1) **Hello World**

#include <stdio.h>

int main(void) {

printf("Hello World!\n");

return 0;

}

**Output:**  
Hello World!

2) **Subtraction of Two Numbers**

#include <stdio.h>

int main(void) {

int a, b, sub;

printf("Enter two numbers: ");

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

sub = a - b;

printf("Subtraction is = %d\n", sub);

return 0;

}

**Output:**  
Enter two numbers: 10 5  
Subtraction is = 5

3) **Sum and Average of Two Numbers**

#include <stdio.h>

int main(void) {

int a, b, sum;

float avg;

printf("Enter two numbers: ");

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

sum = a + b;

avg = sum / 2.0f;

printf("Sum = %d\nAverage = %.2f\n", sum, avg);

return 0;

}

**Output:**  
Enter two numbers: 8 12  
Sum = 20  
Average = 10.00

4) **ASCII Value of a Character**

#include <stdio.h>

int main(void) {

char ch;

printf("Enter the character: ");

scanf(" %c", &ch);

printf("ASCII is = %d\n", (int)ch);

return 0;

}

**Output:**  
Enter the character: A  
ASCII is = 65

5) **Cube of an Integer**

#include <stdio.h>

#include <math.h>

int main(void) {

int n;

printf("Enter a number: ");

scanf("%d", &n);

printf("Cube (without pow) = %d\n", n \* n \* n);

printf("Cube (using pow) = %.0f\n", pow((double)n, 3.0));

return 0;

}

**Output:**  
Enter a number: 3  
Cube (without pow) = 27  
Cube (using pow) = 27

6) **Quotient and Remainder**

#include <stdio.h>

int main(void) {

int dividend, divisor, quotient, remainder;

printf("Enter dividend and divisor: ");

scanf("%d %d", &dividend, &divisor);

quotient = dividend / divisor;

remainder = dividend % divisor;

printf("Quotient = %d\nRemainder = %d\n", quotient, remainder);

return 0;

}

**Output:**  
Enter dividend and divisor: 20 3  
Quotient = 6  
Remainder = 2

7) **Simple Interest**

#include <stdio.h>

int main(void) {

float principal, rate, time, si;

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

scanf("%f %f %f", &principal, &rate, &time);

si = (principal \* rate \* time) / 100.0f;

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

return 0;

}

**Output:**  
Enter principal, rate, time: 1000 5 3  
Simple Interest = 150.00

8) **EVEN or ODD**

#include <stdio.h>

int main(void) {

int num;

printf("Enter an integer number: ");

scanf("%d", &num);

if (num % 2 == 0)

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

else

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

return 0;

}

**Output:**  
Enter an integer number: 7  
7 is an ODD number.

9) **Largest Among Three Numbers**

#include <stdio.h>

int main(void) {

int a, b, c, largest;

printf("Enter three numbers: ");

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

largest = a;

if (b > largest) largest = b;

if (c > largest) largest = c;

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

return 0;

}

**Output:**  
Enter three numbers: 3 7 5  
Largest number = 7

10) **Voting Eligibility**

#include <stdio.h>

int main(void) {

int age;

printf("Enter age: ");

scanf("%d", &age);

if (age >= 18)

printf("Eligible for voting\n");

else

printf("Not eligible\n");

return 0;

}

**Output:**  
Enter age: 16  
Not eligible

11) **Total, Percentage, Division**

#include <stdio.h>

int main(void) {

int m1, m2, m3, total;

float per;

printf("Enter marks: ");

scanf("%d %d %d", &m1, &m2, &m3);

total = m1 + m2 + m3;

per = total / 3.0f;

printf("Total = %d, Percentage = %.2f\n", total, per);

if (per >= 60) printf("First Division\n");

else if (per >= 50) printf("Second Division\n");

else if (per >= 40) printf("Third Division\n");

else printf("Fail\n");

return 0;

}

**Output:**  
Enter marks: 66 70 72  
Total = 208, Percentage = 69.33  
First Division

12) **Gross Salary**

#include <stdio.h>

int main(void) {

float basic, hra, da, pf, gross;

printf("Enter Basic: ");

scanf("%f", &basic);

hra = 0.10f \* basic;

da = 0.12f \* basic;

pf = 0.08f \* basic;

gross = basic + hra + da - pf;

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

return 0;

}

**Output:**  
Enter Basic: 25000  
Gross = 27000.00

13) **Fahrenheit/Celsius Conversion**

#include <stdio.h>

int main(void) {

float temp, c, f;

int choice;

printf("Temp: ");

scanf("%f", &temp);

printf("1.Celsius 2.Fahrenheit: ");

scanf("%d", &choice);

if (choice == 1) {

c = (temp - 32.0f) \* 5.0f / 9.0f;

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

} else if (choice == 2) {

f = temp \* 9.0f / 5.0f + 32.0f;

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

} else {

printf("Invalid\n");

}

return 0;

}

**Output:**  
Temp: 98.6  
1.Celsius 2.Fahrenheit: 1  
Celsius = 37.00

14) **X^N using pow**

#include <stdio.h>

#include <math.h>

int main(void) {

int x, n;

double res;

printf("X N: ");

scanf("%d %d", &x, &n);

res = pow((double)x, (double)n);

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

return 0;

}

**Output:**  
X N: 2 5  
32.00

15) **Absolute Difference**

#include <stdio.h>

int main(void) {

int a, b, d;

printf("Enter two integers: ");

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

d = (a > b) ? (a - b) : (b - a);

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

return 0;

}

**Output:**  
Enter two integers: 10 15

16) **Size of Variables**

#include <stdio.h>

int main(void) {

int a;

float b;

double c;

char d;

printf("%zu %zu %zu %zu\n", sizeof(a), sizeof(b), sizeof(c), sizeof(d));

return 0;

}

**Output:**  
4 4 8 1

17) **Escape Sequences**

#include <stdio.h>

int main(void) {

printf("Hello\\nWorld\\nTab\\tdemo\\n\\\"\\n");

return 0;

}

**Output:**  
Hello\nWorld\nTab\tdemo\n"

18) **Circle Area & Perimeter**

#include <stdio.h>

#define PI 3.1416f

int main(void) {

float r, a, p;

printf("Enter radius: ");

scanf("%f", &r);

a = PI \* r \* r;

p = 2 \* PI \* r;

printf("%.2f %.2f\n", a, p);

return 0;

}

**Output:**  
Enter radius: 7  
153.94 43.98

19) Rectangle Area

#include <stdio.h>

int main(void) {

float l, b, a;

printf("Enter length and breadth: ");

scanf("%f %f", &l, &b);

a = l \* b;

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

return 0;

}

Output:  
Enter length and breadth: 5 8  
40.00

20) **HCF of Two Numbers**

#include <stdio.h>

int main(void) {

int a, b, h = 1;

printf("Enter two integers: ");

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

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

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

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

return 0;

}

**Output:**  
Enter two integers: 36 60  
12

21) **LCM of Two Numbers**

#include <stdio.h>

int main(void) {

int a, b, max;

printf("Enter two integers: ");

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

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

while (1) {

if (max % a == 0 && max % b == 0) {

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

break;

}

max++;

}

return 0;

}

**Output:**  
Enter two integers: 4 6  
12

22) **Triangle Area (Three Sides, Heron)**

#include <stdio.h>

#include <math.h>

int main(void) {

double a, b, c, s, area;

printf("Enter three sides: ");

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

s = (a + b + c) / 2.0;

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

printf("%.2lf\n", area);

return 0;

}

**Output:**  
Enter three sides: 5 6 7  
14.70

23) Triangle Area (Base & Height)

#include <stdio.h>

int main(void) {

float b, h, a;

printf("Enter base and height: ");

scanf("%f %f", &b, &h);

a = 0.5f \* b \* h;

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

return 0;

}

Output:  
Enter base and height: 8 4  
16.00

24) **Trapezium Area**

#include <stdio.h>

int main(void) {

float b1, b2, h, a;

printf("Enter b1 b2 h: ");

scanf("%f %f %f", &b1, &b2, &h);

a = 0.5f \* (b1 + b2) \* h;

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

return 0;

