**Day 6**

1. Write a function to calculate the factorial of a number.

int factorial(int n) {

if(n == 0) return 1;

return n \* factorial(n - 1);

}

2. Write a function to check whether a number is prime or not.

int isPrime(int n) {

if(n <= 1) return 0;

for(int i = 2; i <= n/2; i++) {

if(n % i == 0) return 0;

}

return 1;

}

3. Write a function to calculate power using recursion.

int power(int base, int exp) {

if(exp == 0) return 1;

return base \* power(base, exp - 1);

}

4. Write a function to check whether a number is a palindrome using recursion.

int reverseNum(int n, int rev) {

if(n == 0) return rev;

return reverseNum(n / 10, rev \* 10 + n % 10);

}

int isPalindrome(int n) {

return n == reverseNum(n, 0);

}

5. Write a program to calculate nCr using a function.

int factorial(int n) {

if(n == 0) return 1;

return n \* factorial(n - 1);

}

int nCr(int n, int r) {

return factorial(n) / (factorial(r) \* factorial(n - r));

}

6. Write a program to demonstrate call by value and call by reference.

void byValue(int a) {

a = 100;

}

void byReference(int \*a) {

\*a = 100;

}

7. Write a function to swap two numbers.

void swap(int \*a, int \*b) {

int temp = \*a;

\*a = \*b;

\*b = temp;

}

8. Write a function to generate Fibonacci series using recursion.

int fibonacci(int n) {

if(n == 0) return 0;

if(n == 1) return 1;

return fibonacci(n - 1) + fibonacci(n - 2);

}

9. Write a function to find GCD and LCM of two numbers.

int gcd(int a, int b) {

if(b == 0) return a;

return gcd(b, a % b);

}

int lcm(int a, int b) {

return (a \* b) / gcd(a, b);

}

10. Write a program to demonstrate global and local variables.

int global = 10;

void display() {

int local = 20;

printf("Global: %d, Local: %d\n", global, local);

}