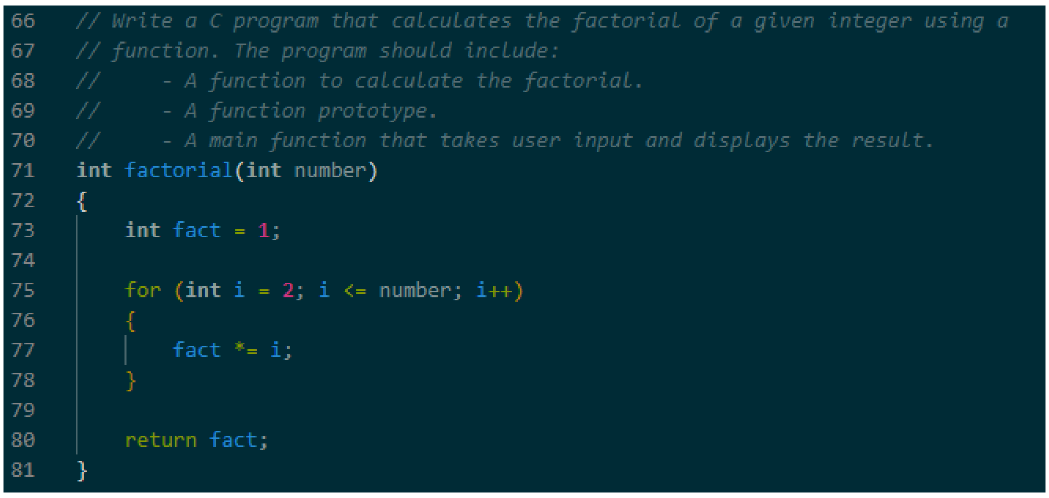
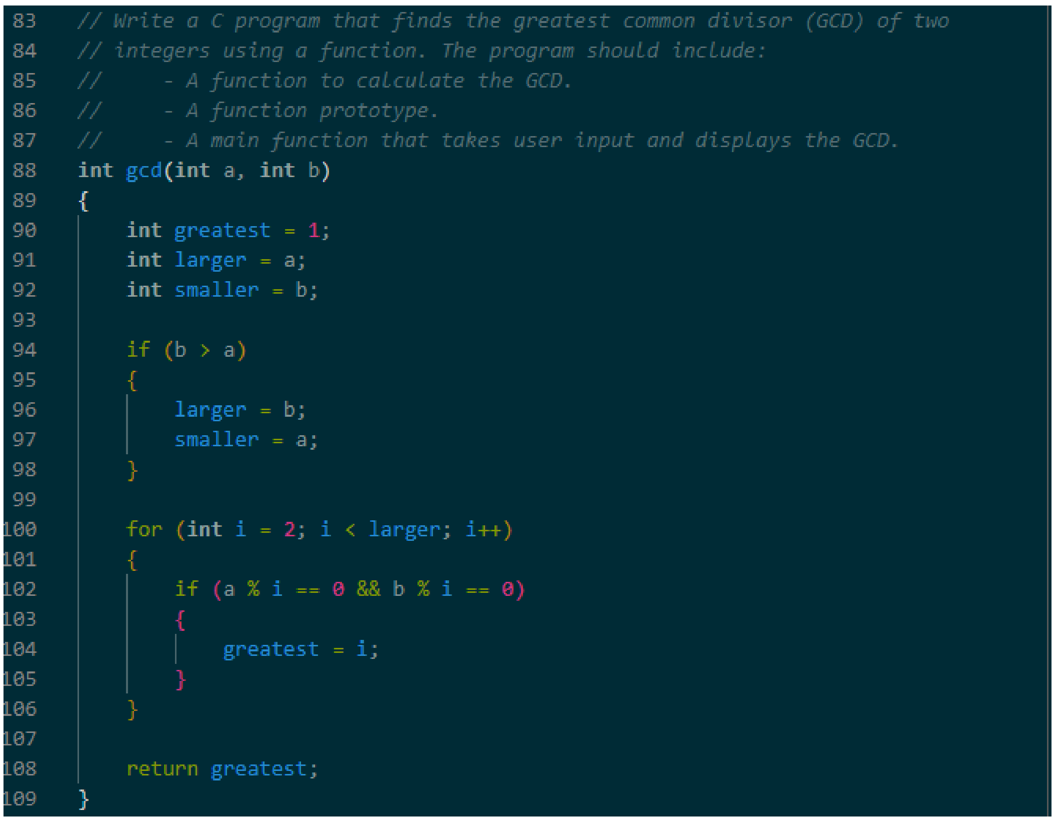
Dr. Chatterjee's CS 3150
Report by Weston Shakespear
Fall 2023

Lab 1

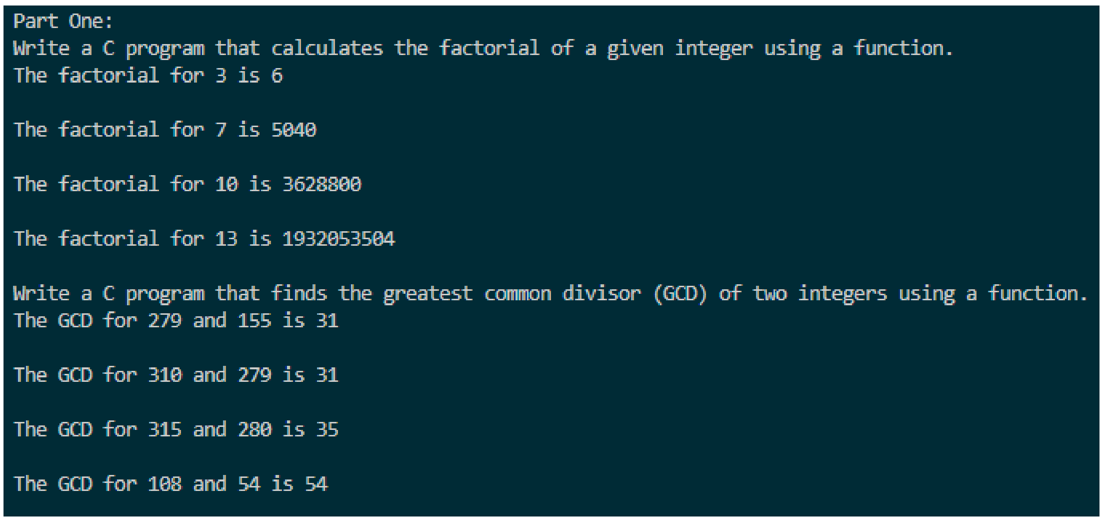
Part 1: Function Basics



```c
// Write a C program that calculates the factorial of a given integer using a
// function. The program should include:
//      - A function to calculate the factorial.
//      - A function prototype.