}

**Output:**  
Enter b1 b2 h: 6 8 5  
35.00

25) **Area of Rhombus**

#include <stdio.h>

int main(void) {

float d1, d2, area;

printf("Enter diagonal1 and diagonal2: ");

scanf("%f %f", &d1, &d2);

area = 0.5f \* d1 \* d2;

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

return 0;

}

**Output:**  
Enter diagonal1 and diagonal2: 7 4  
Area = 14.00

26) **Area of Parallelogram**

#include <stdio.h>

int main(void) {

float base, height, area;

printf("Enter base and height: ");

scanf("%f %f", &base, &height);

area = base \* height;

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

return 0;

}

**Output:**  
Enter base and height: 10 3  
Area = 30.00

27) **Surface Area & Volume of Cylinder**

#include <stdio.h>

#include <math.h>

#define PI 3.1416f

int main(void) {

float r, h, sa, vol;

printf("Enter radius and height: ");

scanf("%f %f", &r, &h);

sa = 2.0f \* PI \* r \* (r + h);

vol = PI \* r \* r \* h;

printf("Surface Area = %.2f\nVolume = %.2f\n", sa, vol);

return 0;

}

**Output:**  
Enter radius and height: 3 7  
Surface Area = 188.40  
Volume = 197.82

28) **Surface Area, Volume & Space Diagonal of Cuboid**

#include <stdio.h>

#include <math.h>

int main(void) {

float l, w, h, sa, vol, diag;

printf("Enter length, width, height: ");

scanf("%f %f %f", &l, &w, &h);

sa = 2.0f \* (l \* w + w \* h + h \* l);

vol = l \* w \* h;

diag = sqrtf(l \* l + w \* w + h \* h);

printf("Surface Area = %.2f\nVolume = %.2f\nSpace diagonal = %.2f\n", sa, vol, diag);

return 0;

}

**Output:**  
Enter length, width, height: 4 5 6  
Surface Area = 148.00  
Volume = 120.00  
Space diagonal = 8.77

29) **Surface Area & Volume of Cone**

#include <stdio.h>

#include <math.h>

#define PI 3.1416f

int main(void) {

float r, h, slant, sa, vol;

printf("Enter radius and height: ");

scanf("%f %f", &r, &h);

slant = sqrtf(h \* h + r \* r);

sa = PI \* r \* (r + slant);

vol = (PI \* r \* r \* h) / 3.0f;

printf("Surface Area = %.2f\nVolume = %.2f\n", sa, vol);

return 0;

}

**Output:**  
Enter radius and height: 3 4  
Surface Area = 75.40  
Volume = 37.70

30) **Surface Area & Volume of Sphere**

#include <stdio.h>

#define PI 3.1416f

int main(void) {

float r, sa, vol;

printf("Enter radius: ");

scanf("%f", &r);

sa = 4.0f \* PI \* r \* r;

vol = (4.0f / 3.0f) \* PI \* r \* r \* r;

printf("Surface Area = %.2f\nVolume = %.2f\n", sa, vol);

return 0;

}

**Output:**  
Enter radius: 5  
Surface Area = 314.16  
Volume = 523.60

**31) C Program to Read Weekday Number and Print Weekday Name Using Switch**

#include <stdio.h>

int main(void) {

int day;

printf("Enter weekday 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 weekday number\n");

}

return 0;

}

**Output :**

Enter weekday number (1-7): 5

Thursday

**32) C Program to Read Gender (M/F) and Print Corresponding Gender Using Switch**

#include <stdio.h>

int main(void) {

char g;

printf("Enter gender (M/F): ");

scanf(" %c", &g);

switch (g) {

case 'M':

case 'm':

printf("Male\n");

break;

case 'F':

case 'f':

printf("Female\n");

break;

default:

printf("Invalid input (use M or F)\n");

}

return 0;

}

**Output :**

Enter gender (M/F): F

Female

**33) C Program to Check Whether a Character is Vowel or Consonant Using Switch**

#include <stdio.h>

int main(void) {

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:

printf("Consonant\n");

}

return 0;

}

**Output :**

Enter a character: e

Vowel

**34) C Program to Design Calculator with Basic Operations Using Switch**

#include <stdio.h>

int main(void) {

int a, b;

char op;

printf("Enter two numbers: ");

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

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

scanf(" %c", &op);

switch (op) {

case '+': printf("Result = %d\n", a + b); break;

case '-': printf("Result = %d\n", a - b); break;

case '\*': printf("Result = %d\n", a \* b); break;

case '/':

if (b != 0) printf("Result = %d\n", a / b);

else printf("Division by zero not allowed\n");

break;

case '%':

if (b != 0) printf("Result = %d\n", a % b);

else printf("Modulo by zero not allowed\n");

break;

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

}

return 0;

}

**Output**

Enter two numbers: 20 5

Enter operator (+, -, \*, /, %): /

Result = 4

**35) C Program to Check Whether Number is Even or Odd Using Switch**

#include <stdio.h>

int main(void) {

int num;

printf("Enter a number: ");

scanf("%d", &num);

switch (num % 2) {

case 0: printf("Even\n"); break;

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

}

return 0;

}

**Output :**

Enter a number: 9

Odd

**36) C Program to Find Number of Days in a Month Using Switch**

#include <stdio.h>

int main(void) {

int month;

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

scanf("%d", &month);

switch (month) {

case 1: case 3: case 5: case 7: case 8: case 10: case 12:

printf("31 days\n");

break;

case 4: case 6: case 9: case 11:

printf("30 days\n");

break;

case 2:

printf("28 or 29 days (depends on leap year)\n");

break;

default:

printf("Invalid month number\n");

}

return 0;

}

**Output**

Enter month number (1-12): 2

28 or 29 days (depends on leap year)

**37: C Program to Find Binary Number of a Decimal Number**

#include <stdio.h>

int main() {

int n;

printf("Enter a decimal number: ");

scanf("%d", &n);

printf("Binary: ");

for (int i = 31; i >= 0; i--) {

printf("%d", (n >> i) & 1);

}

printf("\n");

return 0;

}

**Output**

Enter a decimal number: 10

Binary: 00000000000000000000000000001010

**38: C Program to Get Minimum Number of Bits to Store an Integer**

#include <stdio.h>

int main() {

int num, bits = 0;

printf("Enter a number: ");

scanf("%d", &num);

int temp = num;

while (temp) {

bits++;

temp >>= 1;

}

printf("Minimum bits required: %d\n", bits);

return 0;

}

**Output**

Enter a number: 8

Minimum bits required: 4

**39: C Program to Swap Two Bits**

#include <stdio.h>

int main() {

int num, p1, p2;

printf("Enter number: ");

scanf("%d", &num);

printf("Enter positions to swap: ");

scanf("%d %d", &p1, &p2);

int bit1 = (num >> p1) & 1;

int bit2 = (num >> p2) & 1;

if (bit1 != bit2)

num ^= (1U << p1) | (1U << p2);

printf("Number after swapping bits: %d\n", num);

return 0;

}

**Output 39**

Enter number: 43

Enter positions to swap: 1 5

Number after swapping bits: 11

**40: C Program to Check Whether All Bits Are UNSET**

#include <stdio.h>

int main() {

unsigned char num;

printf("Enter a number (0-255): ");

scanf("%hhu", &num);

if ((num & 0xFF) == 0)

printf("All bits are UNSET\n");

else

printf("All bits are not UNSET\n");

return 0;

}

**Output**

Enter a number (0-255): 0

All bits are UNSET

**41: C Program to Swap Bytes (e.g. 0x1234 → 0x3412)**

#include <stdio.h>

int main() {

unsigned short x;

printf("Enter a 16-bit number (hex): ");

scanf("%hx", &x);

unsigned short swapped = (x >> 8) | (x << 8);

printf("After swapping bytes: 0x%04X\n", swapped);

return 0;

}

**Output**

Enter a 16-bit number (hex): 1234

After swapping bytes: 0x3412

**42: C Program to Reverse Bits of a Number**

#include <stdio.h>

int main() {

unsigned int n, rev = 0;

printf("Enter a number: ");

scanf("%u", &n);

for (int i = 0; i < 32; i++) {

rev <<= 1;

rev |= (n >> i) & 1;

}

printf("Reversed bits number: %u\n", rev);

return 0;

