**Bài 1: Hiển thị N số nguyên dương đầu tiên**

#include <stdio.h>

int main() {

int n;

printf("How many numbers that you want to display? ");

scanf("%d", &n);

if (n <= 0) {

printf("Accept positive number only!\n");

} else {

printf("The first %d positive integers are:\n", n);

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

printf("%d", i);

if (i < n) printf(", ");

}

printf("\n");

}

return 0;

}

**Bài 2: Hiển thị tất cả các số nguyên dương có 2 chữ số**

#include <stdio.h>

int main() {

printf("All 2-digit positive integers are:\n");

for (int i = 10; i <= 99; i++) {

printf("%d", i);

if (i < 99) printf(", ");

}

printf("\n");

return 0;

}

**Bài 3: Hiển thị tất cả các số nguyên âm có 3 chữ số**

#include <stdio.h>

int main() {

printf("All 3-digit negative integers are:\n");

for (int i = -999; i <= -100; i++) {

printf("%d", i);

if (i < -100) printf(", ");

}

printf("\n");

return 0;

}

**Bài 4: Hiển thị N số chẵn dương đầu tiên**

#include <stdio.h>

int main() {

int n;

printf("How many positive even numbers that you want to display? ");

scanf("%d", &n);

if (n <= 0) {

printf("Accept positive number only!\n");

} else {

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

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

printf("%d", 2 \* i);

if (i < n) printf(", ");

}

printf("\n");

}

return 0;

}

**Bài 5: Hiển thị N số lẻ dương đầu tiên**

#include <stdio.h>

int main() {

int n;

printf("How many positive odd numbers that you want to display? ");

scanf("%d", &n);

if (n <= 0) {

printf("Accept positive number only!\n");

} else {

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

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

printf("%d", 2 \* i - 1);

if (i < n) printf(", ");

}

printf("\n");

}

return 0;

}

Dưới đây là mã C cho 10 bài tiếp theo:

**Bài 6: Hiển thị tất cả các số Armstrong**

#include <stdio.h>

int main() {

printf("All Armstrong numbers are:\n");

for (int i = 100; i <= 999; i++) {

int a = i / 100; // Hàng trăm

int b = (i / 10) % 10; // Hàng chục

int c = i % 10; // Hàng đơn vị

if (a \* a \* a + b \* b \* b + c \* c \* c == i) {

printf("%d ", i);

}

}

printf("\n");

return 0;

}

**Bài 7: Tính tổng S=1+2+3+...+NS = 1 + 2 + 3 + ... + NS=1+2+3+...+N**

#include <stdio.h>

int main() {

int n, sum = 0;

printf("Please enter positive integer N: ");

scanf("%d", &n);

if (n <= 0) {

printf("Accept positive number only!\n");

} else {

printf("The sum is S = ");

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

sum += i;

printf("%d", i);

if (i < n) printf(" + ");

}

printf(" = %d\n", sum);

}

return 0;

}

**Bài 8: Tính tổng S=1−2+3−4+...+(−1)N+1×NS = 1 - 2 + 3 - 4 + ... + (-1)^{N+1} \times NS=1−2+3−4+...+(−1)N+1×N**

#include <stdio.h>

int main() {

int n, sum = 0;

printf("Please enter positive integer N: ");

scanf("%d", &n);

if (n <= 0) {

printf("Accept positive number only!\n");

} else {

printf("The sum is S = ");

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

if (i % 2 == 0) sum -= i;

else sum += i;

printf("%d", (i % 2 == 0) ? -i : i);

if (i < n) printf(" + ");

}

printf(" = %d\n", sum);

}

return 0;

}

**Bài 9: Tính tổng S=1+12+13+...+1NS = 1 + \frac{1}{2} + \frac{1}{3} + ... + \frac{1}{N}S=1+21​+31​+...+N1​**

#include <stdio.h>

int main() {

int n;

double sum = 0.0;

printf("Please enter positive integer N: ");

scanf("%d", &n);

if (n <= 0) {

printf("Accept positive number only!\n");

} else {

printf("The sum is S = ");

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

sum += 1.0 / i;

printf("1/%d", i);

if (i < n) printf(" + ");

}

printf(" = %.6f\n", sum);

}

return 0;

}

**Bài 10: Tính tổng S=−1+12−13+14−...+(−1)N×1NS = -1 + \frac{1}{2} - \frac{1}{3} + \frac{1}{4} - ... + (-1)^N \times \frac{1}{N}S=−1+21​−31​+41​−...+(−1)N×N1​**

#include <stdio.h>

int main() {

int n;

double sum = 0.0;

printf("Please enter positive integer N: ");

scanf("%d", &n);

if (n <= 0) {

printf("Accept positive number only!\n");

} else {

printf("The sum is S = ");