//      - A main function that takes user input and displays the result.
int factorial(int number)
{
    int fact = 1;

    for (int i = 2; i <= number; i++)
    {
        fact *= i;
    }

    return fact;
}
```

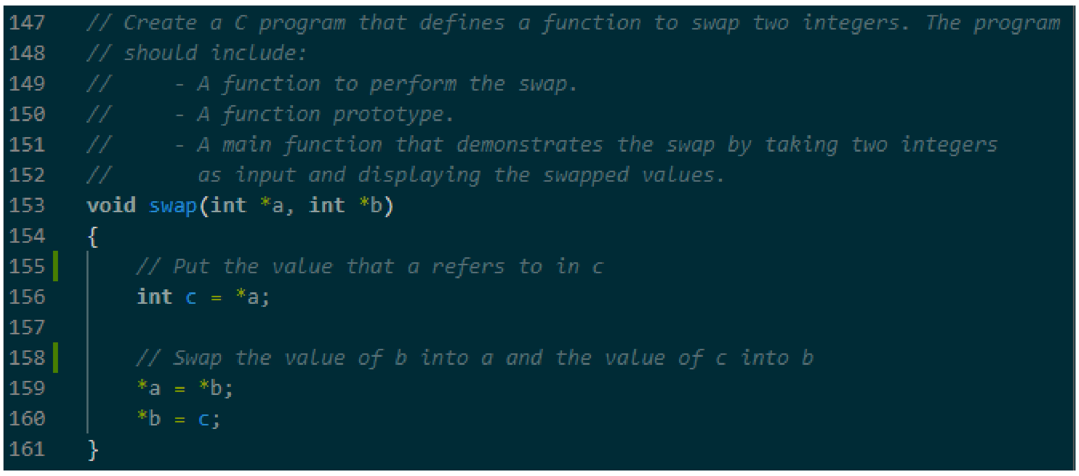


```c
// Write a C program that finds the greatest common divisor (GCD) of two
// integers using a function. The program should include:
//      - A function to calculate the GCD.