}

**Output**

Enter a number: 5

Reversed bits number: 2684354560

**43: C Program to Count Number of 1’s in a Number**

#include <stdio.h>

int main() {

unsigned int n, count = 0;

printf("Enter a number: ");

scanf("%u", &n);

while (n) {

count += n & 1;

n >>= 1;

}

printf("Total set bits = %u\n", count);

return 0;

}

**Output**

Enter a number: 15

Total set bits = 4

**44: C Program to Swap Nibbles of a Byte**

#include <stdio.h>

int main() {

unsigned char num;

printf("Enter a byte (0-255): ");

scanf("%hhu", &num);

unsigned char swapped = (num >> 4) | (num << 4);

printf("After swapping nibbles: %u\n", swapped);

return 0;

}

**Output**

Enter a byte (0-255): 100

After swapping nibbles: 70

**45: C Program to Demonstrate Left Shift (<<) Operator**

#include <stdio.h>

int main() {

int a = 5;

printf("Before shift: %d\n", a);

a = a << 2;

printf("After left shift by 2: %d\n", a);

return 0;

}

**Output**

Before shift: 5

After left shift by 2: 20

**46: C Program to Demonstrate Right Shift (>>) Operator**

#include <stdio.h>

int main() {

int a = 20;

printf("Before shift: %d\n", a);

a = a >> 2;

printf("After right shift by 2: %d\n", a);

return 0;

}

**Output**

Before shift: 20

After right shift by 2: 5

**47: C Program to Set and Clear Bits**

#include <stdio.h>

int main() {

unsigned int num = 10;

printf("Original: %u\n", num);

num = ~0;

printf("All bits set: %u\n", num);

num = 0;

printf("All bits cleared: %u\n", num);

return 0;

}

**Output**

Original: 10

All bits set: 4294967295

All bits cleared: 0

**48: C Program to Swap Two Numbers Using Bitwise Operator**

#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("After swap: a=%d, b=%d\n", a, b);

return 0;

}

**Output**

Enter two numbers: 5 9

After swap: a=9, b=5

**49: C Program to Count Trailing Zeroes**

#include <stdio.h>

int main() {

int n, count = 0;

printf("Enter an integer: ");

scanf("%d", &n);

while ((n & 1) == 0 && n != 0) {

count++;

n >>= 1;

}

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

return 0;

}

**Output**

Enter an integer: 8

Trailing zeros: 3

**50: C Program to Find Highest Bit Set**

#include <stdio.h>

int main() {

unsigned int n;

printf("Enter a number: ");

scanf("%u", &n);

int pos = -1;

while (n) {

pos++;

n >>= 1;

}

printf("Highest bit position: %d\n", pos);

return 0;

}

**Output**

Enter a number: 16

Highest bit position: 4

**51: C Program to Check if All Bits Are 1**

#include <stdio.h>

int main() {

unsigned int n;

printf("Enter a number: ");

scanf("%u", &n);

if (~n == 0)

printf("All bits are 1\n");

else

printf("Not all bits are 1\n");

return 0;

}

**Output**

Enter a number: 4294967295

All bits are 1

**52: C Program to Count Bits Set to 1**

#include <stdio.h>

int main() {

unsigned int n, count = 0;

printf("Enter a number: ");

scanf("%u", &n);

while (n) {

count += n & 1;

n >>= 1;

}

printf("Bits set to 1: %u\n", count);

return 0;

}

**Output**

Enter a number: 7

Bits set to 1: 3

**53: C Program to Check Palindrome (Bitwise)**

#include <stdio.h>

int main() {

unsigned int n, rev = 0, temp;

printf("Enter a number: ");

scanf("%u", &n);

temp = n;

for (int i = 0; i < 32; i++) {

rev <<= 1;

rev |= (n >> i) & 1;

}

if (temp == rev)

printf("Palindrome (binary)\n");

else

printf("Not palindrome (binary)\n");

return 0;

}

**Output**

Enter a number: 9

Palindrome (binary)

**54: C Program to Find Odd or Even Using Bitmask**

#include <stdio.h>

int main() {

int n;

printf("Enter a number: ");

scanf("%d", &n);

if (n & 1)

printf("Odd\n");

else

printf("Even\n");

return 0;

}

**Output**

Enter a number: 7

Odd

**55: C Program to Replace Bit from Another Integer**

#include <stdio.h>

int main() {

int num1, num2, pos;

printf("Enter num1, num2 and position: ");

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

int bit = (num2 >> pos) & 1;

num1 = (num1 & ~(1 << pos)) | (bit << pos);

printf("New number: %d\n", num1);

return 0;

}

**Output**

Enter num1, num2 and position: 8 3 0

New number: 9

**56: C Program to Swap Two Integers Using Bitwise Operators**

#include <stdio.h>

int main() {

int a, b;

printf("Enter two numbers: ");

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

a ^= b;

b ^= a;

a ^= b;

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

return 0;

}

**Output**

Enter two numbers: 12 25

After swapping: a=25, b=12

**57: C Program to Check if nth Bit is Set**

#include <stdio.h>

int main() {

int n, pos;

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

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

if (n & (1 << pos))

printf("Bit is SET\n");

else

printf("Bit is NOT set\n");

return 0;

}

**Output**

Enter number and bit position: 8 3

Bit is SET

**58: C Program to Check Alternate Bit Pattern**

#include <stdio.h>

int main() {

unsigned int n;

printf("Enter number: ");

scanf("%u", &n);

if ((n & (n >> 1)) == 0)

printf("Contains alternate bit pattern\n");

else

printf("Does not contain alternate bit pattern\n");

return 0;

}

**Output**

Enter number: 85

Contains alternate bit pattern

**59: C Program to Find Next Power of 2**

#include <stdio.h>

int main() {

unsigned int n;

printf("Enter number: ");

scanf("%u", &n);

n--;

n |= n >> 1;

n |= n >> 2;

n |= n >> 4;

n |= n >> 8;

n |= n >> 16;

n++;

printf("Next power of 2: %u\n", n);

return 0;

}

**Output**

Enter number: 10

Next power of 2: 16

**🧾 Program 60: C Program to Find Position of MSB**

#include <stdio.h>

int main() {

unsigned int n;

printf("Enter a number: ");

scanf("%u", &n);

int pos = 0;

while (n >>= 1)

pos++;

printf("MSB position: %d\n", pos);

return 0;

}

**Output**

Enter a number: 16

MSB position: 4

**61: C Program to Round Off to Next Lower Multiple of 2**

#include <stdio.h>

int main() {

int n;

printf("Enter number: ");

scanf("%d", &n);

n = n & (~1);

printf("Rounded off to lower multiple of 2: %d\n", n);

return 0;

}

**Output**

Enter number: 15

Rounded off to lower multiple of 2: 14

**62: C Program to Count Leading Zeros**

#include <stdio.h>

int main() {

unsigned int n;

printf("Enter number: ");

scanf("%u", &n);

int count = 0;

for (int i = 31; i >= 0; i--) {

if ((n >> i) & 1)

break;

count++;

}

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

return 0;

}

**Output**

Enter number: 8

Leading zeros: 28

**63: C Program to Check Power of 2**

#include <stdio.h>

int main() {

unsigned int n;

printf("Enter number: ");

scanf("%u", &n);

if (n && !(n & (n - 1)))

printf("Power of 2\n");

else

printf("Not power of 2\n");

return 0;

}

**Output**

Enter number: 16

Power of 2

**64: C program to print indexes of a particular character in a string**

#include <stdio.h>

#include <string.h>

int main() {

char s[200], ch;

int len, found = 0;

printf("Enter string: ");

fgets(s, sizeof s, stdin);

if (s[strlen(s)-1] == '\n') s[strlen(s)-1] = '\0';

printf("Enter character to find: ");

scanf("%c", &ch);

len = strlen(s);

printf("Indexes: ");

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

if (s[i] == ch) {

printf("%d ", i);

found = 1;

}

}

if (!found) printf("Not found");

printf("\n");

return 0;

}

**Output**

Enter string: banana

Enter character to find: a

Indexes: 1 3 5

**65: C program to compare two strings using pointers**

#include <stdio.h>

int strcmp\_ptr(const char \*a, const char \*b) {

while (\*a && (\*a == \*b)) { a++; b++; }

return (unsigned char)\*a - (unsigned char)\*b;

