**Programs**

**1. Display Basic Data Types and Their Sizes**

#include <stdio.h>

int main() {

printf("Size of int: %lu\n", sizeof(int));

printf("Size of float: %lu\n", sizeof(float));

printf("Size of char: %lu\n", sizeof(char));

printf("Size of double: %lu\n", sizeof(double));

return 0;

}

**2. Use of constants with #define (Area of circle)**

#include <stdio.h>

#define PI 3.14159

int main() {

float r, area;

printf("Enter radius: ");

scanf("%f", &r);

area = PI \* r \* r;

printf("Area = %.2f\n", area);

return 0;

}

**3.Celsius to Fahrenheit conversion**

#include <stdio.h>

int main() {

float C, F;

printf("Enter temperature in Celsius: ");

scanf("%f", &C);

F = (9\*C)/5 + 32;

printf("Fahrenheit = %.2f\n", F);

return 0;

}

**4.two numbers with a temp variable**

#include <stdio.h>

int main() {

int a, b, temp;

printf("Enter two numbers: ");

scanf("%d %d", &a, &b);

temp = a;

a = b;

b = temp;

printf("Swapped: a=%d, b=%d\n", a, b);

return 0;

}

**5.Swap two numbers without a temp variable**

#include <stdio.h>

int main() {

int a, b;

printf("Enter two numbers: ");

scanf("%d %d", &a, &b);

a = a + b;

b = a - b;

a = a - b;

printf("Swapped: a=%d, b=%d\n", a, b);

return 0;

}

**6.Calculate simple interest**

#include <stdio.h>

int main() {

float p, r, t, SI;

printf("Enter principal, rate, time: ");

scanf("%f %f %f", &p, &r, &t);

SI = (p \* r \* t)/100;

printf("Simple Interest = %.2f\n", SI);

return 0;

}

**7.Calculate compound interest**

#include <stdio.h>

#include <math.h>

int main() {

float p, r, t, CI;

printf("Enter principal, rate, time: ");

scanf("%f %f %f", &p, &r, &t);

CI = p \* pow((1 + r/100), t) - p;

printf("Compound Interest = %.2f\n", CI);

return 0;

}

**8.Print ASCII value of a character**

#include <stdio.h>

int main() {

char c;

printf("Enter a character: ");

scanf("%c", &c);

printf("ASCII value of %c = %d\n", c, c);

return 0;

}

**9.Check even or odd**

#include <stdio.h>

int main() {

int num;

printf("Enter a number: ");

scanf("%d", &num);

if(num % 2 == 0)

printf("%d is even\n", num);

else

printf("%d is odd\n", num);

return 0;

}

**10.Sum of two numbers**

#include <stdio.h>

int main() {

int a, b;

printf("Enter two numbers: ");

scanf("%d %d", &a, &b);

printf("Sum = %d\n", a + b);

return 0;

}

**11.Greatest of three numbers**

#include <stdio.h>

int main() {

int a, b, c, max;

printf("Enter three numbers: ");

scanf("%d %d %d", &a, &b, &c);

max = (a > b) ? a : b;

max = (max > c) ? max : c;

printf("Greatest number is %d\n", max);

return 0;

}

**12.Roots of quadratic equation**

#include <stdio.h>

#include <math.h>

int main() {

float a, b, c, d, root1, root2;

printf("Enter coefficients a, b, c: ");

scanf("%f %f %f", &a, &b, &c);

d = b\*b - 4\*a\*c;

if(d > 0) {

root1 = (-b + sqrt(d)) / (2\*a);

root2 = (-b - sqrt(d)) / (2\*a);

printf("Roots are real and distinct: %.2f, %.2f\n", root1, root2);

} else if(d == 0) {

root1 = root2 = -b / (2\*a);

printf("Roots are real and equal: %.2f\n", root1);

} else {

printf("Roots are imaginary\n");

}

return 0;

}

**13.Cube and square of a number**

#include <stdio.h>

int main() {

int n;

printf("Enter a number: ");

scanf("%d", &n);

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

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

return 0;

}

**14.Variable initialization at declaration**

#include <stdio.h>

int main() {

int a = 10, b = 20;

printf("a = %d, b = %d\n", a, b);

return 0;

}

**15.Use typedef to create a new data type**

#include <stdio.h>

typedef int marks;

int main() {

marks m1 = 95, m2 = 89;

printf("Marks: %d, %d\n", m1, m2);

return 0;

}

**16.Demonstrate storage classes (static example)**

#include <stdio.h>

void func() {

static int count = 0;

count++;

printf("Count = %d\n", count);

}

int main() {

func();

func();

func();

return 0;

}

**17.Triangle area from sides (Heron's formula)**

#include <stdio.h>

#include <math.h>

int main() {

float a, b, c, s, area;

printf("Enter three sides of triangle: ");

scanf("%f %f %f", &a, &b, &c);

s = (a + b + c)/2;

area = sqrt(s \* (s-a) \* (s-b) \* (s-c));

printf("Area = %.2f\n", area);

return 0;

}

**18.Add two long double values**

#include <stdio.h>

int main() {

long double a = 12345.6789, b = 98765.4321, sum;

sum = a + b;

printf("Sum = %.4Lf\n", sum);

return 0;

}

**19.Display float number with two decimal places**

#include <stdio.h>

int main() {

float num;

printf("Enter a float number: ");

scanf("%f", &num);

printf("Number with 2 decimals: %.2f\n", num);

return 0;

}

**20.Evaluate a Complex Arithmetic Expression**

#include <stdio.h>

int main() {

int a = 5, b = 8, c = 4, result;

// Expression: (a + b) \* c - b / a

result = (a + b) \* c - b / a;

printf("Result of (a + b) \* c - b / a = %d\n", result);

return 0;

}

**21.Basic Arithmetic Operations**

#include <stdio.h>

int main() {

int a, b;

printf("Enter two integers: ");

scanf("%d %d", &a, &b);

printf("Sum = %d\n", a + b);

printf("Difference = %d\n", a - b);

printf("Product = %d\n", a \* b);

printf("Quotient = %d\n", a / b);

printf("Remainder = %d\n", a % b);

return 0;

}

**22.Evaluate Arithmetic Expression**

#include <stdio.h>

int main() {

int a, b, c;

printf("Enter three integers: ");

scanf("%d %d %d", &a, &b, &c);

int result = a + b \* c - (a / b);

printf("Result = %d\n", result);

return 0;

}

**23**. **Operator Precedence and Associativity**

#include <stdio.h>

int main() {

int x = 5, y = 10, z = 15, result;

result = x + y \* z / x - y;

printf("Result = %d\n", result);

return 0;

}  
**24.Prefix and Postfix Increment/Decrement**

#include <stdio.h>

int main() {

int a = 5, b;

b = ++a; // prefix

printf("After prefix increment: a = %d, b = %d\n", a, b);

a = 5;

b = a++; // postfix

printf("After postfix increment: a = %d, b = %d\n", a, b);

return 0;

}

**25.Circle Area and Circumference**

#include <stdio.h>

#define PI 3.14159

int main() {

float r, area, circumference;

printf("Enter radius: ");

scanf("%f", &r);

area = PI \* r \* r;

circumference = 2 \* PI \* r;

printf("Area = %.2f\nCircumference = %.2f\n", area, circumference);

return 0;

}

**26.Quadratic Expression Evaluation**

#include <stdio.h>

int main() {

int a, b, c, x;

printf("Enter a, b, c and x: ");

scanf("%d %d %d %d", &a, &b, &c, &x);

int result = a\*x\*x + b\*x + c;

printf("Result = %d\n", result);

return 0;

}

**27.Swap Two Numbers using XOR**

#include <stdio.h>

int main() {

int a, b;

printf("Enter two numbers: ");

scanf("%d %d", &a, &b);

a = a ^ b;

b = a ^ b;

a = a ^ b;

printf("Swapped: a = %d, b = %d\n", a, b);

return 0;

}

**28.Demonstrate Relational Operators**

#include <stdio.h>

int main() {

int a, b;

printf("Enter two numbers: ");

scanf("%d %d", &a, &b);

printf("%d > %d is %d\n", a, b, a > b);

printf("%d == %d is %d\n", a, b, a == b);

return 0;

}

**29.Logical Operators Demonstration**

#include <stdio.h>

int main() {

int a = 1, b = 0;

printf("a && b = %d\n", a && b); // Logical AND

printf("a || b = %d\n", a || b); // Logical OR

printf("!a = %d\n", !a); // Logical NOT

return 0;