//      - A function prototype.
//      - A main function that takes user input and displays the GCD.
int gcd(int a, int b)
{
    int greatest = 1;
    int larger = a;
    int smaller = b;

    if (b > a)
    {
        larger = b;
        smaller = a;
    }

    for (int i = 2; i < larger; i++)
    {
        if (a % i == 0 && b % i == 0)
        {
            greatest = i;
        }
    }

    return greatest;
}
```

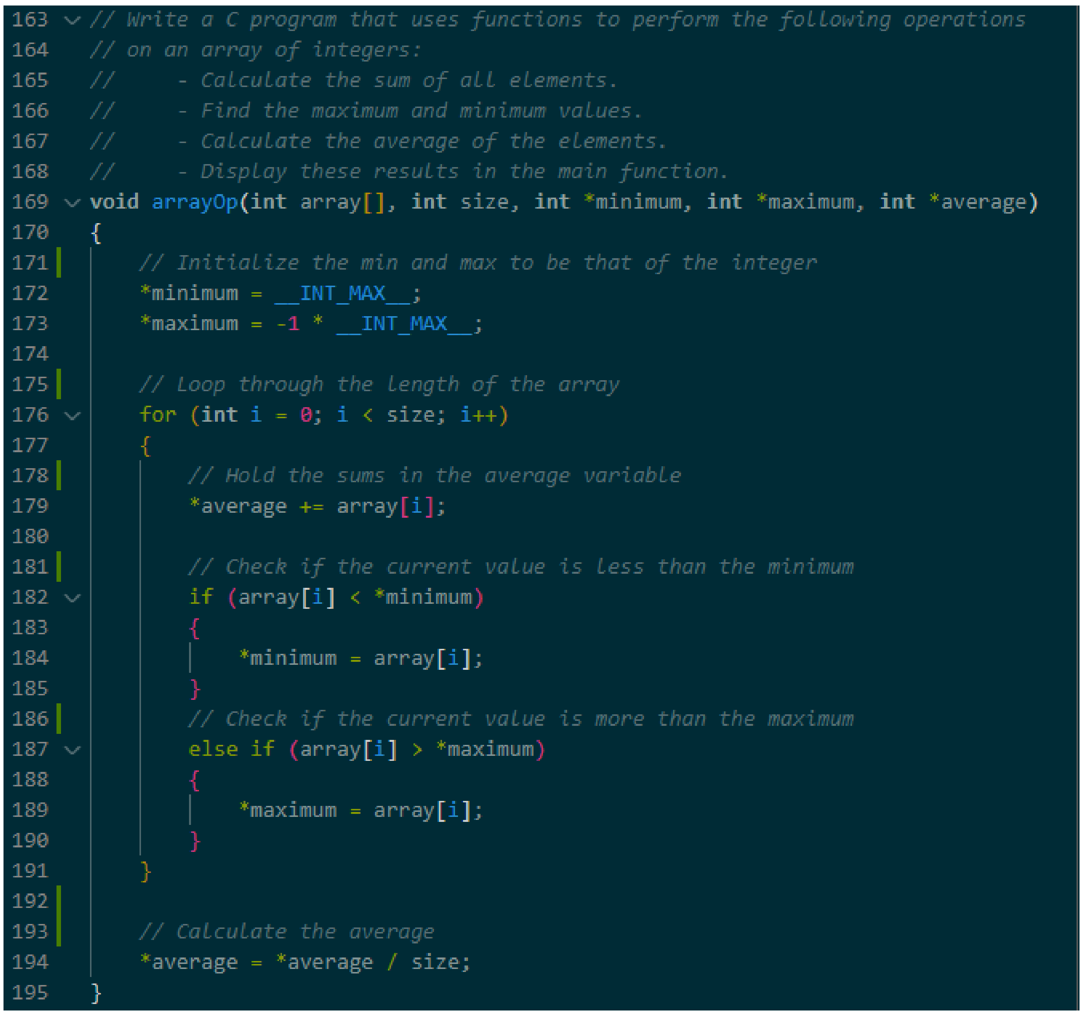


```
Part One:
Write a C program that calculates the factorial of a given integer using a function.
The factorial for 3 is 6

The factorial for 7 is 5040

The factorial for 10 is 3628800

The factorial for 13 is 1932053504

Write a C program that finds the greatest common divisor (GCD) of two integers using a function.