}

int main() {

char s1[200], s2[200];

printf("Enter first string: ");

fgets(s1, sizeof s1, stdin);

if (s1[strlen(s1)-1] == '\n') s1[strlen(s1)-1] = '\0';

printf("Enter second string: ");

fgets(s2, sizeof s2, stdin);

if (s2[strlen(s2)-1] == '\n') s2[strlen(s2)-1] = '\0';

int res = strcmp\_ptr(s1, s2);

if (res == 0) printf("Strings are equal\n");

else if (res < 0) printf("First is less than second\n");

else printf("First is greater than second\n");

return 0;

}

**Output**

Enter first string: apple

Enter second string: banana

First is less than second

**66: C program to create and print array of strings**

#include <stdio.h>

#include <string.h>

int main() {

int n;

printf("How many strings: ");

if (scanf("%d", &n) != 1) return 0;

getchar();

char arr[n][200];

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

printf("Enter string %d: ", i+1);

fgets(arr[i], sizeof arr[i], stdin);

if (arr[i][strlen(arr[i])-1] == '\n') arr[i][strlen(arr[i])-1] = '\0';

}

printf("You entered:\n");

for (int i = 0; i < n; i++) printf("%s\n", arr[i]);

return 0;

}

**Output**

How many strings: 3

Enter string 1: Alpha

Enter string 2: Beta

Enter string 3: Gamma

You entered:

Alpha

Beta

Gamma

**67: C program to capitalize first character of each word in a string**

#include <stdio.h>

#include <ctype.h>

#include <string.h>

int main() {

char s[500];

printf("Enter string: ");

fgets(s, sizeof s, stdin);

if (s[strlen(s)-1] == '\n') s[strlen(s)-1] = '\0';

int newWord = 1;

for (int i = 0; s[i]; i++) {

if (isspace((unsigned char)s[i])) {

newWord = 1;

} else {

if (newWord) s[i] = toupper((unsigned char)s[i]);

else s[i] = tolower((unsigned char)s[i]);

newWord = 0;

}

}

printf("Result: %s\n", s);

return 0;

}

**Output**

Enter string: hi there HOW are you

Result: Hi There How Are You

**68: C program to find the frequency of a character in a string**

#include <stdio.h>

#include <string.h>

int main() {

char s[300], ch;

int count = 0;

printf("Enter string: ");

fgets(s, sizeof s, stdin);

if (s[strlen(s)-1] == '\n') s[strlen(s)-1] = '\0';

printf("Enter character: ");

scanf("%c", &ch);

for (int i = 0; s[i]; i++)

if (s[i] == ch) count++;

printf("Frequency of '%c' = %d\n", ch, count);

return 0;

}

**Output**

Enter string: programming

Enter character: g

Frequency of 'g' = 2

**69: C program to read a string and print the length of each word**

#include <stdio.h>

#include <ctype.h>

#include <string.h>

int main() {

char s[500];

printf("Enter string: ");

fgets(s, sizeof s, stdin);

if (s[strlen(s)-1] == '\n') s[strlen(s)-1] = '\0';

int len = 0;

printf("Lengths: ");

for (int i = 0; s[i]; i++) {

if (!isspace((unsigned char)s[i])) len++;

else {

if (len > 0) { printf("%d ", len); len = 0; }

}

}

if (len > 0) printf("%d", len);

printf("\n");

return 0;

}

**Output**

Enter string: Hi there how are you?

Lengths: 2 5 3 3 4

**70: C program to eliminate/remove all vowels from a string**

#include <stdio.h>

#include <string.h>

#include <ctype.h>

int is\_vowel(char c) {

c = tolower((unsigned char)c);

return c=='a'||c=='e'||c=='i'||c=='o'||c=='u';

}

int main() {

char s[500], out[500];

printf("Enter string: ");

fgets(s, sizeof s, stdin);

if (s[strlen(s)-1] == '\n') s[strlen(s)-1] = '\0';

int j = 0;

for (int i = 0; s[i]; i++) {

if (!is\_vowel(s[i])) out[j++] = s[i];

}

out[j] = '\0';

printf("Without vowels: %s\n", out);

return 0;

}

**Output**

Enter string: education

Without vowels: dctn

**71: C program to eliminate/remove first character of each word from a string**

#include <stdio.h>

#include <ctype.h>

#include <string.h>

int main() {

char s[500], out[500];

printf("Enter string: ");

fgets(s, sizeof s, stdin);

if (s[strlen(s)-1] == '\n') s[strlen(s)-1] = '\0';

int j = 0, inWord = 0;

for (int i = 0; s[i]; i++) {

if (!isspace((unsigned char)s[i]) && !inWord) {

inWord = 1; // skip this first char

continue;

} else if (isspace((unsigned char)s[i])) {

inWord = 0;

out[j++] = s[i];

} else {

out[j++] = s[i];

}

}

out[j] = '\0';

printf("Result: %s\n", out);

return 0;

}

**Output**

Enter string: Hello World Example

Result: ello orld xample

**72: C program to read n strings and print each string's length**

#include <stdio.h>

#include <string.h>

int main() {

int n;

printf("How many strings: ");

scanf("%d", &n);

getchar();

char s[200];

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

printf("Enter string %d: ", i+1);

fgets(s, sizeof s, stdin);

if (s[strlen(s)-1] == '\n') s[strlen(s)-1] = '\0';

printf("Length = %zu\n", strlen(s));

}

return 0;

}

**Output**

How many strings: 3

Enter string 1: one

Length = 3

Enter string 2: two words

Length = 9

Enter string 3: abc

Length = 3

**73: C program to copy one string to another and count copied characters (without library)**

#include <stdio.h>

int main() {

char src[500], dest[500];

int i = 0;

printf("Enter source string: ");

fgets(src, sizeof src, stdin);

if (src[strlen(src)-1] == '\n') src[strlen(src)-1] = '\0';

while (src[i]) {

dest[i] = src[i];

i++;

}

dest[i] = '\0';

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

printf("Copied characters: %d\n", i);

return 0;

}

**Output**

Enter source string: Hello!

Copied string: Hello!

Copied characters: 6

**74: C program to remove all spaces from a given string**

#include <stdio.h>

#include <string.h>

#include <ctype.h>

int main() {

char s[500], out[500];

printf("Enter string: ");

fgets(s, sizeof s, stdin);

if (s[strlen(s)-1] == '\n') s[strlen(s)-1] = '\0';

int j = 0;

for (int i = 0; s[i]; i++) {

if (!isspace((unsigned char)s[i])) out[j++] = s[i];

}

out[j] = '\0';

printf("Without spaces: %s\n", out);

return 0;

}

**Output**

Enter string: a b c d

Without spaces: abcd

**75: C program to convert a string to sentence case**

#include <stdio.h>

#include <ctype.h>

#include <string.h>

int main() {

char s[500];

printf("Enter string: ");

fgets(s, sizeof s, stdin);

if (s[strlen(s)-1] == '\n') s[strlen(s)-1] = '\0';

int start = 1;

for (int i = 0; s[i]; i++) {

if (start && isalpha((unsigned char)s[i])) {

s[i] = toupper((unsigned char)s[i]);

start = 0;

} else {

s[i] = tolower((unsigned char)s[i]);

}

if (s[i] == '.' || s[i] == '?' || s[i] == '!') start = 1;

if (isspace((unsigned char)s[i]) && s[i-1] != '.' && s[i-1] != '?' && s[i-1] != '!') {

/\* continue sentence \*/

}

}

printf("Sentence case: %s\n", s);

return 0;

}

**Output**

Enter string: hELLO. how ARE you?

Sentence case: Hello. How are you?