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

sum += (i % 2 == 0 ? 1.0 : -1.0) / i;

printf("%s1/%d", (i % 2 == 0) ? "+" : "-", i);

}

printf(" = %.6f\n", sum);

}

return 0;

}

**Bài 11: Tính giai thừa N!=1×2×3×...×NN! = 1 \times 2 \times 3 \times ... \times NN!=1×2×3×...×N**

#include <stdio.h>

int main() {

int n;

long long factorial = 1;

printf("Please enter positive integer N: ");

scanf("%d", &n);

if (n < 0) {

printf("N must be greater than or equal 0!\n");

} else {

printf("Result: %d! = ", n);

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

factorial \*= i;

printf("%d", i);

if (i < n) printf("\*");

}

printf(" = %lld\n", factorial);

}

return 0;

}

**Bài 12: Tính XNX^NXN với X là cơ số và N là số mũ**

#include <stdio.h>

int main() {

int x, n;

long long result = 1;

printf("Please enter the base X: ");

scanf("%d", &x);

printf("Please enter the exponent N: ");

scanf("%d", &n);

if (n < 0) {

printf("The exponent N must be greater than or equal 0!\n");

} else {

printf("Result: %d^%d = ", x, n);

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

result \*= x;

}

printf("%lld\n", result);

}

return 0;

}

**Bài 13: Tính XNX^NXN với N có thể là số âm**

#include <stdio.h>

#include <math.h>

int main() {

int x, n;

double result;

printf("Please enter the base X: ");

scanf("%d", &x);

printf("Please enter the exponent N: ");

scanf("%d", &n);

result = pow(x, n);

printf("Result: %d^%d = %.5f\n", x, n, result);

return 0;

}

**Bài 14: Hiển thị tất cả ước số của một số nguyên dương N**

#include <stdio.h>

int main() {

int n;

printf("Please enter positive integer N: ");

scanf("%d", &n);

if (n <= 0) {

printf("Accept positive number only!\n");

} else {

printf("All divisors of %d are:\n", n);

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

if (n % i == 0) {

printf("%d ", i);

}

}

printf("\n");

}

return 0;

}

**Bài 15: Tính tổng tất cả các ước số của một số nguyên dương N**

#include <stdio.h>

int main() {

int n, sum = 0;

printf("Please enter positive integer N: ");

scanf("%d", &n);

if (n <= 0) {

printf("Accept positive number only!\n");

} else {

printf("The sum of all divisors of %d is S = ", n);

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

if (n % i == 0) {

sum += i;

printf("%d", i);

if (i < n) printf(" + ");

}

}

printf(" = %d\n", sum);

}

return 0;

}

**Bài 16: Đếm số ước của một số nguyên dương NNN**

#include <stdio.h>

int main() {

int n, count = 0;

printf("Please enter positive integer N: ");

scanf("%d", &n);

if (n <= 0) {

printf("Accept positive number only!\n");

} else {

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

if (n % i == 0) count++;

}

printf("The number of divisors of %d is %d\n", n, count);

}

return 0;

}

**Bài 17: Kiểm tra số hoàn hảo**

#include <stdio.h>

int main() {

int n, sum = 0;

printf("Please enter positive integer N: ");

scanf("%d", &n);

if (n <= 0) {

printf("Accept positive number only!\n");

} else {

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

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

}

if (sum == n)

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

else

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

}

return 0;

}

**Bài 18: Hiển thị tất cả các số hoàn hảo nhỏ hơn 1000**

#include <stdio.h>

int main() {

printf("All perfect numbers smaller than 1000 are:\n");

for (int n = 1; n < 1000; n++) {

int sum = 0;

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

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

}

if (sum == n) printf("%d ", n);

}

printf("\n");

return 0;

}

**Bài 19: Kiểm tra số nguyên tố**

#include <stdio.h>

#include <stdbool.h>

int main() {

int n;

bool isPrime = true;

printf("Please enter positive integer N: ");

scanf("%d", &n);

if (n <= 1) {

printf("Accept positive number only!\n");

} else {

for (int i = 2; i <= n / 2; i++) {

if (n % i == 0) {

isPrime = false;

break;

}

}

if (isPrime)

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

else

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

}

return 0;

}

**Bài 20: Hiển thị tất cả các số nguyên tố từ A đến B**

#include <stdio.h>

#include <stdbool.h>

int main() {

int a, b;

printf("Please enter the lower bound A: ");

scanf("%d", &a);

printf("Please enter the upper bound B: ");

scanf("%d", &b);

if (a <= 0 || b <= 0) {

printf("Both bounds must be positive integers!\n");

} else if (a > b) {

printf("The lower bound must be smaller than or equal to the upper bound!\n");

} else {

printf("All prime numbers from %d to %d are:\n", a, b);

for (int n = a; n <= b; n++) {

bool isPrime = true;

if (n < 2) continue;

for (int i = 2; i <= n / 2; i++) {

if (n % i == 0) {

isPrime = false;

break;