The GCD for 279 and 155 is 31

The GCD for 310 and 279 is 31

The GCD for 315 and 280 is 35

The GCD for 108 and 54 is 54
```

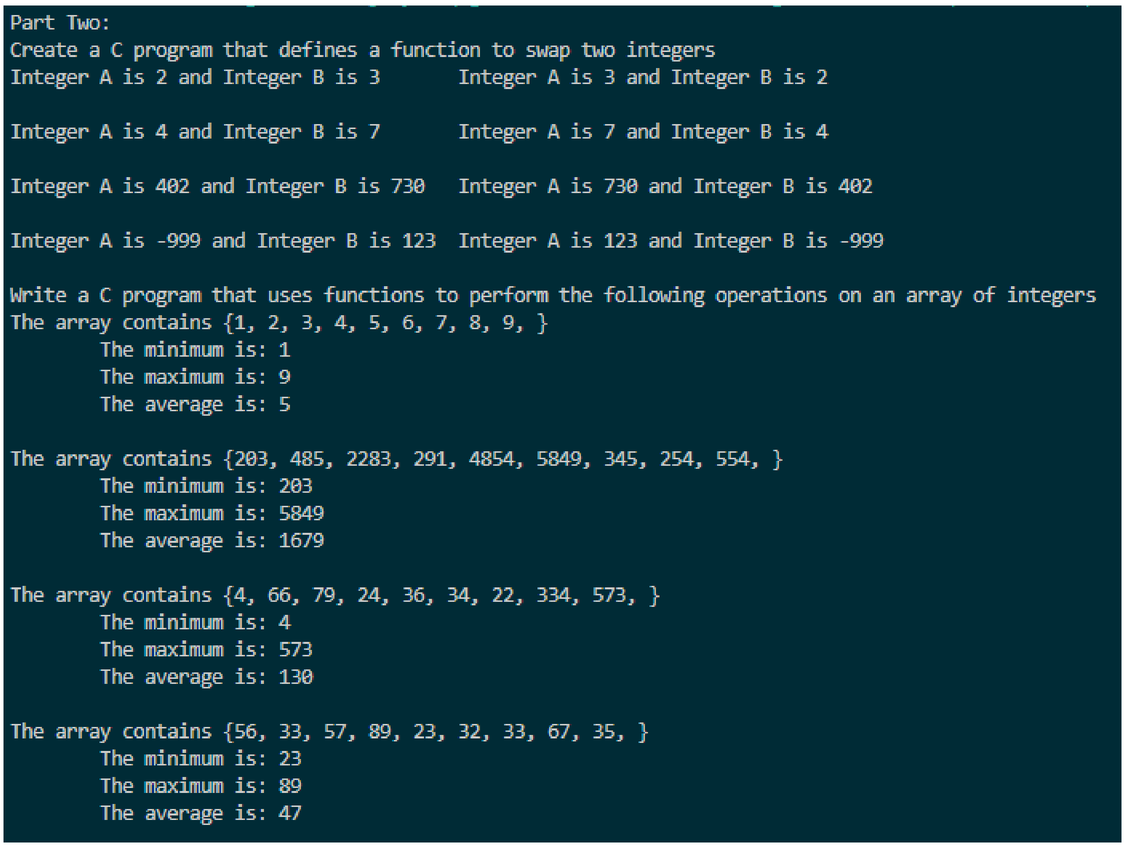
Part 2: Parameter Passing and Return Values



```c
// Create a C program that defines a function to swap two integers. The program
// should include:
//      - A function to perform the swap.
//      - A function prototype.
//      - A main function that demonstrates the swap by taking two integers
//        as input and displaying the swapped values.