**76: C program to remove alphabets from an alphanumeric string**

#include <stdio.h>

#include <ctype.h>

#include <string.h>

int main() {

char s[500], out[500];

printf("Enter alphanumeric string: ");

fgets(s, sizeof s, stdin);

if (s[strlen(s)-1] == '\n') s[strlen(s)-1] = '\0';

int j = 0;

for (int i = 0; s[i]; i++) {

if (!isalpha((unsigned char)s[i])) out[j++] = s[i];

}

out[j] = '\0';

printf("After removing alphabets: %s\n", out);

return 0;

}

**Output**

Enter alphanumeric string: a1b2c3

After removing alphabets: 123

**77: C program to eliminate all vowels from a string (duplicate of earlier but provided again as requested)**

#include <stdio.h>

#include <string.h>

#include <ctype.h>

int is\_vowel(char c) {

c = tolower((unsigned char)c);

return c=='a'||c=='e'||c=='i'||c=='o'||c=='u';

}

int main() {

char s[500], out[500];

printf("Enter string: ");

fgets(s, sizeof s, stdin);

if (s[strlen(s)-1] == '\n') s[strlen(s)-1] = '\0';

int j = 0;

for (int i = 0; s[i]; i++) {

if (!is\_vowel(s[i])) out[j++] = s[i];

}

out[j] = '\0';

printf("Without vowels: %s\n", out);

return 0;

}

**Output**

Enter string: Orange

Without vowels: rng

**78: C program to swap adjacent characters of a string (string length should be even)**

#include <stdio.h>

#include <string.h>

int main() {

char s[500];

printf("Enter string (even length): ");

fgets(s, sizeof s, stdin);

int len = strlen(s);

if (len && s[len-1] == '\n') { s[len-1] = '\0'; len--; }

for (int i = 0; i+1 < len; i += 2) {

char t = s[i];

s[i] = s[i+1];

s[i+1] = t;

}

printf("After swapping: %s\n", s);

return 0;

}

**Output**

Enter string (even length): abcdef

After swapping: badcfe

**Program 79: C program to read time in string format and extract hours, minutes and seconds and check validity (HH:MM:SS)**

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

int main() {

char time[20];

int hh, mm, ss;

printf("Enter time (HH:MM:SS): ");

fgets(time, sizeof time, stdin);

if (time[strlen(time)-1] == '\n') time[strlen(time)-1] = '\0';

if (sscanf(time, "%d:%d:%d", &hh, &mm, &ss) != 3) {

printf("Invalid format\n");

return 0;

}

if (hh < 0 || hh > 23 || mm < 0 || mm > 59 || ss < 0 || ss > 59) {

printf("Invalid time\n");

} else {

printf("Hours=%d Minutes=%d Seconds=%d\n", hh, mm, ss);

}

return 0;

}

**Output**

Enter time (HH:MM:SS): 12:30:45

Hours=12 Minutes=30 Seconds=45

**Program 80: Creating string buffer (character pointer), allocating memory at run time in C**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int main() {

size\_t n;

printf("Enter required buffer size: ");

if (scanf("%zu", &n) != 1) return 0;

getchar();

char \*buf = malloc(n);

if (!buf) { printf("Allocation failed\n"); return 1; }

printf("Enter string: ");

fgets(buf, n, stdin);

if (buf[strlen(buf)-1] == '\n') buf[strlen(buf)-1] = '\0';

printf("You entered: %s\n", buf);

free(buf);

return 0;

}

**Output**

Enter required buffer size: 50

Enter string: dynamic buffer test

You entered: dynamic buffer test

**81: memcpy() function in C with Example**

#include <stdio.h>

#include <string.h>

int main() {

char src[] = "Hello memcpy";

char dest[50];

memcpy(dest, src, strlen(src)+1);

printf("Source: %s\nDestination: %s\n", src, dest);

return 0;

}

**Output**

Source: Hello memcpy

Destination: Hello memcpy

**Program 82: Write your own memcpy() function in C**

#include <stdio.h>

void \*my\_memcpy(void \*dest, const void \*src, size\_t n) {

unsigned char \*d = dest;

const unsigned char \*s = src;

while (n--) \*d++ = \*s++;

return dest;

}

int main() {

char src[] = "Custom memcpy";

char dest[50];

my\_memcpy(dest, src, sizeof src);

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

return 0;

}

**Output**

Copied: Custom memcpy

**Program 83: memset() function in C with Example**

#include <stdio.h>

#include <string.h>

int main() {

char buffer[20];

memset(buffer, 'A', sizeof buffer-1);

buffer[sizeof buffer-1] = '\0';

printf("Buffer: %s\n", buffer);

return 0;

}

**Output (sample)**

Buffer: AAAAAAAAAAAAAAAAAAA

**Program 84: Write your own memset() function in C**

#include <stdio.h>

void \*my\_memset(void \*s, int c, size\_t n) {

unsigned char \*p = s;

while (n--) \*p++ = (unsigned char)c;

return s;

}

int main() {

char buffer[10];

my\_memset(buffer, 'x', 9);

buffer[9] = '\0';

printf("Buffer: %s\n", buffer);

return 0;

}

**Output**

Buffer: xxxxxxxxx

**Program 85: C program to compare strings using strcmp() function**

#include <stdio.h>

#include <string.h>

int main() {

char s1[200], s2[200];

printf("Enter first: ");

fgets(s1, sizeof s1, stdin);

if (s1[strlen(s1)-1] == '\n') s1[strlen(s1)-1] = '\0';

printf("Enter second: ");

fgets(s2, sizeof s2, stdin);

if (s2[strlen(s2)-1] == '\n') s2[strlen(s2)-1] = '\0';

int r = strcmp(s1, s2);

if (r == 0) printf("Equal\n");

else if (r < 0) printf("First < Second\n");

else printf("First > Second\n");

return 0;

}

**Output**

Enter first: test

Enter second: test

Equal

**Program 86: C program to check a string is palindrome or not without using library function**

#include <stdio.h>

#include <string.h>

#include <ctype.h>

int main() {

char s[500];

printf("Enter string: ");

fgets(s, sizeof s, stdin);

int len = strlen(s);

if (len && s[len-1] == '\n') { s[--len] = '\0'; }

int i = 0, j = len - 1;

while (i < j) {

while (i < j && !isalnum((unsigned char)s[i])) i++;

while (i < j && !isalnum((unsigned char)s[j])) j--;

if (tolower((unsigned char)s[i]) != tolower((unsigned char)s[j])) {

printf("Not palindrome\n");

return 0;

}

i++; j--;

}

printf("Palindrome\n");

return 0;

}

**Output**

Enter string: A man, a plan, a canal: Panama

Palindrome

**Program 87: C program to check a string is palindrome or not using recursion**

#include <stdio.h>

#include <string.h>

#include <ctype.h>

int ispal(char \*s, int i, int j) {

while (i < j && !isalnum((unsigned char)s[i])) i++;

while (i < j && !isalnum((unsigned char)s[j])) j--;

if (i >= j) return 1;

if (tolower((unsigned char)s[i]) != tolower((unsigned char)s[j])) return 0;

return ispal(s, i+1, j-1);

}

int main() {

char s[500];

fgets(s, sizeof s, stdin);

int len = strlen(s);

if (len && s[len-1]=='\n') s[--len]='\0';

if (ispal(s, 0, len-1)) printf("Palindrome\n");

else printf("Not palindrome\n");

return 0;

}

**Output**

Aba

Palindrome

**Program 88: C program to print the biggest and smallest palindrome words in a string**

#include <stdio.h>

#include <string.h>

#include <ctype.h>

int ispalword(const char \*s, int len) {

int i = 0, j = len-1;

while (i < j) {

if (tolower((unsigned char)s[i]) != tolower((unsigned char)s[j])) return 0;

i++; j--;

}

return 1;

}

int main() {

char s[1000];

printf("Enter string: ");

fgets(s, sizeof s, stdin);

if (s[strlen(s)-1]=='\n') s[strlen(s)-1]=0;

char \*p = s;

int maxlen = 0, minlen = 10000;

char maxw[200] = "", minw[200] = "";

while (\*p) {

while (\*p && isspace((unsigned char)\*p)) p++;

if (!\*p) break;

char word[200]; int i=0;

while (\*p && !isspace((unsigned char)\*p)) word[i++] = \*p++;

word[i] = '\0';

if (i>0 && ispalword(word, i)) {

if (i > maxlen) { maxlen = i; strcpy(maxw, word); }

if (i < minlen) { minlen = i; strcpy(minw, word); }

}

}

if (maxlen==0) printf("No palindrome words found\n");

else {

printf("Biggest palindrome: %s (len=%d)\n", maxw, maxlen);

printf("Smallest palindrome: %s (len=%d)\n", minw, minlen);

