# Day 12: Number-Based Programs

1. **Write a program to find the LCM and HCF of two numbers.**

#include <stdio.h>

int main() {

int num1, num2, hcf, lcm, i;

printf("Enter two positive integers: "); scanf("%d %d", &num1, &num2);

// Finding HCF

for (i = 1; i <= num1 && i <= num2; ++i) { if (num1 % i == 0 && num2 % i == 0) {

hcf = i;

}

}

// Finding LCM

lcm = (num1 \* num2) / hcf;

printf("HCF of %d and %d = %d\n", num1, num2, hcf); printf("LCM of %d and %d = %d\n", num1, num2, lcm);

return 0;

}

1. **Write a program to find the GCD of two numbers.** (Note: GCD is the same as HCF)

#include <stdio.h>

int main() {

int num1, num2, gcd, i;

printf("Enter two positive integers: "); scanf("%d %d", &num1, &num2);

// Finding GCD

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);

return 0;

}

# Write a program to convert binary to decimal.

#include <stdio.h> #include <math.h>

int main() {

long long binary;

int decimal = 0, i = 0, rem;

printf("Enter a binary number: "); scanf("%lld", &binary);

while (binary != 0) { rem = binary % 10; binary /= 10;

decimal += rem \* pow(2, i);

++i;

}

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

return 0;

}

# Write a program to convert decimal to binary.

#include <stdio.h>

int main() {

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

printf("Enter a decimal number: "); scanf("%d", &decimal);

if (decimal == 0) {

printf("Binary equivalent: 0\n"); return 0;

}

while (decimal > 0) { binary[i] = decimal % 2; decimal /= 2;

i++;

}

printf("Binary equivalent: "); for (j = i - 1; j >= 0; j--) {

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

}

printf("\n");

return 0;

}

# Write a program to find the sum of the digits of a number.

#include <stdio.h>

int main() {

int num, sum = 0, rem;

printf("Enter a number: "); scanf("%d", &num);

while (num != 0) { rem = num % 10; sum += rem; num /= 10;

}

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

return 0;

}

# Write a program to count the number of digits in a number.

#include <stdio.h>

int main() {

long long num; int count = 0;

printf("Enter a number: "); scanf("%lld", &num);

if (num == 0) { count = 1;

} else {

while (num != 0) { num /= 10;

++count;

}

}

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

return 0;

}

# Write a program to check if a number is an Armstrong number.

#include <stdio.h> #include <math.h>

int main() {

int num, originalNum, rem, n = 0, result = 0;

printf("Enter an integer: "); scanf("%d", &num);

originalNum = num;

// Count the number of digits while (originalNum != 0) {

originalNum /= 10;

++n;

}

originalNum = num; // Reset originalNum

// Calculate sum of powers of digits while (originalNum != 0) {

rem = originalNum % 10; result += pow(rem, n); originalNum /= 10;

}

if (result == num)

printf("%d is an Armstrong number.\n", num); else

printf("%d is not an Armstrong number.\n", num);

return 0;

}

# Write a program to print the Fibonacci series up to n terms.

#include <stdio.h>

int main() {

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

printf("Enter the 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");

return 0;

}

# Write a program to print all prime numbers between two numbers.

#include <stdio.h>

int main() {

int low, high, i, flag;

printf("Enter two numbers (intervals): "); scanf("%d %d", &low, &high);

printf("Prime numbers between %d and %d are: ", low, high); while (low < high) {

flag = 0;

if (low <= 1) { // 0 and 1 are not prime numbers

++low; continue;

}

for (i = 2; i <= low / 2; ++i) { if (low % i == 0) {

flag = 1; break;

}

}

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

++low;

}

printf("\n");

return 0;

}

# Write a program to print all factors of a given number.

#include <stdio.h>

int main() {

int num, i;

printf("Enter a positive integer: "); scanf("%d", &num);

printf("Factors of %d are: ", num); for (i = 1; i <= num; ++i) {

if (num % i == 0) { printf("%d ", i);

}

}

printf("\n");

return 0;

}