}

}

if (isPrime) printf("%d ", n);

}

printf("\n");

}

return 0;

}

**Bài 21: Tính Ước số chung lớn nhất (GCD) và Bội số chung nhỏ nhất (LCM) bằng thuật toán trừ**

#include <stdio.h>

int gcd(int a, int b) {

while (a != b) {

if (a > b) a -= b;

else b -= a;

}

return a;

}

int lcm(int a, int b) {

return (a \* b) / gcd(a, b);

}

int main() {

int a, b;

printf("Please enter the positive integer A: ");

scanf("%d", &a);

printf("Please enter the positive integer B: ");

scanf("%d", &b);

if (a <= 0 || b <= 0) {

printf("A and B must be greater than or equal 0!\n");

} else {

printf("The greatest common divisor of %d and %d is %d\n", a, b, gcd(a, b));

printf("The least common multiple of %d and %d is %d\n", a, b, lcm(a, b));

}

return 0;

}

**Bài 22: Tính GCD và LCM bằng thuật toán chia**

#include <stdio.h>

int gcd(int a, int b) {

while (b != 0) {

int temp = a % b;

a = b;

b = temp;

}

return a;

}

int lcm(int a, int b) {

return (a \* b) / gcd(a, b);

}

int main() {

int a, b;

printf("Please enter the positive integer A: ");

scanf("%d", &a);

printf("Please enter the positive integer B: ");

scanf("%d", &b);

if (a <= 0 || b <= 0) {

printf("A and B must be greater than or equal 0!\n");

} else {

printf("The greatest common divisor of %d and %d is %d\n", a, b, gcd(a, b));

printf("The least common multiple of %d and %d is %d\n", a, b, lcm(a, b));

}

return 0;

}

**Bài 23: Tính tổng các chữ số của một số nguyên dương NNN**

#include <stdio.h>

int main() {

long long n;

int sum = 0;

printf("Please enter the positive integer N: ");

scanf("%lld", &n);

if (n < 0) {

printf("N must be greater than or equal 0!\n");

} else {

while (n > 0) {

sum += n % 10;

n /= 10;

}

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

}

return 0;

}

**Bài 24: Hiển thị số đảo ngược của một số nguyên dương NNN**

#include <stdio.h>

int main() {

long long n, reversed = 0;

printf("Please enter the positive integer N: ");

scanf("%lld", &n);

if (n < 0) {

printf("N must be greater than or equal 0!\n");

} else {

while (n > 0) {

reversed = reversed \* 10 + n % 10;

n /= 10;

}

printf("The reverse number is %lld\n", reversed);

}

return 0;

}

**Bài 25: Hiển thị tất cả các số có 2 chữ số mà tổng các chữ số bằng NNN**

#include <stdio.h>

int main() {

int n;

printf("Please enter the positive integer N: ");

scanf("%d", &n);

if (n < 1 || n > 18) {

printf("Don’t have any 2-digit number that the sum of digits of that number equal to %d\n", n);

} else {

printf("All 2-digit numbers that the sum of digits of that number equal to %d are:\n", n);

for (int i = 10; i <= 99; i++) {

int tens = i / 10;

int units = i % 10;

if (tens + units == n) {

printf("%d ", i);

}

}

printf("\n");

}

return 0;

}

**Bài 26: Hiển thị bảng cửu chương của NNN**

#include <stdio.h>

int main() {

int n;

printf("Please enter the positive integer N: ");

scanf("%d", &n);

if (n <= 0) {

printf("N must be greater than 0!\n");

} else {

printf("The multiplication table of %d is:\n", n);

for (int i = 1; i <= 10; i++) {

printf("%d x %d = %d\n", i, n, i \* n);

}

}

return 0;

}

**Bài 27: Hiển thị bảng cửu chương từ 2 đến 9**

#include <stdio.h>

int main() {

for (int n = 2; n <= 9; n++) {

printf("The multiplication table of %d:\n", n);

for (int i = 1; i <= 10; i++) {

printf("%d x %d = %d\n", i, n, i \* n);

}

printf("\n");

}

return 0;

}

**Bài 28: Hiển thị bảng cửu chương của NNN với lựa chọn tiếp tục**

#include <stdio.h>

int main() {

int n;

char choice;

do {

printf("Please enter the positive integer N: ");

scanf("%d", &n);

if (n <= 0) {

printf("N must be greater than 0!\n");

} else {

printf("The multiplication table of %d is:\n", n);

for (int i = 1; i <= 10; i++) {

printf("%d x %d = %d\n", i, n, i \* n);

}

}

printf("Do you want to continue (y/n)? ");

scanf(" %c", &choice);

} while (choice == 'y' || choice == 'Y');

printf("Thank you for using our software!\n");

return 0;

}