}

return 0;

}

**Output**

Enter string: level noon civic rotor kayak refer

Biggest palindrome: level (len=5)

Smallest palindrome: noon (len=4)

**Program 89: C program to print the smallest word in a string**

#include <stdio.h>

#include <string.h>

#include <ctype.h>

int main() {

char s[1000];

printf("Enter string: ");

fgets(s, sizeof s, stdin);

if (s[strlen(s)-1]=='\n') s[strlen(s)-1]=0;

char \*p = s;

int minlen = 10000; char minw[200] = "";

while (\*p) {

while (\*p && isspace((unsigned char)\*p)) p++;

if (!\*p) break;

char word[200]; int i=0;

while (\*p && !isspace((unsigned char)\*p)) word[i++]=\*p++;

word[i]=0;

if (i < minlen) { minlen = i; strcpy(minw, word); }

}

if (minlen==10000) printf("No words\n");

else printf("Smallest word: %s (len=%d)\n", minw, minlen);

return 0;

}

**Output**

Enter string: this is an example

Smallest word: an (len=2)

**Program 90: C program to print the biggest word in a string**

#include <stdio.h>

#include <string.h>

#include <ctype.h>

int main() {

char s[1000];

printf("Enter string: ");

fgets(s, sizeof s, stdin);

if (s[strlen(s)-1]=='\n') s[strlen(s)-1]=0;

char \*p = s;

int maxlen = 0; char maxw[200]="";

while (\*p) {

while (\*p && isspace((unsigned char)\*p)) p++;

if (!\*p) break;

char word[200]; int i=0;

while (\*p && !isspace((unsigned char)\*p)) word[i++]=\*p++;

word[i]=0;

if (i > maxlen) { maxlen = i; strcpy(maxw, word); }

}

if (maxlen==0) printf("No words\n");

else printf("Biggest word: %s (len=%d)\n", maxw, maxlen);

return 0;

}

**Output**

Enter string: find the longestword here

Biggest word: longestword (len=11)

**Program 91: C program to reverse a string using recursion**

#include <stdio.h>

#include <string.h>

void rev(char \*s, int i, int j) {

if (i >= j) return;

char t = s[i]; s[i] = s[j]; s[j] = t;

rev(s, i+1, j-1);

}

int main() {

char s[500];

printf("Enter string: ");

fgets(s, sizeof s, stdin);

int len = strlen(s);

if (len && s[len-1]=='\n') s[--len]=0;

rev(s, 0, len-1);

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

return 0;

}

**Output**

Enter string: Hello

Reversed: olleH

**Program 92: C program to reverse every word of the given string**

#include <stdio.h>

#include <string.h>

#include <ctype.h>

void revrange(char \*s, int i, int j) {

while (i < j) {

char t = s[i]; s[i] = s[j]; s[j] = t;

i++; j--;

}

}

int main() {

char s[1000];

printf("Enter string: ");

fgets(s, sizeof s, stdin);

int len = strlen(s);

if (len && s[len-1]=='\n') s[--len]=0;

int start = 0;

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

if (s[i] == ' ' || s[i]=='\0') {

revrange(s, start, i-1);

start = i+1;

}

}

printf("Result: %s\n", s);

return 0;

}

**Output**

Enter string: Hello world

Result: olleH dlrow

**Program 93: C program to remove a given word from the string**

#include <stdio.h>

#include <string.h>

int main() {

char s[1000], word[200], out[1000];

printf("Enter string: ");

fgets(s, sizeof s, stdin);

if (s[strlen(s)-1]=='\n') s[strlen(s)-1]=0;

printf("Enter word to remove: ");

fgets(word, sizeof word, stdin);

if (word[strlen(word)-1]=='\n') word[strlen(word)-1]=0;

out[0]=0;

char \*p = strtok(s, " ");

int first = 1;

while (p) {

if (strcmp(p, word) != 0) {

if (!first) strcat(out, " ");

strcat(out, p);

first = 0;

}

p = strtok(NULL, " ");

}

printf("Result: %s\n", out);

return 0;

}

**Output**

Enter string: this is test this is

Enter word to remove: this

Result: is test is

**Program 94: C program to delete duplicate words in the string**

#include <stdio.h>

#include <string.h>

int main() {

char s[1000];

printf("Enter string: ");

fgets(s, sizeof s, stdin);

if (s[strlen(s)-1]=='\n') s[strlen(s)-1]=0;

char \*words[200]; int wc=0;

char \*p = strtok(s, " ");

while (p) {

int found = 0;

for (int i=0;i<wc;i++) if (strcmp(words[i], p)==0) { found = 1; break; }

if (!found) {

words[wc++] = strdup(p);

}

p = strtok(NULL, " ");

}

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

printf("%s", words[i]);

if (i < wc-1) printf(" ");

}

printf("\n");

// free duplicated memory

for (int i=0;i<wc;i++) free(words[i]);

return 0;

}

**Output**

Enter string: this is is a test test

this is a test

**Program 95: C program to sort strings in alphabetical order**

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

int main() {

int n;

printf("How many strings: ");

if (scanf("%d", &n) != 1) return 0;

getchar();

char \*\*arr = malloc(n \* sizeof \*arr);

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

char buf[500];

fgets(buf, sizeof buf, stdin);

if (buf[strlen(buf)-1]=='\n') buf[strlen(buf)-1]=0;

arr[i] = strdup(buf);

}

qsort(arr, n, sizeof(char\*), (int(\*)(const void\*,const void\*)) strcmp);

printf("Sorted:\n");

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

printf("%s\n", arr[i]);

free(arr[i]);

}

free(arr);

return 0;

}

**Output**

How many strings: 3

apple

banana

apricot

Sorted:

apricot

apple

banana

**Program 96: Convert number from Decimal to Binary**

#include <stdio.h>

int main() {

int n;

printf("Enter decimal number: ");

scanf("%d", &n);

printf("Binary: ");

for (int i = 31; i >= 0; i--) {

int b = (n >> i) & 1;

if (b || i == 0) printf("%d", b);

}

printf("\n");

return 0;

}

**Output:**

Enter decimal number: 10

Binary: 1010

**Program 97: Convert number from Decimal to Octal**

#include <stdio.h>

int main() {

int n;

printf("Enter decimal number: ");

scanf("%d", &n);

printf("Octal: %o\n", n);

return 0;

}

**Output:**

Enter decimal number: 10

Octal: 12

**Program 98: Convert number from Decimal to Hexadecimal**

#include <stdio.h>

int main() {

int n;

printf("Enter decimal number: ");

scanf("%d", &n);

printf("Hexadecimal: %X\n", n);

return 0;

}

**Output:**

Enter decimal number: 255

Hexadecimal: FF

**Program 99: Convert number from Binary to Decimal**

#include <stdio.h>

#include <string.h>

#include <math.h>

int main() {

char bin[100];

printf("Enter binary number: ");

scanf("%s", bin);

int decimal = 0, len = strlen(bin);

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

if (bin[i] == '1') decimal += 1 << (len-i-1);

}

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

return 0;

}

**Output:**

Enter binary number: 1010

Decimal: 10

**Program 100: Convert number from Octal to Decimal**

#include <stdio.h>

int main() {

int oct, decimal = 0, i = 0, rem;

printf("Enter octal number: ");

scanf("%o", &oct);

int temp = oct;

while (temp != 0) {

rem = temp % 10;

decimal += rem << (3\*i);

i++;

temp /= 10;

}

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

return 0;

}

**Output:**

Enter octal number: 12

Decimal: 10

**Program 101: Convert number from Hexadecimal to Decimal**

#include <stdio.h>

#include <stdlib.h>

int main() {

char hex[20];

printf("Enter hexadecimal number: ");

scanf("%s", hex);

int decimal = (int)strtol(hex, NULL, 16);

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

return 0;

}

**Output:**

Enter hexadecimal number: FF

Decimal: 255

**Program 102: Convert Binary number to Octal number**

#include <stdio.h>

#include <string.h>

int main() {

char bin[100];

printf("Enter binary number: ");

scanf("%s", bin);

int decimal = 0, len = strlen(bin);

for (int i=0; i<len; i++) if (bin[i]=='1') decimal += 1 << (len-i-1);

printf("Octal: %o\n", decimal);

return 0;

