/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* DAY 12: NUMBER-BASED PROGRAMS – PART 2 \*

\* Contains 10 C programs for advanced number operations \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <math.h> // Used for power calculation

// ======================

// 1. Find LCM of Two Numbers

// ======================

void findLCM() {

int num1, num2, max;

printf("Enter two numbers: ");

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

max = (num1 > num2) ? num1 : num2;

while (1) {

if (max % num1 == 0 && max % num2 == 0) {

printf("LCM of %d and %d: %d\n", num1, num2, max);

break;

}

++max;

}

}

// ======================

// 2. Find GCD of Two Numbers

// ======================

void findGCD() {

int num1, num2, gcd, i;

printf("Enter two numbers: ");

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

for (i = 1; i <= num1 && i <= num2; ++i) {

if (num1 % i == 0 && num2 % i == 0)

gcd = i;

}

printf("GCD of %d and %d: %d\n", num1, num2, gcd);

}

// ======================

// 3. Decimal to Binary

// ======================

void decimalToBinary() {

int decimal, binary[32], i = 0;

printf("Enter a decimal number: ");

scanf("%d", &decimal);

while (decimal > 0) {

binary[i] = decimal % 2;

decimal /= 2;

++i;

}

printf("Binary: ");

for (int j = i - 1; j >= 0; --j)

printf("%d", binary[j]);

printf("\n");

}

// ======================

// 4. Binary to Decimal

// ======================

void binaryToDecimal() {

long binary, decimal = 0, base = 1;

printf("Enter a binary number: ");

scanf("%ld", &binary);

while (binary > 0) {

decimal += (binary % 10) \* base;

binary /= 10;

base \*= 2;

}

printf("Decimal: %ld\n", decimal);

}

// ======================

// 5. Check Harshad Number

// ======================

void checkHarshad() {

int num, original, sum = 0;

printf("Enter a number: ");

scanf("%d", &num);

original = num;

while (num > 0) {

sum += num % 10;

num /= 10;

}

printf("%d is %s\n", original, (original % sum == 0) ? "Harshad" : "Not Harshad");

}

// ======================

// 6. Count Digits in a Number

// ======================

void countDigits() {

long num;

int count = 0;

printf("Enter a number: ");

scanf("%ld", &num);

while (num != 0) {

num /= 10;

++count;

}

printf("Number of digits: %d\n", count);

}

// ======================

// 7. Power of a Number (Using Loops)

// ======================

void calculatePower() {

int base, exponent;

long result = 1;

printf("Enter base and exponent: ");

scanf("%d %d", &base, &exponent);

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

result \*= base;

printf("%d^%d = %ld\n", base, exponent, result);

}

// ======================

// 8. Fibonacci Series up to N Terms

// ======================

void printFibonacci() {

int n, t1 = 0, t2 = 1, nextTerm;

printf("Enter number of terms: ");

scanf("%d", &n);

printf("Fibonacci Series: ");

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

printf("%d ", t1);

nextTerm = t1 + t2;

t1 = t2;

t2 = nextTerm;

}

printf("\n");

}

// ======================

// 9. Sum of First N Natural Numbers

// ======================

void sumOfNaturalNumbers() {

int n, sum = 0;

printf("Enter N: ");

scanf("%d", &n);

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

sum += i;

printf("Sum of first %d natural numbers: %d\n", n, sum);

}

// ======================

// 10. Print All Factors of a Number

// ======================

void printFactors() {

int num;

printf("Enter a number: ");

scanf("%d", &num);

printf("Factors of %d: ", num);

for (int i = 1; i <= num; ++i) {

if (num % i == 0)

printf("%d ", i);

}

printf("\n");

}