}

**30.Ternary Operator Example**

#include <stdio.h>

int main() {

int a, b, max;

printf("Enter two numbers: ");

scanf("%d %d", &a, &b);

max = (a > b) ? a : b;

printf("Maximum = %d\n", max);

return 0;

}

**31**.**Compound Assignment Operators**

#include <stdio.h>

int main() {

int a = 10;

a += 5; // a = a + 5

printf("a += 5: %d\n", a);

a \*= 2; // a = a \* 2

printf("a \*= 2: %d\n", a);

return 0;

}

**32.Sum of Digits of a Number**

#include <stdio.h>

int main() {

int num, sum = 0, temp;

printf("Enter a number: ");

scanf("%d", &num);

temp = num;

while (temp != 0) {

sum += temp % 10;

temp /= 10;

}

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

return 0;

}

**33. Check Divisibility by 3 and 5**

#include <stdio.h>

int main() {

int num;

printf("Enter a number: ");

scanf("%d", &num);

if (num % 3 == 0 && num % 5 == 0)

printf("%d is divisible by both 3 and 5\n", num);

else

printf("%d is NOT divisible by both 3 and 5\n", num);

return 0;

}

**34. Largest of Four Numbers**

#include <stdio.h>

int main() {

int a, b, c, d, max;

printf("Enter four numbers: ");

scanf("%d %d %d %d", &a, &b, &c, &d);

max = (a > b) ? a : b;

max = (max > c) ? max : c;

max = (max > d) ? max : d;

printf("Largest number = %d\n", max);

return 0;

}

**35. Operators Demonstration**

#include <stdio.h>

int main() {

int a = 5, b = 9;

printf("a & b = %d\n", a & b);

printf("a | b = %d\n", a | b);

printf("a ^ b = %d\n", a ^ b);

printf("~a = %d\n", ~a);

printf("b << 1 = %d\n", b << 1);

printf("b >> 1 = %d\n", b >> 1);

return 0;

}

**36.Toggle a Particular Bit**

#include <stdio.h>

int main() {

int num, pos;

printf("Enter number and bit position to toggle: ");

scanf("%d %d", &num, &pos);

num = num ^ (1 << (pos - 1));

printf("Result after toggling bit %d: %d\n", pos, num);

return 0;

}

**37**.**Power of a Number Using pow()**

#include <stdio.h>

#include <math.h>

int main() {

double base, exponent, result;

printf("Enter base and exponent: ");

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

result = pow(base, exponent);

printf("%.2lf^%.2lf = %.2lf\n", base, exponent, result);

return 0;

}

**38.Evaluate Nested Expression**

#include <stdio.h>

int main() {

int a = 4, b = 2, c = 3, result;

result = (a + b) \* (c - b) + b / a;

printf("Result = %d\n", result);

return 0;

}

**39.Character Type Check**

#include <stdio.h>

int main() {

char ch;

printf("Enter a character: ");

scanf("%c", &ch);

if ((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z'))

printf("%c is a letter\n", ch);

else if (ch >= '0' && ch <= '9')

printf("%c is a digit\n", ch);

else

printf("%c is a special character\n", ch);

return 0;

}

**40.Typecasting in Expressions**

#include <stdio.h>

int main() {

int a = 5, b = 2;

float result;

result = (float)a / b; // Typecast to float for accurate division

printf("Result of division = %.2f\n", result);

return 0;

}

**41.Program to check if a number is positive, negative, or zero.**

#include <stdio.h>

int main() {

int num;

printf("Enter a number: ");

scanf("%d", &num);

if (num > 0)

printf("Positive\n");

else if (num < 0)

printf("Negative\n");

else

printf("Zero\n");

return 0;

}

**42.Find Largest of Two Numbers Using if-else**

#include <stdio.h>

int main() {

int a, b;

printf("Enter two numbers: ");

scanf("%d %d", &a, &b);

if (a > b)

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

else

printf("%d is larger\n", b);

return 0;

}

**43.Find Largest of Three Numbers Using Nested if-else**

#include <stdio.h>

int main() {

int a, b, c, largest;

printf("Enter three numbers: ");

scanf("%d %d %d", &a, &b, &c);

if (a > b) {

if (a > c)

largest = a;

else

largest = c;

} else {

if (b > c)

largest = b;

else

largest = c;

}

printf("Largest number is %d\n", largest);

return 0;

}

**44.Check Vowel or Consonant Using if-else**

#include <stdio.h>

int main() {

char ch;

printf("Enter a character: ");

scanf("%c", &ch);

if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u' ||

ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U')

printf("%c is a vowel\n", ch);

else

printf("%c is a consonant\n", ch);

return 0;

}

**45.Check Leap Year**

#include <stdio.h>

int main() {

int year;

printf("Enter year: ");

scanf("%d", &year);

if ((year % 400 == 0) || (year % 4 == 0 && year % 100 != 0))

printf("%d is a leap year\n", year);

else

printf("%d is not a leap year\n", year);

return 0;

}

**46.Student Grade Based on Marks**

#include <stdio.h>

int main() {

int marks;

printf("Enter marks: ");

scanf("%d", &marks);

if (marks >= 90)

printf("Grade: A\n");

else if (marks >= 75)

printf("Grade: B\n");

else if (marks >= 60)

printf("Grade: C\n");

else if (marks >= 50)

printf("Grade: D\n");

else

printf("Grade: F\n");

return 0;

}

**47.Display Day of Week Using Switch**

#include <stdio.h>

int main() {

int day;

printf("Enter day number (1-7): ");

scanf("%d", &day);

switch (day) {

case 1: printf("Sunday\n"); break;

case 2: printf("Monday\n"); break;

case 3: printf("Tuesday\n"); break;

case 4: printf("Wednesday\n"); break;

case 5: printf("Thursday\n"); break;

case 6: printf("Friday\n"); break;

case 7: printf("Saturday\n"); break;

default: printf("Invalid day\n");

}

return 0;

}

**48.Character Type Check Using Switch-Case**

#include <stdio.h>

int main() {

char ch;

printf("Enter a character: ");

scanf("%c", &ch);

switch (ch) {

case 'A': case 'E': case 'I': case 'O': case 'U':

case 'a': case 'e': case 'i': case 'o': case 'u':

printf("Vowel\n"); break;

default:

if ((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z'))

printf("Consonant\n");

else if (ch >= '0' && ch <= '9')

printf("Digit\n");

else

printf("Special character\n");

}

return 0;

}

**49.Calculate Electricity Bill**

#include <stdio.h>

int main() {

int units;

float bill;

printf("Enter units consumed: ");

scanf("%d", &units);

if (units <= 100)

bill = units \* 1.5;

else if (units <= 200)

bill = 100 \* 1.5 + (units - 100) \* 2.0;

else

bill = 100 \* 1.5 + 100 \* 2.0 + (units - 200) \* 3.0;

printf("Electricity bill = %.2f\n", bill);

return 0;

}

**50.Calculate Factorial of a Number Using if-else**

#include <stdio.h>

int main() {

int num, i;

unsigned long long factorial = 1;

printf("Enter a positive integer: ");

scanf("%d", &num);

if (num < 0)

printf("Factorial of negative number doesn't exist.\n");

else {

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

factorial \*= i;

}

printf("Factorial of %d = %llu\n", num, factorial);

}

return 0;

}

**51.Check Whether a Number is Prime Using if-else and Loop**

#include <stdio.h>

int main() {

int num, i, isPrime = 1;

printf("Enter a positive integer: ");

scanf("%d", &num);

if (num <= 1) {

isPrime = 0;

} else {

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

if (num % i == 0) {

isPrime = 0;

break;

}

}

}

if (isPrime)

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

else

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

return 0;

}

**52.Absolute Value of a Number**

#include <stdio.h>

int main() {

int num;

printf("Enter number: ");

scanf("%d", &num);

if (num < 0)

num = -num;

printf("Absolute value = %d\n", num);

return 0;

}

**53.Month Name from Number**

#include <stdio.h>

int main() {

int month;

printf("Enter month number (1-12): ");

scanf("%d", &month);

switch(month) {

case 1: printf("January\n"); break;

case 2: printf("February\n"); break;

case 3: printf("March\n"); break;

case 4: printf("April\n"); break;

case 5: printf("May\n"); break;

case 6: printf("June\n"); break;

case 7: printf("July\n"); break;

case 8: printf("August\n"); break;

case 9: printf("September\n"); break;

case 10: printf("October\n"); break;

case 11: printf("November\n"); break;

case 12: printf("December\n"); break;

default: printf("Invalid month\n");