}

**Output:**

Enter binary number: 1010

Octal: 12

**Program 103: Convert Binary number to Hexadecimal number**

#include <stdio.h>

#include <string.h>

int main() {

char bin[100];

printf("Enter binary number: ");

scanf("%s", bin);

int decimal = 0, len = strlen(bin);

for (int i=0; i<len; i++) if (bin[i]=='1') decimal += 1 << (len-i-1);

printf("Hexadecimal: %X\n", decimal);

return 0;

}

**Output:**

Enter binary number: 1010

Hexadecimal: A

**Program 104: Convert Octal number into Binary number**

#include <stdio.h>

int main() {

int oct;

printf("Enter octal number: ");

scanf("%o", &oct);

int decimal = oct;

int bin[32], i = 0;

while(decimal > 0) {

bin[i++] = decimal % 2;

decimal /= 2;

}

printf("Binary: ");

for(int j = i-1; j >= 0; j--) printf("%d", bin[j]);

printf("\n");

return 0;

}

**Output:**

Enter octal number: 12

Binary: 1010

**Program 105: Convert Hexadecimal number into Binary number**

#include <stdio.h>

#include <stdlib.h>

int main() {

char hex[20];

printf("Enter hexadecimal number: ");

scanf("%s", hex);

int decimal = (int)strtol(hex, NULL, 16);

int bin[32], i = 0;

while(decimal > 0) {

bin[i++] = decimal % 2;

decimal /= 2;

}

printf("Binary: ");

for(int j = i-1; j >= 0; j--) printf("%d", bin[j]);

printf("\n");

return 0;

}

**Output:**

Enter hexadecimal number: A

Binary: 1010

**Program 106: Convert Decimal number to Roman number**

#include <stdio.h>

int main() {

int n;

printf("Enter decimal number (1-3999): ");

scanf("%d", &n);

int val[] = {1000,900,500,400,100,90,50,40,10,9,5,4,1};

char \*rom[] = {"M","CM","D","CD","C","XC","L","XL","X","IX","V","IV","I"};

printf("Roman: ");

for(int i=0; i<13; i++) {

while(n >= val[i]) {

printf("%s", rom[i]);

n -= val[i];

}

}

printf("\n");

return 0;

}

**Output:**

Enter decimal number (1-3999): 1994

Roman: MCMXCIV

**Program 107: Convert Roman number to Decimal number**

#include <stdio.h>

#include <string.h>

int value(char r) {

if(r=='I') return 1;

if(r=='V') return 5;

if(r=='X') return 10;

if(r=='L') return 50;

if(r=='C') return 100;

if(r=='D') return 500;

if(r=='M') return 1000;

return 0;

}

int main() {

char roman[20];

printf("Enter Roman number: ");

scanf("%s", roman);

int total=0, len=strlen(roman);

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

if(i+1<len && value(roman[i]) < value(roman[i+1])) total -= value(roman[i]);

else total += value(roman[i]);

}

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

return 0;

}

**Output:**

Enter Roman number: MCMXCIV

Decimal: 1994

**Program 108: Convert Decimal to Binary using Recursion**

#include <stdio.h>

void decToBin(int n) {

if(n > 1) decToBin(n/2);

printf("%d", n%2);

}

int main() {

int n;

printf("Enter decimal number: ");

scanf("%d", &n);

printf("Binary: ");

decToBin(n);

printf("\n");

return 0;

}

**Output:**

Enter decimal number: 10

Binary: 1010

**Program 109: Convert Binary number to Gray Code**

#include <stdio.h>

int main() {

int n;

printf("Enter binary number: ");

scanf("%d", &n);

int gray = n ^ (n>>1);

printf("Gray code: %d\n", gray);

return 0;

}

**Output:**

Enter binary number: 10

Gray code: 15

**Program 110: Convert Binary number to Gray Code using Recursion**

#include <stdio.h>

int binToGray(int n) {

if(n == 0) return 0;

return (n ^ (n>>1));

}

int main() {

int n;

printf("Enter binary number: ");

scanf("%d", &n);

printf("Gray code: %d\n", binToGray(n));

return 0;

}

**Output:**

Enter binary number: 10

Gray code: 15

**Program 111: Convert Decimal to Hexadecimal using Recursion**

#include <stdio.h>

void decToHex(int n) {

if(n == 0) return;

decToHex(n/16);

int rem = n % 16;

if(rem < 10) printf("%d", rem);

else printf("%c", 'A' + rem - 10);

}

int main() {

int n;

printf("Enter decimal number: ");

scanf("%d", &n);

if(n==0) printf("0");

else decToHex(n);

printf("\n");

return 0;

}

**Output:**

Enter decimal number: 255

FF

**Program 112: Initialising byte array with decimal, octal and hexadecimal numbers**

#include <stdio.h>

int main() {

unsigned char arr[] = {10, 012, 0xA, 15};

int n = sizeof(arr)/sizeof(arr[0]);

printf("Array elements:\n");

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

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

printf("\n");

return 0;

}

Output:

Array elements:

10 10 10 15

**Program 113: Count Array elements using sizeof() operator**

#include <stdio.h>

int main() {

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

int n = sizeof(arr)/sizeof(arr[0]);

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

return 0;

}

Output:

Number of elements: 6

**Program 114: Swap first element with last, second to second last (reverse array)**

#include <stdio.h>

int main() {

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

int n = sizeof(arr)/sizeof(arr[0]);

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

int temp = arr[i];

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

arr[n-i-1] = temp;

}

printf("Reversed array: ");

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

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

printf("\n");

return 0;

}

Output:

Reversed array: 5 4 3 2 1

**Program 115: Find nearest lesser and greater element in an array**

#include <stdio.h>

int main() {

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

int n = sizeof(arr)/sizeof(arr[0]);

int x = 5;

int lesser = -1, greater = -1;

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

if(arr[i]<x && (lesser==-1 || arr[i]>lesser)) lesser = arr[i];

if(arr[i]>x && (greater==-1 || arr[i]<greater)) greater = arr[i];

}

printf("Nearest lesser: %d, nearest greater: %d\n", lesser, greater);

return 0;

}

Output:

Nearest lesser: 4, nearest greater: 6

**Program 116: Merge two arrays into third array dynamically**

#include <stdio.h>

#include <stdlib.h>

int main() {

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

int n1 = sizeof(a)/sizeof(a[0]);

int n2 = sizeof(b)/sizeof(b[0]);

int \*c = malloc((n1+n2)\*sizeof(int));

for(int i=0;i<n1;i++) c[i]=a[i];

for(int i=0;i<n2;i++) c[n1+i]=b[i];

printf("Merged array: ");

for(int i=0;i<n1+n2;i++) printf("%d ", c[i]);

printf("\n");

free(c);

return 0;

}

Output:

Merged array: 1 2 3 4 5

**Program 117: Calculate median of an array**

#include <stdio.h>

#include <stdlib.h>

int compare(const void \*a, const void \*b) {

return (\*(int\*)a - \*(int\*)b);

}

int main() {

int arr[] = {5,2,8,1,3};

int n = sizeof(arr)/sizeof(arr[0]);

qsort(arr,n,sizeof(int),compare);

float median = (n%2) ? arr[n/2] : (arr[n/2-1]+arr[n/2])/2.0;

printf("Median: %.2f\n", median);

return 0;

}

Output:

Median: 3.00

**Program 118: Delete prime numbers from an array**

#include <stdio.h>

#include <math.h>

int isPrime(int n){

if(n<2) return 0;

for(int i=2;i<=sqrt(n);i++)

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

return 1;

}

int main(){

int arr[] = {2,3,4,5,6,7,8};

int n = sizeof(arr)/sizeof(arr[0]);

int res[100], k=0;

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

if(!isPrime(arr[i])) res[k++] = arr[i];

printf("Array after removing primes: ");

for(int i=0;i<k;i++) printf("%d ", res[i]);

printf("\n");

return 0;

}

Output:

Array after removing primes: 4 6 8

**Program 119: Check prime numbers in an array**

#include <stdio.h>

#include <math.h>

int isPrime(int n){

if(n<2) return 0;

for(int i=2;i<=sqrt(n);i++)