void swap(int *a, int *b)
{
    // Put the value that a refers to in c
    int c = *a;

    // Swap the value of b into a and the value of c into b
    *a = *b;
    *b = c;
}
```

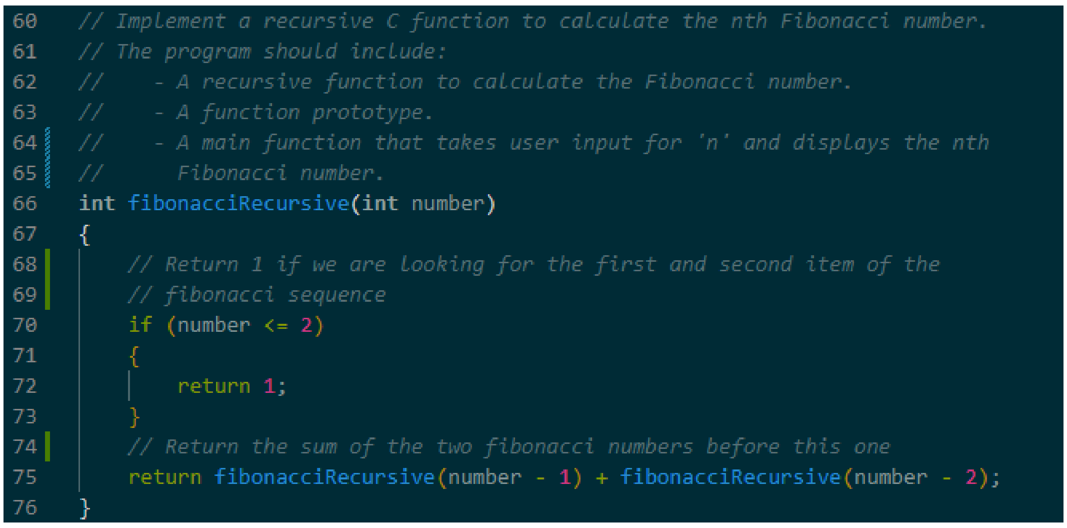


```c
// Write a C program that uses functions to perform the following operations
// on an array of integers:
//      - Calculate the sum of all elements.
//      - Find the maximum and minimum values.
//      - Calculate the average of the elements.
//      - Display these results in the main function.
void arrayOp(int array[], int size, int *minimum, int *maximum, int *average)
{
    // Initialize the min and max to be that of the integer
    *minimum = __INT_MAX__;
    *maximum = -1 * __INT_MAX__;

    // Loop through the length of the array
    for (int i = 0; i < size; i++)
    {
        // Hold the sums in the average variable
        *average += array[i];

        // Check if the current value is less than the minimum
        if (array[i] < *minimum)
        {
            *minimum = array[i];
        }
        // Check if the current value is more than the maximum
        else if (array[i] > *maximum)
        {
            *maximum = array[i];
        }
    }

    // Calculate the average
    *average = *average / size;
}
```

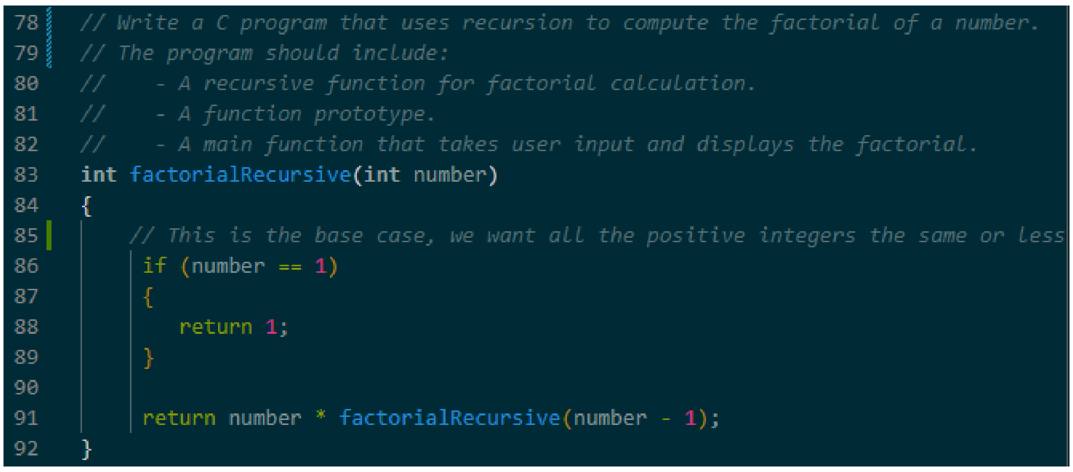


```
Part Two:
Create a C program that defines a function to swap two integers
Integer A is 2 and Integer B is 3       Integer A is 3 and Integer B is 2

Integer A is 4 and Integer B is 7       Integer A is 7 and Integer B is 4

Integer A is 402 and Integer B is 730   Integer A is 730 and Integer B is 402

Integer A is -999 and Integer B is 123  Integer A is 123 and Integer B is -999

Write a C program that uses functions to perform the following operations on an array of integers
The array contains {1, 2, 3, 4, 5, 6, 7, 8, 9, }
        The minimum is: 1
        The maximum is: 9
        The average is: 5

The array contains {203, 485, 2283, 291, 4854, 5849, 345, 254, 554, }
        The minimum is: 203
        The maximum is: 5849
        The average is: 1679

The array contains {4, 66, 79, 24, 36, 34, 22, 334, 573, }