}

return 0;

}

**54.Commission Based on Sales Amount**

#include <stdio.h>

int main() {

float sales, commission;

printf("Enter sales amount: ");

scanf("%f", &sales);

if (sales > 10000)

commission = sales \* 0.10;

else if (sales > 5000)

commission = sales \* 0.07;

else

commission = sales \* 0.05;

printf("Commission = %.2f\n", commission);

return 0;

}

**55.Nature of Roots of Quadratic Equation**

#include <stdio.h>

int main() {

float a, b, c, d;

printf("Enter coefficients a, b, c: ");

scanf("%f %f %f", &a, &b, &c);

d = b\*b - 4\*a\*c;

if (d > 0)

printf("Roots are real and distinct\n");

else if (d == 0)

printf("Roots are real and equal\n");

else

printf("Roots are imaginary\n");

return 0;

}

**56.Check Eligibility for Voting and Driving License**

#include <stdio.h>

int main() {

int age;

printf("Enter age: ");

scanf("%d", &age);

if (age >= 18) {

printf("Eligible for voting\n");

if (age >= 21)

printf("Eligible for driving license\n");

else

printf("Not eligible for driving license\n");

} else {

printf("Not eligible for voting or driving license\n");

}

return 0;

}  
**57.Simple Calculator Using Switch**

#include <stdio.h>

int main() {

char op;

float num1, num2;

printf("Enter operator (+, -, \*, /): ");

scanf(" %c", &op);

printf("Enter two numbers: ");

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

switch(op) {

case '+': printf("%.2f + %.2f = %.2f\n", num1, num2, num1 + num2); break;

case '-': printf("%.2f - %.2f = %.2f\n", num1, num2, num1 - num2); break;

case '\*': printf("%.2f \* %.2f = %.2f\n", num1, num2, num1 \* num2); break;

case '/':

if (num2 != 0)

printf("%.2f / %.2f = %.2f\n", num1, num2, num1 / num2);

else

printf("Division by zero error!\n");

break;

default: printf("Invalid operator\n");

}

return 0;

}

**58. Check Divisibility by 5 and 7**

#include <stdio.h>

int main() {

int num;

printf("Enter number: ");

scanf("%d", &num);

if (num % 5 == 0 && num % 7 == 0)

printf("%d is divisible by both 5 and 7\n", num);

else

printf("%d is not divisible by both 5 and 7\n", num);

return 0;

}

**59.Character Type Check (Alphabet, Digit, Special)**

#include <stdio.h>

int main() {

char ch;

printf("Enter a character: ");

scanf("%c", &ch);

if ((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z'))

printf("%c is an alphabet\n", ch);

else if (ch >= '0' && ch <= '9')

printf("%c is a digit\n", ch);

else

printf("%c is a special character\n", ch);

return 0;

}

**60.Swap Two Numbers if First Is Greater**

#include <stdio.h>

int main() {

int a, b, temp;

printf("Enter two numbers: ");

scanf("%d %d", &a, &b);

if (a > b) {

temp = a;

a = b;

b = temp;

}

printf("After swapping if needed: a = %d, b = %d\n", a, b);

return 0;

}

**61.Print first N natural numbers using for loop**

#include <stdio.h>

int main() {

int i, n;

printf("Enter n: ");

scanf("%d", &n);

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

printf("%d ", i);

printf("\n");

return 0;

}

**62.Print first N natural numbers in reverse using while loop**

#include <stdio.h>

int main() {

int i, n;

printf("Enter n: ");

scanf("%d", &n);

i = n;

while(i >= 1) {

printf("%d ", i);

i--;

}

printf("\n");

return 0;

}

**63.Print first N odd numbers using do-while loop**

#include <stdio.h>

int main() {

int i = 1, count = 0, n;

printf("Enter n: ");

scanf("%d", &n);

do {

if(i % 2 != 0){

printf("%d ", i);

count++;

}

i++;

} while(count < n);

printf("\n");

return 0;

}

**64.Calculate factorial of a number using for loop**

#include <stdio.h>

int main() {

int i, n;

unsigned long long fact = 1;

printf("Enter a number: ");

scanf("%d", &n);

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

fact \*= i;

printf("Factorial of %d = %llu\n", n, fact);

return 0;

}

**65.Display sum of first N natural numbers using while loop**

#include <stdio.h>

int main() {

int i = 1, n, sum = 0;

printf("Enter n: ");

scanf("%d", &n);

while(i <= n) {

sum += i;

i++;

}

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

return 0;

}

**66.Print multiplication table of a number using for loop**

#include <stdio.h>

int main() {

int n, i;

printf("Enter a number: ");

scanf("%d", &n);

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

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

return 0;

}

**67.Print all prime numbers between two intervals**

#include <stdio.h>

int main() {

int i, j, n1, n2, isPrime;

printf("Enter two numbers(intervals): ");

scanf("%d %d", &n1, &n2);

for(i = n1; i <= n2; i++) {

if (i < 2) continue;

isPrime = 1;

for(j=2; j\*j <= i; j++) {

if (i % j == 0) {

isPrime = 0;

break;

}

}

if (isPrime)

printf("%d ", i);

}

printf("\n");

return 0;

}  
**68.Find GCD of two numbers using while loop**

#include <stdio.h>

int main() {

int a, b, temp;

printf("Enter two numbers: ");

scanf("%d %d", &a, &b);

while(b != 0) {

temp = b;

b = a % b;

a = temp;

}

printf("GCD = %d\n", a);

return 0;

}

**69.Calculate sum of digits of a number using do-while loop**

#include <stdio.h>

int main() {

int num, sum = 0, rem;

printf("Enter a number: ");

scanf("%d", &num);

do {

rem = num % 10;

sum += rem;

num /= 10;

} while(num != 0);

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

return 0;

}

**70.Print Fibonacci series up to N terms**

#include <stdio.h>

int main() {

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

printf("Enter number of terms: ");

scanf("%d", &n);

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

printf("%d ", t1);

nextTerm = t1 + t2;

t1 = t2;

t2 = nextTerm;

}

printf("\n");

return 0;

}

**71.Find power of a number using for loop**

#include <stdio.h>

int main() {

int base, exp, i;

long long power = 1;

printf("Enter base and exponent: ");

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

for(i = 1; i <= exp; i++)

power \*= base;

printf("%d^%d = %lld\n", base, exp, power);

return 0;

}

**72.Count number of digits in an integer**

#include <stdio.h>

int main() {

int num, count = 0;

printf("Enter a number: ");

scanf("%d", &num);

while(num != 0) {

num /= 10;

count++;

}

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

return 0;

}

**73.Reverse a number and check if it is palindrome**

#include <stdio.h>

int main() {

int num, rev = 0, temp, rem;

printf("Enter number: ");

scanf("%d", &num);

temp = num;

while (temp != 0) {

rem = temp % 10;

rev = rev \* 10 + rem;

temp /= 10;

}

printf("Reversed number = %d\n", rev);

if (num == rev)

printf("Number is palindrome.\n");

else

printf("Number is not palindrome.\n");

return 0;

}

**74.Print all Armstrong numbers between 1 and 1000**

#include <stdio.h>

#include <math.h>

int main() {

int num, temp, sum, rem, digits, i;

for(num = 1; num <= 1000; num++) {

temp = num;

sum = 0;

digits = (int)log10(num) + 1;

while(temp != 0) {

rem = temp % 10;

sum += pow(rem, digits);

temp /= 10;

}

if (num == sum)

printf("%d ", num);

}

printf("\n");

return 0;

}

**75.Print Pascal’s Triangle (first N rows)**

#include <stdio.h>

int factorial(int n) {

int i, fact = 1;

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

fact \*= i;

return fact;

}

int main() {

int rows, i, j;

printf("Enter number of rows: ");

scanf("%d", &rows);

for (i = 0; i < rows; i++) {

for (j = 0; j <= i; j++)

printf("%d ", factorial(i) / (factorial(j)\*factorial(i-j)));

printf("\n");

}

return 0;

}

**76.Find sum of squares of first N natural numbers**

#include <stdio.h>

int main() {

int n, i, sum=0;

printf("Enter n: ");

scanf("%d", &n);

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

sum += i\*i;

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

return 0;

