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

#include <stdio.h>

int factorial(int n) {

if (n == 0) return 1;

return n \* factorial(n - 1);

}

int main() {

int num = 5;

printf("Factorial: %d\n", factorial(num));

}

1. Write a function to check whether a number is prime.

#include <stdio.h>

int isPrime(int n) {

if (n <= 1) return 0;

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

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

}

return 1;

}

int main() {

int num = 7;

if (isPrime(num))

printf("Prime number\n");

else

printf("Not a prime number\n");

return 0;

}

1. Write a function to calculate power using recursion.

#include <stdio.h>

int power(int base, int exp) {

if (exp == 0) return 1;

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

}

int main() {

printf("Power: %d\n", power(2, 3));

return 0;

}

1. Write a function to check palindrome number using recursion.

#include <stdio.h>

int reverse(int n, int rev) {

if (n == 0) return rev;

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

}

int main() {

int num = 121;

if (num == reverse(num, 0))

printf("Palindrome\n");

else

printf("Not Palindrome\n");

}

1. Write a function to calculate nCr (combinations).

#include <stdio.h>

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));

}

int main() {

printf("nCr: %d\n", nCr(5, 2));

}

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

#include <stdio.h>

void callByValue(int x) {

x = x + 5;

printf("Inside callByValue: %d\n", x);

}

void callByReference(int \*x) {

\*x = \*x + 5;

printf("Inside callByReference: %d\n", \*x);

}

int main() {

int a = 10;

callByValue(a);

printf("After callByValue: %d\n", a);

callByReference(&a);

printf("After callByReference: %d\n", a);

}

1. Write a program using function to swap two numbers.

#include <stdio.h>

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

int t = \*a;

\*a = \*b;

\*b = t;

}

int main() {

int x = 3, y = 4;

printf("Before: x = %d, y = %d\n", x, y);

swap(&x, &y);

printf("After: x = %d, y = %d\n", x, y);

}

1. Write a recursive function to find the nth Fibonacci number.

#include <stdio.h>

int fibonacci(int n) {

if (n == 0) return 0;

if (n == 1) return 1;

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

}

int main() {

int n = 5;

printf("Fibonacci: %d\n", fibonacci(n));

}

1. Write a program to find GCD and LCM using functions

#include <stdio.h>

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);

}

int main() {

int x = 12, y = 18;

printf("GCD: %d\n", gcd(x, y));

printf("LCM: %d\n", lcm(x, y));

}

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

#include <stdio.h>

int global = 100;

void show() {

int local = 50;

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

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

}

int main() {

show();

printf("Access Global in main: %d\n", global);

}