        The minimum is: 4
        The maximum is: 573
        The average is: 130

The array contains {56, 33, 57, 89, 23, 32, 33, 67, 35, }
        The minimum is: 23
        The maximum is: 89
        The average is: 47
```

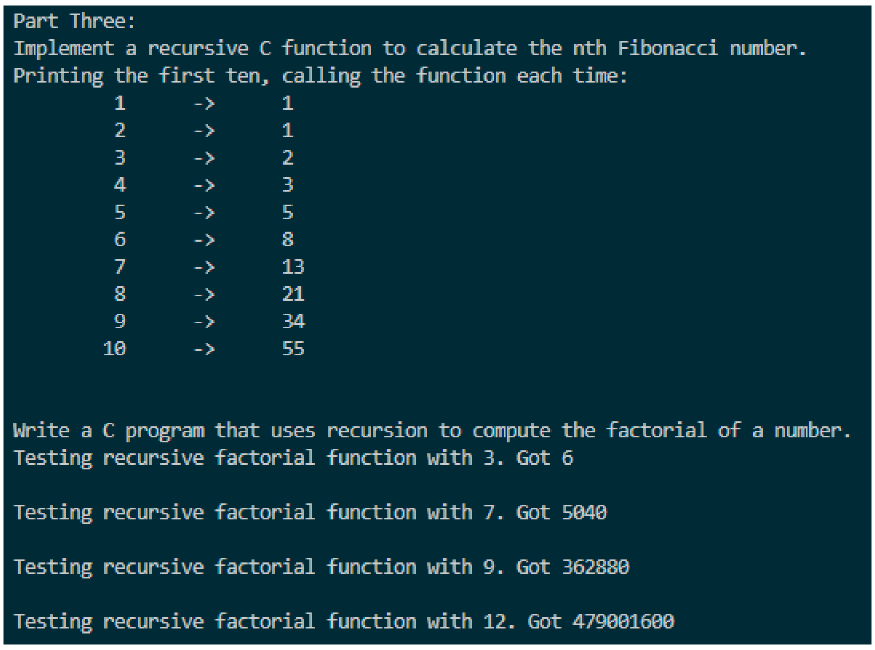
Part 3: Recursion



```c
// Implement a recursive C function to calculate the nth Fibonacci number.
// The program should include:
//     - A recursive function to calculate the Fibonacci number.
//     - A function prototype.
//     - A main function that takes user input for 'n' and displays the nth
//       Fibonacci number.
int fibonacciRecursive(int number)
{
    // Return 1 if we are looking for the first and second item of the
    // fibonacci sequence
    if (number <= 2)
    {
        return 1;
    }
    // Return the sum of the two fibonacci numbers before this one
    return fibonacciRecursive(number - 1) + fibonacciRecursive(number - 2);
}
```



```c
// Write a C program that uses recursion to compute the factorial of a number.
// The program should include:
//     - A recursive function for factorial calculation.
//     - A function prototype.
//     - A main function that takes user input and displays the factorial.
int factorialRecursive(int number)
{
    // This is the base case, we want all the positive integers the same or less
    if (number == 1)
    {
        return 1;
    }

    return number * factorialRecursive(number - 1);
}
```



```
Part Three:
Implement a recursive C function to calculate the nth Fibonacci number.
Printing the first ten, calling the function each time:
         1      ->      1
         2      ->      1
         3      ->      2
         4      ->      3
         5      ->      5
         6      ->      8
         7      ->      13
         8      ->      21
         9      ->      34
        10      ->      55


Write a C program that uses recursion to compute the factorial of a number.
Testing recursive factorial function with 3. Got 6

Testing recursive factorial function with 7. Got 5040

Testing recursive factorial function with 9. Got 362880

Testing recursive factorial function with 12. Got 479001600
```