}

**77.Generate HCF using Euclid’s algorithm**

#include <stdio.h>

int main() {

int a, b, hcf, i;

printf("Enter two numbers: ");

scanf("%d %d", &a, &b);

for(i = 1; i <= a && i <= b; ++i) {

if(a % i == 0 && b % i == 0)

hcf = i;

}

printf("HCF = %d\n", hcf);

return 0;

}

**78.Sum of first N even numbers**

#include <stdio.h>

int main() {

int n, i, sum = 0;

printf("Enter n: ");

scanf("%d", &n);

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

sum += 2 \* i;

printf("Sum of first %d even numbers = %d\n", n, sum);

return 0;

}

**79.Print diamond pattern of stars with N rows**

#include <stdio.h>

int main() {

int n, i, j;

printf("Enter number of rows: ");

scanf("%d", &n);

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

for (j=i; j<n; j++)

printf(" ");

for (j=1; j<=2\*i-1; j++)

printf("\*");

printf("\n");

}

for (i=n-1; i>=1; i--) {

for (j=n; j>i; j--)

printf(" ");

for (j=1; j<=2\*i-1; j++)

printf("\*");

printf("\n");

}

return 0;

}

**80.Count number of vowels and consonants in a string (looping char array)**

#include <stdio.h>

int main() {

char str[100];

int i, vowels=0, consonants=0;

printf("Enter a string: ");

fgets(str, sizeof(str), stdin);

for (i=0; str[i] != '\0'; i++) {

char ch = str[i];

if ((ch >= 'A' && ch<= 'Z') || (ch >= 'a' && ch <= 'z')) {

if (ch=='a' || ch=='e' || ch=='i' || ch=='o' || ch=='u' ||

ch=='A' || ch=='E' || ch=='I' || ch=='O' || ch=='U')

vowels++;

else

consonants++;

}

}

printf("Vowels = %d, Consonants = %d\n", vowels, consonants);

return 0;

}

**81.Declare and initialize an array and display elements**

#include <stdio.h>

int main() {

int arr[5] = {1, 2, 3, 4, 5};

int i;

for (i = 0; i < 5; i++)

printf("%d ", arr[i]);

printf("\n");

return 0;

}

**82.Input n elements in array and find their sum**

#include <stdio.h>

int main() {

int arr[100], n, i, sum = 0;

printf("Enter number of elements: ");

scanf("%d", &n);

printf("Enter elements:\n");

for (i=0; i<n; i++) {

scanf("%d", &arr[i]);

sum += arr[i];

}

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

return 0;

}

**83.Find the largest and smallest element in an array**

#include <stdio.h>

int main() {

int arr[100], n, i, max, min;

printf("Enter number of elements: ");

scanf("%d", &n);

printf("Enter elements:\n");

for (i=0; i<n; i++) {

scanf("%d", &arr[i]);

}

max = min = arr[0];

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

if (arr[i] > max) max = arr[i];

if (arr[i] < min) min = arr[i];

}

printf("Max = %d, Min = %d\n", max, min);

return 0;

}

**84.Calculate average of n array elements**

#include <stdio.h>

int main() {

int arr[100], n, i, sum=0;

float avg;

printf("Enter number of elements: ");

scanf("%d", &n);

printf("Enter elements:\n");

for(i=0; i<n; i++) {

scanf("%d", &arr[i]);

sum += arr[i];

}

avg = (float)sum / n;

printf("Average = %.2f\n", avg);

return 0;

}

**85.Reverse an array**

#include <stdio.h>

int main() {

int arr[100], n, i, temp;

printf("Enter number of elements: ");

scanf("%d", &n);

printf("Enter elements:\n");

for(i=0; i<n; i++)

scanf("%d", &arr[i]);

for(i=0; i<n/2; i++) {

temp = arr[i];

arr[i] = arr[n-i-1];

arr[n-i-1] = temp;

}

printf("Reversed array: ");

for(i=0; i<n; i++)

printf("%d ", arr[i]);

printf("\n");

return 0;

}

**86.Find second largest element in array**

#include <stdio.h>

int main() {

int arr[100], n, i, max, secondMax;

printf("Enter number of elements: ");

scanf("%d", &n);

printf("Enter elements:\n");

for(i=0; i<n; i++)

scanf("%d", &arr[i]);

if (arr[0] > arr[1]) {

max = arr[0];

secondMax = arr[1];

} else {

max = arr[1];

secondMax = arr[0];

}

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

if (arr[i] > max) {

secondMax = max;

max = arr[i];

} else if (arr[i] > secondMax && arr[i] != max) {

secondMax = arr[i];

}

}

printf("Second largest element: %d\n", secondMax);

return 0;

}

**87.Calculate sum of all even numbers in array**

#include <stdio.h>

int main() {

int arr[100], n, i, sum=0;

printf("Enter number of elements: ");

scanf("%d", &n);

printf("Enter elements:\n");

for(i=0; i<n; i++) {

scanf("%d", &arr[i]);

if (arr[i] % 2 == 0)

sum += arr[i];

}

printf("Sum of even elements = %d\n", sum);

return 0;

}

**88.Count frequency of an element in array**

#include <stdio.h>

int main() {

int arr[100], n, i, ele, count=0;

printf("Enter number of elements: ");

scanf("%d", &n);

printf("Enter elements:\n");

for(i=0; i<n; i++)

scanf("%d", &arr[i]);

printf("Enter element to find frequency: ");

scanf("%d", &ele);

for(i=0; i<n; i++) {

if (arr[i] == ele)

count++;

}

printf("Frequency of %d is %d\n", ele, count);

return 0;

}

**89,Linear search in array**

#include <stdio.h>

int main() {

int arr[100], n, i, ele, found=0;

printf("Enter number of elements: ");

scanf("%d", &n);

printf("Enter elements:\n");

for(i=0; i<n; i++)

scanf("%d", &arr[i]);

printf("Enter element to search: ");

scanf("%d", &ele);

for(i=0; i<n; i++) {

if (arr[i] == ele) {

found = 1;

break;

}

}

if (found)

printf("%d is found at position %d\n", ele, i+1);

else

printf("%d is not found\n", ele);

return 0;

}

**90.Bubble sort array in ascending order**

#include <stdio.h>

int main() {

int arr[100], n, i, j, temp;

printf("Enter number of elements: ");

scanf("%d", &n);

printf("Enter elements:\n");

for(i=0; i<n; i++)

scanf("%d", &arr[i]);

for(i=0; i<n-1; i++) {

for(j=0; j<n-i-1; j++) {

if(arr[j] > arr[j+1]) {

temp=arr[j];

arr[j]=arr[j+1];

arr[j+1]=temp;

}

}

}

printf("Sorted array: ");

for(i=0; i<n; i++)

printf("%d ", arr[i]);

printf("\n");

return 0;

}

**91.Sum of diagonal elements of a square matrix**

#include <stdio.h>

int main() {

int matrix[10][10], n, i, j, sum=0;

printf("Enter size of square matrix: ");

scanf("%d", &n);

printf("Enter matrix elements:\n");

for(i=0; i<n; i++) {

for(j=0; j<n; j++)

scanf("%d", &matrix[i][j]);

}

for(i=0; i<n; i++)

sum += matrix[i][i];

printf("Sum of diagonal elements = %d\n", sum);

return 0;

}

**92.Transpose of a matrix**

#include <stdio.h>

int main() {

int matrix[10][10], transpose[10][10], row, col, i, j;

printf("Enter rows and columns: ");

scanf("%d %d", &row, &col);

printf("Enter matrix elements:\n");

for(i=0; i<row; i++)

for(j=0; j<col; j++)

scanf("%d", &matrix[i][j]);

for(i=0; i<row; i++)

for(j=0; j<col; j++)

transpose[j][i] = matrix[i][j];

printf("Transpose matrix:\n");

for(i=0; i<col; i++) {

for(j=0; j<row; j++)

printf("%d ", transpose[i][j]);

printf("\n");

}

return 0;

}

**93.String input and output using gets and puts**

#include <stdio.h>

int main() {

char str[100];

printf("Enter a string: ");

fgets(str, sizeof(str), stdin);

printf("You entered: ");

puts(str);

return 0;

}

**94.Calculate length of a string without using strlen**

#include <stdio.h>

int main() {

char str[100];

int i, length = 0;

printf("Enter a string: ");

fgets(str, sizeof(str), stdin);

for(i=0; str[i] != '\0'; i++) {

if (str[i] != '\n') length++;

