implementation of the factorial function.
2. Recall that n! = 1×2×…×n, with the special case that 0! = 1.
3. Write a recursive program to print the nth Fibonacci number.
4. Write a recursive program to calculate the power of xy, where y is a non-negative integer.
5. Write a recursive program to count the number of digits of an integer.
6. Write a recursive program to find the sum of digits of an integer.
7. Write a recursive program to find the sum of the elements of an array of size n.
8. Write a recursive program to find the products of the elements of an array of size n.
9. Write a recursive program to find the maximum of the elements of an array of size n.
10. Write a recursive program to find the minimum of the elements of an array of size n.
11. Write a recursive program to find the average of the elements of an array of size n.
12. Write a recursive program to count/print the odd/even numbers of an array of n
13. integers.
14. Write a recursive program to print an array of size n in reverse order.
15. Write a recursive program to print the even numbers in a given range.

|  |  |
| --- | --- |
| Sample input | Sample output |
| 3 10 | 4 6 8 10 |

1. Write a recursive program to check if a given string is a palindrome or not.
2. Write a recursive program to find the maximum of a 2d array.
3. Write a recursive program to count the prime numbers of a given 2d array.
4. Find the sum of the following series up to Nth position.

(a) 1 + 2 + 3 + …

(b) 12+ 22+ 32+ …

(c) 1 \* 3 + 2 \* 5 + 3 \* 7 + 4 \* 9 + …

(d) 2 \* 3 + 4 \* 5 + 8 \* 7 + 16 \* 9 +...

(e) 2 \* 3 \* 4 + 4 \* 5 \* 3 + 8 \* 7 \* 2 + 16 \* 9 \* 1 +…

1. Write a recursive C++ function to implement the merge sort algorithm for efficiently sorting an array. Ensure that your code handles the merging step correctly.
2. Implement a recursive C++ function for the quick sort algorithm to sort an array. Provide code that includes the partitioning step and explain how it works in each recursive call.
3. Create a C++ program with a recursive function to perform binary search on a sorted array. Your function should find the position of a specific element.
4. Write a C++ program that utilizes recursion to implement depth-first search (DFS) on a graph or tree. Implement a recursive C++ function to search for a specific key in a binary search tree (BST).
5. Develop a C++ program containing a recursive function to find the maximum and minimum elements in an array.