**Q1:** Write a program in C to print first n numbers using recursion.

#include <stdio.h>

void printNumbers(int n){

if (n > 0)

{

printNumbers(n - 1);

printf("%d ", n);

}

}

int main()

{

int n;

printf("Enter the value of n: ");

scanf("%d", &n);

printf("First %d numbers: ", n);

printNumbers(n);

printf("\n");

return 0;

}

**Q2:** Write a program in C to calculate the sum of numbers from 1 to n using recursion.

#include<stdio.h>

int calsum(int n){

if(n==0)

return 1;

else

return n + calsum(n-1);

}

int main(){

int n,sum;

printf("Enter a Number: ");

scanf("%d",&n);

sum=calsum(n);

printf("Sum of numbers from 1 to %d is %d.\n",n,sum);

return 0;

}

**Q3:** Write a program to find the factorial of a number.

#include<stdio.h>

long long factorial(int n){

if(n==0 || n==1)

return 1;

else

return n \* factorial(n-1);

}

int main(){

int n;

printf("Enter a Number: ");

scanf("%d",&n);

printf("Factorials of numbers from 1 to %d:\n", n);

for (int i = 1; i <= n; i++)

{

long long fact = factorial(i);

printf("%d! = %llu\n", i, fact);

}

return 0;

}

**Q4:** Write a program in C to Print Fibonacci Series using recursion.

#include <stdio.h>

int fibonacci(int n)

{

if (n <= 1)

return n;

else

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

}

void printFibonacciSeries(int n)

{

printf("Fibonacci Series up to %d terms: ", n);

for (int i = 0; i < n; i++)

{

printf("%d ", fibonacci(i));

}

printf("\n");

}

int main()

{

int n;

printf("Enter the number of terms: ");

scanf("%d", &n);

printFibonacciSeries(n);

return 0;

}

**Q5:** Write a program in C to sum the digits of a given number using recursion.

#include <stdio.h>

int sumDigits(int number)

{

if (number == 0)

return 0;

else

return (number % 10) + sumDigits(number / 10);

}

int main()

{

int number;

printf("Enter a number: ");

scanf("%d", &number);

int sum = sumDigits(number);

printf("Sum of the digits: %d\n", sum);

return 0;

}

**Q6:** Write a program in C to find GCD of two numbers using recursion

#include <stdio.h>

int gcd(int num1, int num2)

{

if (num2 == 0)

return num1;

else

return gcd(num2, num1 % num2);

}

int main()

{

int num1, num2;

printf("Enter two numbers: ");

scanf("%d %d", &num1, &num2);

int result = gcd(num1, num2);

printf("GCD of %d and %d is %d\n", num1, num2, result);

return 0;

}

**Q7:** Write a program to convert decimal to binary using recursion

#include <stdio.h>

void decimalToBinary(int decimal)

{

if (decimal > 0)

{

decimalToBinary(decimal / 2);

printf("%d", decimal % 2);

}

}

int main()

{

int decimal;

printf("Enter a decimal number: ");

scanf("%d", &decimal);

printf("Binary representation: ");

decimalToBinary(decimal);

printf("\n");

return 0;

}

**Q8:** Write a program to reverse a number using recursion

#include <stdio.h>

int reverseNumber(int number, int reversed)

{

if (number == 0)

return reversed;

else

return reverseNumber(number / 10, reversed \* 10 + number % 10);

}

int main()

{

int number;

printf("Enter a number: ");

scanf("%d", &number);

int reversed = reverseNumber(number, 0);

printf("Reversed number: %d\n", reversed);

return 0;

}

**Q9:** Write a program in C to check a number is a prime number or not using recursion

#include <stdio.h>

int isPrime(int number, int divisor)

{

if (number <= 2)

{

if (number == 2)

return 1;

else

return 0;

}

if (divisor == 1)

return 1;

if (number % divisor == 0)

return 0;

return isPrime(number, divisor - 1);

}

int main()

{

int number;

printf("Enter a number: ");

scanf("%d", &number);

if (isPrime(number, number - 1))

printf("%d is a prime number.\n", number);

else

printf("%d is not a prime number.\n", number);

return 0;

}

**Q10:** Write a program in C to find the LCM of two numbers using recursion.

#include <stdio.h>

int gcd(int num1, int num2)

{

if (num2 == 0)

return num1;

else

return gcd(num2, num1 % num2);

}

int lcm(int num1, int num2)

{

int gcdValue = gcd(num1, num2);

return (num1 \* num2) / gcdValue;

}

int main()

{

int num1, num2;

printf("Enter two numbers: ");

scanf("%d %d", &num1, &num2);

int lcmValue = lcm(num1, num2);

printf("LCM of %d and %d is %d\n", num1, num2, lcmValue);

return 0;

}

**Q11:** Write a program in C to calculate the power of any number using recursion.

#include <stdio.h>

double power(double base, int exponent)

{

if (exponent == 0)

return 1;

else if (exponent > 0)

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

else

return (1 / base) \* power(base, exponent + 1);

}

int main()

{

double base;

int exponent;

printf("Enter the base number: ");

scanf("%lf", &base);

printf("Enter the exponent: ");

scanf("%d", &exponent);

double result = power(base, exponent);

printf("Result: %.2lf\n", result);

return 0;

}