}

printf("Length of string = %d\n", length);

return 0;

}

**95.Compare two strings without using strcmp**

#include <stdio.h>

int main() {

char str1[100], str2[100];

int i, flag = 0;

printf("Enter first string: ");

fgets(str1, sizeof(str1), stdin);

printf("Enter second string: ");

fgets(str2, sizeof(str2), stdin);

for(i=0; str1[i] != '\0' && str2[i] != '\0'; i++) {

if(str1[i] != str2[i]) {

flag = 1;

break;

}

}

if(flag == 0 && str1[i] == '\0' && str2[i] == '\0')

printf("Strings are equal\n");

else

printf("Strings are not equal\n");

return 0;

}

**96.Concatenate two strings without strcat**

#include <stdio.h>

int main() {

char str1[100], str2[100];

int i, j;

printf("Enter first string: ");

fgets(str1, sizeof(str1), stdin);

printf("Enter second string: ");

fgets(str2, sizeof(str2), stdin);

for(i=0; str1[i] != '\0'; i++);

for(j=0; str2[j] != '\0'; j++, i++)

str1[i] = str2[j];

str1[i] = '\0';

printf("Concatenated string: %s\n", str1);

return 0;

}

**97.Copy one string to another without strcpy**

#include <stdio.h>

int main() {

char str1[100], str2[100];

int i;

printf("Enter a string: ");

fgets(str1, sizeof(str1), stdin);

for(i=0; str1[i] != '\0'; i++)

str2[i] = str1[i];

str2[i] = '\0';

printf("Copied string: %s\n", str2);

return 0;

}

**98.Count vowels, consonants, digits, and spaces in a string**

#include <stdio.h>

int main() {

char str[100];

int i, vowels=0, consonants=0, digits=0, spaces=0;

printf("Enter a string: ");

fgets(str, sizeof(str), stdin);

for(i=0; str[i] != '\0'; i++) {

char ch = str[i];

if ((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z')) {

if (ch=='a'||ch=='e'||ch=='i'||ch=='o'||ch=='u'||ch=='A'||ch=='E'||ch=='I'||ch=='O'||ch=='U')

vowels++;

else

consonants++;

} else if (ch >= '0' && ch<='9')

digits++;

else if (ch == ' ')

spaces++;

}

printf("Vowels = %d\nConsonants = %d\nDigits = %d\nSpaces = %d\n", vowels, consonants, digits, spaces);

return 0;

}

**99.Reverse a string**

#include <stdio.h>

#include <string.h>

int main() {

char str[100], rev[100];

int len, i, j;

printf("Enter a string: ");

fgets(str, sizeof(str), stdin);

len = 0;

while (str[len] != '\0')

len++;

for(i=len-2, j=0; i>=0; i--, j++)

rev[j] = str[i];

rev[j] = '\0';

printf("Reversed string: %s\n", rev);

return 0;

}

**100.Check if a string is palindrome**

#include <stdio.h>

#include <string.h>

int main() {

char str[100];

int len, i, flag=1;

printf("Enter a string: ");

fgets(str, sizeof(str), stdin);

len = 0;

while (str[len] != '\0')

len++;

len -= 1; // Exclude newline

for(i=0; i<len/2; i++) {

if(str[i] != str[len-i-1]) {

flag = 0;

break;

}

}

if(flag)

printf("String is palindrome\n");

else

printf("String is not palindrome\n");

return 0;

}  
**101.Write a simple function to print "Hello, World!"**

#include <stdio.h>

void greet() {

printf("Hello, World!\n");

}

int main() {

greet();

return 0;

}

**102.Function to add two integers and return the sum**

#include <stdio.h>

int add(int a, int b) {

return a + b;

}

int main() {

int result = add(5, 7);

printf("Sum = %d\n", result);

return 0;

}

**103.Write a function to find the factorial of a number**

#include <stdio.h>

unsigned long long factorial(int n) {

if (n <= 1) return 1;

else return n \* factorial(n-1);

}

int main() {

int num = 6;

printf("Factorial of %d is %llu\n", num, factorial(num));

return 0;

}

**104.Function to calculate power using recursion**

#include <stdio.h>

int power(int base, int exp) {

if (exp == 0) return 1;

else return base \* power(base, exp -1);

}

int main() {

printf("3^4 = %d\n", power(3,4));

return 0;

}

**105.Write a function to check if a number is prime**

#include <stdio.h>

int isPrime(int n) {

if (n <= 1) return 0;

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

if (n % i == 0) return 0;

}

return 1;

}

int main() {

int num = 29;

if (isPrime(num)) printf("%d is prime\n", num);

else printf("%d is not prime\n", num);

return 0;

}

**106.Function with no return and no parameters**

#include <stdio.h>

void display() {

printf("Function with no return and no parameters\n");

}

int main() {

display();

return 0;

}

**107.Function with no return but with parameters**

#include <stdio.h>

void printSum(int a, int b) {

printf("Sum = %d\n", a + b);

}

int main() {

printSum(10, 15);

return 0;

}

**108.Function to swap two numbers using pointers**

#include <stdio.h>

void swap(int \*a, int \*b) {

int temp = \*a;

\*a = \*b;

\*b = temp;

}

int main() {

int x = 5, y = 10;

swap(&x, &y);

printf("After swap: x = %d, y = %d\n", x, y);

return 0;

}

**109.Use of static variable inside a function**

#include <stdio.h>

void counter() {

static int count = 0;

count++;

printf("Count = %d\n", count);

}

int main() {

counter();

counter();

counter();

return 0;

}

**110.Function to find GCD of two numbers**

#include <stdio.h>

int gcd(int a, int b) {

if (b == 0) return a;

else return gcd(b, a % b);

}

int main() {

printf("GCD of 36 and 60 = %d\n", gcd(36, 60));

return 0;

}

**111.Function to find LCM using GCD**

#include <stdio.h>

int gcd(int a, int b) {

return (b == 0) ? a : gcd(b, a % b);

}

int lcm(int a, int b) {

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

}

int main() {

printf("LCM of 12 and 15 = %d\n", lcm(12, 15));

return 0;

}

**112.Recursive function to compute nth Fibonacci number**

#include <stdio.h>

int fibonacci(int n) {

if (n == 0) return 0;

if (n == 1) return 1;

return fibonacci(n-1) + fibonacci(n-2);

}

int main() {

int n = 7;

printf("Fibonacci number at position %d is %d\n", n, fibonacci(n));

return 0;

}

**113.Function to check palindrome string (using pointers)**

#include <stdio.h>

#include <string.h>

int isPalindrome(char \*str) {

char \*end = str + strlen(str) - 1;

while(str < end) {

if(\*str != \*end)

return 0;

str++;

end--;

}

return 1;

}

int main() {

char str[] = "madam";

if (isPalindrome(str))

printf("Palindrome\n");

else

printf("Not a palindrome\n");

return 0;

}

**114.Function to print an array**

#include <stdio.h>

void printArray(int arr[], int n) {

for(int i=0; i<n; i++)

printf("%d ", arr[i]);

printf("\n");

}

int main() {

int arr[] = {10, 20, 30, 40};

printArray(arr, 4);

return 0;

}

**115.Function to find average of array elements**

#include <stdio.h>

double average(int arr[], int n) {

int sum = 0;

for(int i=0; i<n; i++)

sum += arr[i];

return (double)sum / n;

}

int main() {

int arr[] = {10, 20, 30, 40, 50};

printf("Average = %.2lf\n", average(arr, 5));

return 0;

}

**116.Function to reverse a string**

#include <stdio.h>

#include <string.h>

void reverseString(char str[]) {

int i, len = strlen(str);

for(i=0; i<len/2; i++) {

char temp = str[i];

str[i] = str[len - i - 1];

str[len - i -1] = temp;

}

}

int main() {

char str[] = "embedded";

reverseString(str);

printf("Reversed string: %s\n", str);

return 0;

}

**117.Function to count vowels in string**

#include <stdio.h>

#include <string.h>

int countVowels(char str[]) {

int i, count = 0;

for(i=0; str[i]!='\0'; i++) {

char c = str[i];

if(c=='a'||c=='e'||c=='i'||c=='o'||c=='u'||

c=='A'||c=='E'||c=='I'||c=='O'||c=='U')

count++;

}

return count;

}

int main() {

char str[] = "Programming";

printf("Vowels = %d\n", countVowels(str));

return 0;

}

**118.Function to print multiplication table of number**

#include <stdio.h>

void printTable(int n) {

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

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

}

int main() {

printTable(7);

return 0;

