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

\* DAY 11: NUMBER-BASED PROGRAMS – PART 1 \*

\* Contains 10 C programs for basic number operations. \*

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

#include <stdio.h>

#include <math.h> // Used for Armstrong number calculation

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

// 1. Check Even or Odd

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

void checkEvenOdd() {

int num;

printf("Enter a number: ");

scanf("%d", &num);

printf("%d is %s\n", num, (num % 2 == 0) ? "Even" : "Odd");

}

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

// 2. Check Prime Number

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

void checkPrime() {

int num, i, flag = 0;

printf("Enter a number: ");

scanf("%d", &num);

for (i = 2; i <= num / 2; ++i) {

if (num % i == 0) {

flag = 1;

break;

}

}

printf("%d is %s\n", num, (flag == 0 && num > 1) ? "Prime" : "Not Prime");

}

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

// 3. Print Prime Numbers in Range

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

void printPrimesInRange() {

int low, high, i, flag;

printf("Enter range (low high): ");

scanf("%d %d", &low, &high);

printf("Primes between %d and %d:\n", low, high);

while (low <= high) {

flag = 0;

for (i = 2; i <= low / 2; ++i) {

if (low % i == 0) {

flag = 1;

break;

}

}

if (flag == 0 && low > 1) printf("%d ", low);

++low;

}

printf("\n");

}

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

// 4. Check Palindrome

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

void checkPalindrome() {

int num, reversed = 0, original;

printf("Enter a number: ");

scanf("%d", &num);

original = num;

while (num != 0) {

reversed = reversed \* 10 + num % 10;

num /= 10;

}

printf("%d is %s\n", original, (original == reversed) ? "Palindrome" : "Not Palindrome");

}

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

// 5. Check Armstrong Number

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

void checkArmstrong() {

int num, original, remainder, n = 0, result = 0;

printf("Enter a number: ");

scanf("%d", &num);

original = num;

while (original != 0) { original /= 10; ++n; }

original = num;

while (original != 0) {

remainder = original % 10;

result += pow(remainder, n);

original /= 10;

}

printf("%d is %s\n", num, (result == num) ? "Armstrong" : "Not Armstrong");

}

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

// 6. Check Strong Number

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

int factorial(int n) {

return (n == 0 || n == 1) ? 1 : n \* factorial(n - 1);

}

void checkStrong() {

int num, original, sum = 0;

printf("Enter a number: ");

scanf("%d", &num);

original = num;

while (num > 0) {

sum += factorial(num % 10);

num /= 10;

}

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

}

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

// 7. Check Perfect Number

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

void checkPerfect() {

int num, i, sum = 0;

printf("Enter a number: ");

scanf("%d", &num);

for (i = 1; i <= num / 2; ++i) {

if (num % i == 0) sum += i;

}

printf("%d is %s\n", num, (sum == num) ? "Perfect" : "Not Perfect");

}

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

// 8. Reverse a Number

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

void reverseNumber() {

int num, reversed = 0;

printf("Enter a number: ");

scanf("%d", &num);

while (num != 0) {

reversed = reversed \* 10 + num % 10;

num /= 10;

}

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

}

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

// 9. Sum of Digits

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

void sumOfDigits() {

int num, sum = 0;

printf("Enter a number: ");

scanf("%d", &num);

while (num != 0) {

sum += num % 10;

num /= 10;

}

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

}

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

// 10. Factorial of a Number

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

void calculateFactorial() {

int num, i, fact = 1;

printf("Enter a number: ");

scanf("%d", &num);

for (i = 1; i <= num; ++i) fact \*= i;

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

}