**COMSATS INSTITUTE OF INFORMATION TECHNOLOGY,**

**ABBOTTABAD CAMPUS.**

**ASSIGNMENT NO 6.**

**SUBJECT:**

**PROGRAMMING FUNDAMENTALS.**

**SUBMITTED TO:**

**DR KASHIF BILAL.**

**SUBMITTED BY:**

**AEMAN SAJID**

**REGISTRATION NUMBER:**

**FA22-BSE-084**

**PROGRAM AND SECTION:**

**SOFTWARE ENGINEERING- SEC: 2B.**

1. Write a program in C to print first n numbers using recursion.

#include <stdio.h>

void printNumbers(int n)

{

    if (n <= 0)

        return;

    printNumbers(n - 1); // Recursive call to print n-1 numbers first

    printf("%d ", n);    // Print the current number

}

int main()

{

    int n;

    printf("Enter the value of n: ");

    scanf("%d", &n);

    printf("The first %d numbers are: ", n);

    printNumbers(n);

    return 0;

}

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

#include <stdio.h>

int calculateSum(int n) {

    if (n == 1) {

        return 1;

    } else {

        return n + calculateSum(n - 1);

    }

}

int main() {

    int n;

    printf("Enter a positive integer: ");

    scanf("%d", &n);

    if (n < 1) {

        printf("Invalid input. Please enter a positive integer.\n");

        return 0;

    }

    int sum = calculateSum(n);

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

    return 0;

}

3. Write a program to find the factorial of a number. #include <stdio.h>

Int factorial(int n) {

    if (n == 0 || n == 1) {

        return 1;

    } else {

        return n \* factorial(n - 1);

    }

}

int main() {

    int n;

    printf("Enter a non-negative integer: ");

    scanf("%d", &n);

    if (n < 0) {

        printf("Invalid input. Please enter a non-negative integer.\n");

        return 0;

    }

    int fact = factorial(n);

    printf("The factorial of %d is %llu.\n", n, fact);

    return 0;

}

4. 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 terms) {

    for (int i = 0; i < terms; i++) {

        printf("%d ", fibonacci(i));

    }

    printf("\n");

}

int main() {

    int terms;

    printf("Enter the number of terms in the Fibonacci series: ");

    scanf("%d", &terms);

    if (terms <= 0) {

        printf("Invalid input. Please enter a positive integer.\n");

        return 0;

    }

    printf("Fibonacci series: ");

    printFibonacciSeries(terms);

    return 0;

}

5. Write a program in C to sum the digits of a given number using recursion.

#include <stdio.h>

int sumOfDigits(int n) {

    if (n == 0) {

        return 0;

    } else {

        return (n % 10) + sumOfDigits(n / 10);

    }

}

int main() {

    int number;

    printf("Enter a positive integer: ");

    scanf("%d", &number);

    if (number < 0) {

        printf("Invalid input. Please enter a positive integer.\n");

        return 0;

    }

    int sum = sumOfDigits(number);

    printf("The sum of digits of %d is %d.\n", number, sum);

    return 0;

}

6. Write a program in C to find GCD of two numbers using recursion

 #include <stdio.h>

int gcd(int a, int b) {

    if (b == 0) {

        return a;

    } else {

        return gcd(b, a % b);

    }

}

int main() {

    int num1, num2;

    printf("Enter two positive integers: ");

    scanf("%d %d", &num1, &num2);

    if (num1 <= 0 || num2 <= 0) {

        printf("Invalid input. Please enter positive integers.\n");

        return 0;

    }

    int result = gcd(num1, num2);

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

    return 0;

}

7. 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);

    if (decimal < 0) {

        printf("Invalid input. Please enter a non-negative integer.\n");

        return 0;

    }

    printf("Binary representation: ");

    if (decimal == 0) {

        printf("0");

    } else {

        decimalToBinary(decimal);

    }

    printf("\n");

    return 0;

}

8. Write a program to reverse a number using recursion

#include <stdio.h>

int reverseNumber(int num, int reversedNum) {

    if (num == 0) {

        return reversedNum;

    } else {

        reversedNum = (reversedNum \* 10) + (num % 10);

        return reverseNumber(num / 10, reversedNum);

    }

}

int main() {

    int number;

    printf("Enter a positive integer: ");

    scanf("%d", &number);

    if (number < 0) {

        printf("Invalid input. Please enter a positive integer.\n");

        return 0;

    }

    int reversed = reverseNumber(number, 0);

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

    return 0;

}

9. Write a program in C to check a number is a prime number or not using recursion.

 #include <stdio.h>

int isPrimeRecursive(int num, int divisor) {

    if (divisor == 1) {

        return 1;  // Base case: num is prime

    } else {

        if (num % divisor == 0) {

            return 0;  // Base case: num is not prime

        } else {

            return isPrimeRecursive(num, divisor - 1);  // Recursive call

        }

    }

}

int isPrime(int num) {

    if (num <= 1) {

        return 0;  // Numbers less than or equal to 1 are not prime

    } else {

        return isPrimeRecursive(num, num - 1);

    }

}

int main() {

    int number;

    printf("Enter a positive integer: ");

    scanf("%d", &number);

    if (number < 0) {

        printf("Invalid input. Please enter a positive integer.\n");

        return 0;

    }

    int result = isPrime(number);

    if (result == 1) {

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

    } else {

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

    }

    return 0;

}

10. Write a program in C to find the LCM of two numbers using recursion.

#include <stdio.h>

int gcd(int a, int b) {

    if (b == 0) {

        return a;

    } else {

        return gcd(b, a % b);

    }

}

int lcm(int a, int b) {

    return (a \* b) / gcd(a, b);

}

int main() {

    int num1, num2;

    printf("Enter two positive integers: ");

    scanf("%d %d", &num1, &num2);

    if (num1 <= 0 || num2 <= 0) {

        printf("Invalid input. Please enter positive integers.\n");

        return 0;

    }

    int result = lcm(num1, num2);

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

    return 0;

}

11. 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 1.0;

    } else if (exponent > 0) {

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

    } else {

        return (1.0 / 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("The result is: %.2lf\n", result);

    return 0;

}