}

**119.Function with default arguments simulation using function overloading in C style (two functions)**

#include <stdio.h>

void display() {

printf("No argument function\n");

}

void displayWithArg(int a) {

printf("Function with argument: %d\n", a);

}

int main() {

display();

displayWithArg(10);

return 0;

}

**120.Function returning pointer to array element**

#include <stdio.h>

int\* findMax(int arr[], int n) {

int \*max = &arr[0];

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

if(arr[i] > \*max) max = &arr[i];

return max;

}

int main() {

int arr[] = {5, 9, 2, 7};

int \*maxPtr = findMax(arr, 4);

printf("Max element = %d\n", \*maxPtr);

return 0;

}

**121.Pointer basics: Print address and value of a variable**

#include <stdio.h>

int main() {

int a = 10;

int \*p = &a;

printf("Address of a: %p\n", p);

printf("Value of a: %d\n", \*p);

return 0;

}

**122.Use pointer to change value of variable**

#include <stdio.h>

int main() {

int a = 20;

int \*p = &a;

\*p = 30;

printf("Updated value of a = %d\n", a);

return 0;

}

**123.Pointer arithmetic: Access array elements using pointers**

#include <stdio.h>

int main() {

int arr[] = {10, 20, 30, 40, 50};

int \*p = arr;

for(int i = 0; i < 5; i++) {

printf("%d ", \*(p + i));

}

printf("\n");

return 0;

}

**124.Function to swap two numbers using pointers**

#include <stdio.h>

void swap(int \*x, int \*y) {

int temp = \*x;

\*x = \*y;

\*y = temp;

}

int main() {

int a = 10, b = 20;

swap(&a, &b);

printf("a = %d, b = %d\n", a, b);

return 0;

}

**125.Pointer to pointer usage**

#include <stdio.h>

int main() {

int a = 5;

int \*p = &a;

int \*\*pp = &p;

printf("Value of a = %d\n", \*\*pp);

return 0;

}

**126.Dynamic memory allocation using malloc**

#include <stdio.h>

#include <stdlib.h>

int main() {

int \*ptr;

ptr = (int \*)malloc(5 \* sizeof(int));

if (ptr == NULL) {

printf("Memory not allocated.\n");

return 1;

}

for(int i=0; i<5; i++)

ptr[i] = i + 1;

for(int i=0; i<5; i++)

printf("%d ", ptr[i]);

printf("\n");

free(ptr);

return 0;

}

**127.Pointer and array: Traverse and modify elements**

#include <stdio.h>

int main() {

int arr[] = {2, 4, 6, 8, 10};

int \*p = arr;

for(int i=0; i<5; i++) {

\*(p + i) += 10;

printf("%d ", arr[i]);

}

printf("\n");

return 0;

}

**128.Pointer to a function: Call function through pointer**

#include <stdio.h>

void greet() {

printf("Hello from function pointer!\n");

}

int main() {

void (\*funcPtr)() = greet;

funcPtr();

return 0;

}

**129.Passing array to function using pointer**

#include <stdio.h>

void printArray(int \*arr, int n) {

for(int i=0; i<n; i++)

printf("%d ", arr[i]);

printf("\n");

}

int main() {

int arr[] = {1, 2, 3};

printArray(arr, 3);

return 0;

}

**130.Pointer to structure and accessing members**

#include <stdio.h>

struct Point {

int x, y;

};

int main() {

struct Point p = {10, 20};

struct Point \*ptr = &p;

printf("x = %d, y = %d\n", ptr->x, ptr->y);

return 0;

}

**131.String traversal using pointer**

#include <stdio.h>

int main() {

char str[] = "Pointer";

char \*p = str;

while(\*p != '\0') {

printf("%c ", \*p);

p++;

}

printf("\n");

return 0;

}

**132.Pointer comparison**

#include <stdio.h>

int main() {

int a = 5, b = 10;

int \*p = &a;

int \*q = &b;

if(p > q)

printf("Pointer p points to higher address\n");

else

printf("Pointer q points to higher address\n");

return 0;

}

**133.Pointer and multi-dimensional array**

#include <stdio.h>

int main() {

int arr[2][3] = {{1,2,3}, {4,5,6}};

int (\*p)[3] = arr;

for(int i=0; i<2; i++) {

for(int j=0; j<3; j++)

printf("%d ", \*(\*(p + i) + j));

printf("\n");

}

return 0;

}

**134.Void pointer usage**

#include <stdio.h>

int main() {

int a = 10;

float b = 5.5;

void \*ptr;

ptr = &a;

printf("Integer value = %d\n", \*(int\*)ptr);

ptr = &b;

printf("Float value = %.1f\n", \*(float\*)ptr);

return 0;

}

**135.Pointer to constant integer**

#include <stdio.h>

int main() {

const int a = 10;

const int \*p = &a;

printf("Value = %d\n", \*p);

// \*p = 20; // Not allowed

return 0;

}

**136.Function returning pointer**

#include <stdio.h>

int\* getMax(int \*a, int \*b) {

return (\*a > \*b) ? a : b;

}

int main() {

int x = 10, y = 20;

int \*max = getMax(&x, &y);

printf("Max = %d\n", \*max);

return 0;

}

**137.Pointer to union**

#include <stdio.h>

union Data {

int i;

float f;

char str[20];

};

int main() {

union Data data;

union Data \*ptr = &data;

ptr->i = 10;

printf("data.i = %d\n", data.i);

return 0;

}

**138.Swap two strings using pointers**

#include <stdio.h>

void swap(char \*\*a, char \*\*b) {

char \*temp = \*a;

\*a = \*b;

\*b = temp;

}

int main() {

char \*str1 = "Hello";

char \*str2 = "World";

swap(&str1, &str2);

printf("str1 = %s, str2 = %s\n", str1, str2);

return 0;

}

**139.Pointer with const qualifier**

#include <stdio.h>

int main() {

int a = 5;

int \* const p = &a; // Pointer is constant

\*p = 10;

printf("%d\n", a);

// p = NULL; // Not allowed

return 0;

}

**140.Pointer to an array and modify array elements**

#include <stdio.h>

int main() {

int arr[] = {1, 2, 3, 4};

int (\*p)[4] = &arr;

(\*p)[0] = 10;

(\*p)[1] = 20;

for(int i = 0; i < 4; i++)

printf("%d ", arr[i]);

printf("\n");

return 0;

}  
**141.Pointer to structure: Access and modify structure members**

#include <stdio.h>

struct Point {

int x, y;

};

int main() {

struct Point p = {10, 20};

struct Point \*ptr = &p;

printf("x = %d, y = %d\n", ptr->x, ptr->y);

ptr->x = 30;

printf("Updated x = %d\n", p.x);

return 0;

}

**142.Function to swap two structures using pointers**

#include <stdio.h>

struct Point {

int x, y;

};

void swap(struct Point \*p1, struct Point \*p2) {

struct Point temp = \*p1;

\*p1 = \*p2;

\*p2 = temp;

}

int main() {

struct Point p1 = {1, 2}, p2 = {3, 4};

swap(&p1, &p2);

printf("p1: %d %d\np2: %d %d\n", p1.x, p1.y, p2.x, p2.y);

return 0;

}

**143.Array of structures: Input and output**

#include <stdio.h>

struct Student {

int id;

char name[50];

};

int main() {

struct Student s[3];

for(int i=0; i<3; i++) {

printf("Enter id and name: ");

scanf("%d %s", &s[i].id, s[i].name);

}

printf("Student details:\n");

for(int i=0; i<3; i++)

printf("Id: %d, Name: %s\n", s[i].id, s[i].name);

return 0;

}

**144.Nested structures**

#include <stdio.h>

struct Date {

int day, month, year;

};

struct Student {

int id;

char name[50];

struct Date dob;

};

int main() {

struct Student s1 = {1, "Alice", {25, 12, 2000}};

printf("Name: %s, DOB: %d/%d/%d\n", s1.name, s1.dob.day, s1.dob.month, s1.dob.year);

return 0;

}

**145.Union example: Store either int or float in same space**

#include <stdio.h>

union Data {

int i;

float f;

};

int main() {

union Data d;

d.i = 10;

printf("Integer: %d\n", d.i);

d.f = 15.5;

printf("Float: %.2f\n", d.f);

return 0;

}

**146.Size of structure vs union**

#include <stdio.h>

struct S {

int i;

float f;

char c;

};

union U {

int i;

float f;

char c;

};

int main() {

printf("Size of struct = %lu\n", sizeof(struct S));

printf("Size of union = %lu\n", sizeof(union U));

return 0;

}

**147.Initialize and print structure using function**

#include <stdio.h>

struct Student {

int id;

char name[50];

};

void printStudent(struct Student s) {

printf("Id: %d, Name: %s\n", s.id, s.name);

}

int main() {

struct Student s1 = {101, "Bob"};

printStudent(s1);

return 0;

}

**148.Pointer to union and accessing its members**

#include <stdio.h>

union Data {

int i;

float f;

};

int main() {

union Data d;

union Data \*ptr = &d;

ptr->i = 42;

printf("Value of i: %d\n", ptr->i);

ptr->f = 3.14;

printf("Value of f: %.2f\n", ptr->f);

return 0;

}

**149.Using typedef for structures**

#include <stdio.h>

typedef struct {

int x, y;

} Point;

int main() {

Point p = {10, 20};

printf("Point: (%d, %d)\n", p.x, p.y);

return 0;

}

**150.Passing structure by pointer to function**

#include <stdio.h>

struct Rectangle {

int length, width;

};

void area(struct Rectangle \*r) {

printf("Area = %d\n", r->length \* r->width);

}

int main() {

struct Rectangle rect = {10, 5};

area(&rect);

return 0;

}

**151.Array of pointers to structures**

#include <stdio.h>

struct Student {

int id;

char name[50];

};

int main() {

struct Student s1 = {1, "A"}, s2 = {2, "B"};

struct Student \*arr[] = {&s1, &s2};

for (int i = 0; i < 2; i++)

printf("Id: %d, Name: %s\n", arr[i]->id, arr[i]->name);

return 0;

}

**152.Dynamic memory allocation for structures**

#include <stdio.h>

#include <stdlib.h>

struct Student {

int id;

char name[50];

};

int main() {

struct Student \*s = (struct Student \*)malloc(sizeof(struct Student));

s->id = 1;

strcpy(s->name, "John");

printf("Id: %d, Name: %s\n", s->id, s->name);

free(s);

return 0;

}

**153.Compare two structures**

#include <stdio.h>

#include <string.h>

struct Student {

int id;

char name[50];

};

int compare(struct Student s1, struct Student s2) {

if (s1.id == s2.id && strcmp(s1.name, s2.name) == 0)

return 1;

return 0;

}

int main() {

struct Student s1 = {1, "Ann"}, s2 = {1, "Ann"};

printf("Are same: %d\n", compare(s1, s2));

return 0;

}

**154.Nested union inside structure**

#include <stdio.h>

struct Data {

int type;

union {

int i;

float f;

} value;

};

int main() {

struct Data d;

d.type = 1;

d.value.i = 100;

printf("Integer: %d\n", d.value.i);

d.type = 2;

d.value.f = 98.7;

printf("Float: %.2f\n", d.value.f);

return 0;

}

**155.Bit-fields in structures**

#include <stdio.h>

struct Flags {

unsigned int is\_active : 1;

unsigned int has\_error : 1;

};

int main() {

struct Flags f = {1, 0};

printf("Active: %d, Error: %d\n", f.is\_active, f.has\_error);

return 0;

}

**156.Using flexible array member in structure**

#include <stdio.h>

#include <stdlib.h>

struct Flex {

int size;

int arr[];

};

int main() {

int n = 5;

struct Flex \*f = malloc(sizeof(struct Flex) + n \* sizeof(int));

f->size = n;

for (int i = 0; i < n; i++)

f->arr[i] = i \* i;

for (int i = 0; i < n; i++)

printf("%d ", f->arr[i]);

printf("\n");

free(f);

return 0;

}

**157.Structure padding demonstration**

#include <stdio.h>

struct Padding {

char a;

int b;

char c;

};

int main() {

printf("Size of struct: %lu\n", sizeof(struct Padding));

return 0;

}

**158.Anonymous unions**

#include <stdio.h>

struct S {

int type;

union {

int i;

float f;

};

};

int main() {

struct S s;

s.type = 1; s.i = 10;

printf("i = %d\n", s.i);

s.type = 2; s.f = 3.14;

printf("f = %.2f\n", s.f);

return 0;

}

**159.Function returning structure**

#include <stdio.h>

struct Point {

int x, y;

};

struct Point makePoint(int a, int b) {

struct Point p = {a, b};

return p;

}

int main() {

struct Point p = makePoint(5, 10);

printf("Point: (%d, %d)\n", p.x, p.y);

return 0;

}

**160.Self-referential structure (linked list node)**

#include <stdio.h>

struct Node {

int data;

struct Node \*next;

};

int main() {

struct Node n1, n2;

n1.data = 1; n1.next = &n2;

n2.data = 2; n2.next = NULL;

printf("Node1: %d, Node2: %d\n", n1.data, n1.next->data);

return 0;

}

**161.Create a file and write text to it**

#include <stdio.h>

int main() {

FILE \*fp = fopen("file.txt", "w");

if(fp == NULL) {

printf("Error opening file.\n");

return 1;

}

fprintf(fp, "Hello, File Handling!\n");

fclose(fp);

return 0;

}

**162.Open a file and read its contents**

#include <stdio.h>

int main() {

FILE \*fp = fopen("file.txt", "r");

char c;

if(fp == NULL) {

printf("Error opening file.\n");

return 1;

}

while((c = fgetc(fp)) != EOF) {

putchar(c);

}

fclose(fp);

return 0;

}

**163.Append text to a file**

#include <stdio.h>

int main() {

FILE \*fp = fopen("file.txt", "a");

if(fp == NULL) {

printf("Error opening file.\n");

return 1;

}

fprintf(fp, "Appending this line.\n");

fclose(fp);

return 0;

}

**164.Copy contents of one file to another**

#include <stdio.h>

int main() {

FILE \*src = fopen("file.txt", "r");

FILE \*dest = fopen("copy.txt", "w");

char c;

if(src == NULL || dest == NULL) {

printf("Error opening files.\n");

return 1;

}

while((c = fgetc(src)) != EOF) {

fputc(c, dest);

}

fclose(src);

fclose(dest);

return 0;

}

**165.Count number of characters in a file**

#include <stdio.h>

int main() {

FILE \*fp = fopen("file.txt", "r");

int count = 0;

char c;

if(fp == NULL) {

printf("Error opening file.\n");

return 1;

}

while((c = fgetc(fp)) != EOF) {

count++;

}

printf("Character count: %d\n", count);

fclose(fp);

return 0;

}

**166.number of words in a file**

#include <stdio.h>

#include <ctype.h>

int main() {

FILE \*fp = fopen("file.txt", "r");

int words = 0;

char c, prev = ' ';

if(fp == NULL) {

printf("Error opening file.\n");

return 1;

}

while((c = fgetc(fp)) != EOF) {

if(isspace(c) && !isspace(prev))

words++;

prev = c;

}

if(!isspace(prev)) words++; // count last word

printf("Word count: %d\n", words);

fclose(fp);

return 0;

}

**167.Count number of lines in a file**

#include <stdio.h>

int main() {

FILE \*fp=fopen("file.txt", "r");

int lines=0;

char c;

if(fp==NULL) {

printf("Error opening file.\n");

return 1;

}

while((c=fgetc(fp)) != EOF) {

if(c=='\n') lines++;

}

printf("Line count: %d\n", lines);

fclose(fp);

return 0;

}  
**168.Write array of integers to a binary file**

#include <stdio.h>

int main() {

FILE \*fp = fopen("ints.bin", "wb");

int arr[] = {1,2,3,4,5};

if(fp == NULL) {

printf("Error opening file.\n");

return 1;

}

fwrite(arr, sizeof(int), 5, fp);

fclose(fp);

return 0;

} **169.Read array of integers from binary file**

#include <stdio.h>

int main() {

FILE \*fp = fopen("ints.bin", "rb");

int arr[5];

if(fp == NULL) {

printf("Error opening file.\n");

return 1;

}

fread(arr, sizeof(int), 5, fp);

for(int i=0; i<5; i++)

printf("%d ", arr[i]);

printf("\n");

fclose(fp);

return 0;

}

**170.Modify a specific line in a text file**

#include <stdio.h>

#include <string.h>

int main() {

FILE \*fp = fopen("file.txt", "r+");

char line[100];

int line\_no = 3, curr\_line = 1;

if(fp==NULL) return 1;

while(fgets(line, 100, fp) != NULL) {

if(curr\_line == line\_no) {

fseek(fp, -strlen(line), SEEK\_CUR);

fputs("Modified line\n", fp);

break;

}

curr\_line++;

}

fclose(fp);

return 0;

}

**171.Copy first N characters from one file to another**

#include <stdio.h>

int main() {

FILE \*src = fopen("file.txt", "r");

FILE \*dest = fopen("copy.txt", "w");

char c;

int n = 10, count = 0;

if(src == NULL || dest == NULL) return 1;

while(count < n && (c = fgetc(src)) != EOF) {

fputc(c, dest);

count++;

}

fclose(src); fclose(dest);

return 0;

}

**172.Write user input to a file until "exit" is entered**

#include <stdio.h>

#include <string.h>

int main() {

FILE \*fp = fopen("input.txt", "w");

char str[100];

if(fp == NULL) return 1;

while(1) {

printf("Enter text (exit to quit): ");

fgets(str, 100, stdin);

if(strncmp(str, "exit", 4) == 0) break;

fputs(str, fp);

}

fclose(fp);

return 0;

}

**173.File pointer functions: Use fseek and ftell**

#include <stdio.h>

int main() {

FILE \*fp = fopen("file.txt", "r");

if(fp == NULL) return 1;

fseek(fp, 0, SEEK\_END);

long size = ftell(fp);

printf("File size: %ld bytes\n", size);

fclose(fp);

return 0;

}

**174.Compare two files and display first mismatch**

#include <stdio.h>

int main() {

FILE \*fp1 = fopen("file1.txt", "r");

FILE \*fp2 = fopen("file2.txt", "r");

char c1, c2;

int pos = 0;

if(fp1 == NULL || fp2 == NULL) return 1;

while(1) {

c1 = fgetc(fp1);

c2 = fgetc(fp2);

pos++;

if(c1 != c2) break;

if(c1 == EOF || c2 == EOF) break;

}

if(c1 == c2)

printf("Files are identical\n");

else

printf("Mismatch at position %d\n", pos);

fclose(fp1); fclose(fp2);

return 0;

}

**175.Delete a file using remove()**

#include <stdio.h>

int main() {

if(remove("delete.txt") == 0)

printf("File deleted successfully.\n");

else

printf("Unable to delete the file.\n");

return 0;

}

**176.Rename a file using rename()**

#include <stdio.h>

int main() {

if(rename("oldname.txt", "newname.txt") == 0)

printf("File renamed successfully.\n");

else

printf("Error renaming file.\n");

return 0;

}

**177.Read integers from file and calculate average**

#include <stdio.h>

int main() {

FILE \*fp = fopen("ints.txt", "r");

int num, sum = 0, count = 0;

if(fp == NULL) return 1;

while(fscanf(fp, "%d", &num) == 1) {

sum += num;

count++;

}

if(count > 0)

printf("Average = %.2f\n", (float)sum/count);

else

printf("No numbers found\n");

fclose(fp);

return 0;

}

**178.Write formatted data to file using fprintf**

#include <stdio.h>

int main() {

FILE \*fp = fopen("data.txt", "w");

if(fp == NULL) return 1;

fprintf(fp, "Name: %s\nAge: %d\n", "Alice", 30);

fclose(fp);

return 0;

}

**179.Read formatted data from file using fscanf**

#include <stdio.h>

int main() {

char name[50];

int age;

FILE \*fp = fopen("data.txt", "r");

if(fp == NULL) return 1;

fscanf(fp, "Name: %s\nAge: %d\n", name, &age);

printf("Name: %s, Age: %d\n", name, age);

fclose(fp);

return 0;

}

**180.Check if file exists**

#include <stdio.h>

int main() {

FILE \*fp = fopen("file.txt", "r");

if(fp != NULL) {

printf("File exists\n");

fclose(fp);

} else {

printf("File does not exist\n");

}

return 0;

}

**181.Simple macro to find maximum of two numbers**

#include <stdio.h>

#define MAX(a,b) ((a) > (b) ? (a) : (b))

int main() {

int x = 10, y = 20;

printf("Max = %d\n", MAX(x, y));

return 0;

}

**182.Macro to calculate square of a number**

#include <stdio.h>

#define SQR(x) ((x) \* (x))

int main() {

int num = 5;

printf("Square = %d\n", SQR(num));

return 0;

}

**183.Using #ifdef and #endif for conditional compilation**

#include <stdio.h>

#define DEBUG

int main() {

#ifdef DEBUG

printf("Debug mode is ON\n");

#endif

printf("Program running\n");

return 0;

}

**184.Using #ifndef for checking if macro is not defined**

#include <stdio.h>

#ifndef VERSION

#define VERSION 1

#endif

int main() {

printf("Version: %d\n", VERSION);

return 0;

}

**185.Define constant value using #define**

#include <stdio.h>

#define PI 3.14159

int main() {

double r = 5.0, area;

area = PI \* r \* r;

printf("Area of circle = %.2f\n", area);

return 0;

}

**186.Multi-line macro example**

#include <stdio.h>

#define PRINT\_VALUES(a,b) { \

printf("Value a = %d\n", a); \

printf("Value b = %d\n", b); \

}

int main() {

int x = 10, y = 20;

PRINT\_VALUES(x, y);

return 0;

}

**187.Undefining a macro using #undef**

#include <stdio.h>

#define TEMP 100

#undef TEMP

int main() {

#ifdef TEMP

printf("TEMP value: %d\n", TEMP);

#else

printf("TEMP is undefined\n");

#endif

return 0;

}

**188.Macro with arguments to calculate minimum of two numbers**

#include <stdio.h>

#define MIN(a,b) ((a) < (b) ? (a) : (b))

int main() {

int x = 10, y = 20;

printf("Min = %d\n", MIN(x, y));

return 0;

}

**189.Stringizing operator (#) in macros**

#include <stdio.h>

#define TO\_STRING(x) #x

int main() {

printf("%s\n", TO\_STRING(Hello World));

return 0;

}

**190.Token-pasting operator (##) in macros**

#include <stdio.h>

#define CONCAT(a,b) a##b

int main() {

int xy = 100;

printf("%d\n", CONCAT(x, y));

return 0;

}

**191.compile-time error using #error directive**

**#**include <stdio.h>

#ifndef VERSION

#error "VERSION is not defined!"

#endif

int main() {

printf("Version is defined.\n");

return 0;

}

**192.Include another file with #include directive**

// Create file named "header.h" with content

// #define MESSAGE "Included File"

#include <stdio.h>

#include "header.h"

int main() {

printf("%s\n", MESSAGE);

return 0;

}

**193.Using FILE and LINE predefined macros**

#include <stdio.h>

int main() {

printf("File: %s, Line: %d\n", \_\_FILE\_\_, \_\_LINE\_\_);

return 0;

}

**194.Using DATE and TIME predefined macros**

#include <stdio.h>

int main() {

printf("Compilation date: %s\n", \_\_DATE\_\_);

printf("Compilation time: %s\n", \_\_TIME\_\_);

return 0;

}

**195.Calculate area of rectangle using macro**

#include <stdio.h>

#define AREA(length, width) ((length) \* (width))

int main() {

printf("Area = %d\n", AREA(5, 10));

return 0;

}

**196.Macro with conditional operator**

#include <stdio.h>

#define ABS(x) ((x) < 0 ? -(x) : (x))

int main() {

printf("Absolute value of -10 = %d\n", ABS(-10));

return 0;

}

**197.Nested macros example**

#include <stdio.h>

#define MUL(x,y) ((x)\*(y))

#define SQUARE(x) MUL(x,x)

int main() {

printf("Square of 4 = %d\n", SQUARE(4));

return 0;

}

**198.Using #pragma directive (compiler-specific)**

#include <stdio.h>

#pragma message ("Compiling Chapter 11 programs")

int main() {

printf("Hello, pragma example!\n");

return 0;

}

**199.Macro to swap two variables using XOR**

#include <stdio.h>

#define SWAP(a,b) do { a ^= b; b ^= a; a ^= b; } while (0)

int main() {

int x = 5, y = 10;

SWAP(x, y);

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

return 0;

}

**200.Macro with variable arguments**

#include <stdio.h>

#define DEBUG(fmt, ...) printf(fmt, \_\_VA\_ARGS\_\_)

int main() {

DEBUG("Error code: %d, message: %s\n", 404, "Not Found");

return 0;

}