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

return 1;

}

int main(){

int arr[] = {2,3,4,5,6,7,8};

int n = sizeof(arr)/sizeof(arr[0]);

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

printf("%d is %s\n", arr[i], isPrime(arr[i])?"Prime":"Not Prime");

return 0;

}

Output:

2 is Prime

3 is Prime

4 is Not Prime

5 is Prime

6 is Not Prime

7 is Prime

8 is Not Prime

**Program 120: Create array with reverse elements of one dimensional array**

#include <stdio.h>

int main() {

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

int n = sizeof(arr)/sizeof(arr[0]);

int rev[5];

for(int i=0;i<n;i++) rev[i]=arr[n-i-1];

printf("Reversed array: ");

for(int i=0;i<n;i++) printf("%d ", rev[i]);

printf("\n");

return 0;

}

Output:

Reversed array: 5 4 3 2 1

**Program 121: Count total number of elements divisible by a specific number in an array**

#include <stdio.h>

int main() {

int arr[] = {2,4,6,7,9,12};

int n = sizeof(arr)/sizeof(arr[0]);

int count=0, k=3;

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

if(arr[i]%k==0) count++;

printf("Number of elements divisible by %d: %d\n", k, count);

return 0;

}

Output:

Number of elements divisible by 3: 4

**Program 122: Create a new array from a given array with elements divisible by a specific number**

#include <stdio.h>

int main() {

int arr[] = {2,3,4,5,6,7,8,9,12};

int n = sizeof(arr)/sizeof(arr[0]);

int k = 3, newArr[10], idx=0;

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

if(arr[i]%k==0) newArr[idx++] = arr[i];

printf("New array with elements divisible by %d: ", k);

for(int i=0;i<idx;i++) printf("%d ", newArr[i]);

printf("\n");

return 0;

}

Output:

New array with elements divisible by 3: 3 6 9 12

**Program 123: Find second largest element in a one dimensional array**

#include <stdio.h>

int main() {

int arr[] = {5,1,9,2,7};

int n = sizeof(arr)/sizeof(arr[0]);

int largest = arr[0], second = -2147483648;

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

if(arr[i]>largest){

second = largest;

largest = arr[i];

} else if(arr[i]>second && arr[i]!=largest){

second = arr[i];

}

}

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

return 0;

}

Output:

Second largest element: 7

**Program 124: Find two largest elements in a one dimensional array**

#include <stdio.h>

int main() {

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

int n = sizeof(arr)/sizeof(arr[0]);

int first=-2147483648, second=-2147483648;

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

if(arr[i]>first){

second = first;

first = arr[i];

} else if(arr[i]>second){

second = arr[i];

}

}

printf("Two largest elements: %d %d\n", first, second);

return 0;

}

Output:

Two largest elements: 10 8

**Program 125: Find second smallest element in a one dimensional array**

#include <stdio.h>

int main() {

int arr[] = {7,2,5,1,9};

int n = sizeof(arr)/sizeof(arr[0]);

int first=2147483647, second=2147483647;

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

if(arr[i]<first){

second = first;

first = arr[i];

} else if(arr[i]<second && arr[i]!=first){

second = arr[i];

}

}

printf("Second smallest element: %d\n", second);

return 0;

}

Output:

Second smallest element: 2

**Program 126: Find two smallest elements in a one dimensional array**

#include <stdio.h>

int main() {

int arr[] = {7,2,5,1,9};

int n = sizeof(arr)/sizeof(arr[0]);

int first=2147483647, second=2147483647;

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

if(arr[i]<first){

second = first;

first = arr[i];

} else if(arr[i]<second){

second = arr[i];

}

}

printf("Two smallest elements: %d %d\n", first, second);

return 0;

}

Output:

Two smallest elements: 1 2

**Program 127: Cyclically permute elements of an array**

#include <stdio.h>

int main() {

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

int n = sizeof(arr)/sizeof(arr[0]);

int temp = arr[n-1];

for(int i=n-1;i>0;i--) arr[i]=arr[i-1];

arr[0]=temp;

printf("Cyclically permuted array: ");

for(int i=0;i<n;i++) printf("%d ", arr[i]);

printf("\n");

return 0;

}

Output:

Cyclically permuted array: 5 1 2 3 4

**Program 128: Binary search on a sorted array**

#include <stdio.h>

int main() {

int arr[] = {1,3,5,7,9};

int n = sizeof(arr)/sizeof(arr[0]), key=5;

int low=0, high=n-1, mid;

while(low<=high){

mid=(low+high)/2;

if(arr[mid]==key){ printf("Element found at index %d\n", mid); break; }

else if(arr[mid]<key) low=mid+1;

else high=mid-1;

}

if(low>high) printf("Element not found\n");

return 0;

}

Output:

Element found at index 2

**Program 129: Find biggest number in an array using recursion**

#include <stdio.h>

int maxRec(int arr[], int n){

if(n==1) return arr[0];

int max = maxRec(arr, n-1);

return (arr[n-1]>max)? arr[n-1]:max;

}

int main() {

int arr[] = {2,7,3,9,5};

int n = sizeof(arr)/sizeof(arr[0]);

printf("Biggest number: %d\n", maxRec(arr,n));

return 0;

}

Output:

Biggest number: 9

**Program 130: Print number of subsets whose elements have difference 0 or 1**

#include <stdio.h>

int main() {

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

int n = sizeof(arr)/sizeof(arr[0]);

int count=0;

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

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

int min=arr[i], max=arr[i];

for(int k=i;k<=j;k++){

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

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

}

if(max-min<=1) count++;

}

printf("Number of subsets with difference 0 or 1: %d\n", count);

return 0;

}

Output:

Number of subsets with difference 0 or 1: 8

**Program 131: Read and print one dimensional array of integers**

#include <stdio.h>

int main() {

int arr[10];

printf("Enter 10 elements:\n");

for(int i=0;i<10;i++) scanf("%d",&arr[i]);

printf("Array elements: ");

for(int i=0;i<10;i++) printf("%d ", arr[i]);

printf("\n");

return 0;

}

Output (sample):

Enter 10 elements:

1 2 3 4 5 6 7 8 9 10

Array elements: 1 2 3 4 5 6 7 8 9 10

**Program 132: Calculate sum and product of all array elements**

#include <stdio.h>

int main() {

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

int sum=0, product=1;

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

sum += arr[i];

product \*= arr[i];

}

printf("Sum: %d, Product: %d\n", sum, product);

return 0;

}

Output:

Sum: 15, Product: 120

**Program 133: Find smallest and largest elements in an array**

#include <stdio.h>

int main() {

int arr[5] = {7,3,9,2,5};

int smallest=arr[0], largest=arr[0];

for(int i=1;i<5;i++){

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

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

}

printf("Smallest: %d, Largest: %d\n", smallest, largest);

return 0;

}

Output:

Smallest: 2, Largest: 9

**Program 134: Replace even elements by 0 and odd elements by 1**

#include <stdio.h>

int main() {

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

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

arr[i] = arr[i]%2;

}

printf("Modified array: ");

for(int i=0;i<5;i++) printf("%d ", arr[i]);

printf("\n");

return 0;

}

Output:

Modified array: 1 0 1 0 1

**Program 135: Merge two one dimensional arrays**

#include <stdio.h>

int main() {

int arr1[3]={1,2,3}, arr2[2]={4,5}, merged[5];

for(int i=0;i<3;i++) merged[i]=arr1[i];

for(int i=0;i<2;i++) merged[i+3]=arr2[i];

printf("Merged array: ");

for(int i=0;i<5;i++) printf("%d ", merged[i]);

printf("\n");

return 0;

}

Output:

Merged array: 1 2 3 4 5

**Program 136: Add and subtract two arrays element-wise**

#include <stdio.h>

int main() {

int arr1[3]={1,2,3}, arr2[3]={4,5,6}, sum[3], diff[3];

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

sum[i]=arr1[i]+arr2[i];

diff[i]=arr1[i]-arr2[i];

}

printf("Sum: ");

for(int i=0;i<3;i++) printf("%d ", sum[i]);

printf("\nDifference: ");

for(int i=0;i<3;i++) printf("%d ", diff[i]);

printf("\n");

return 0;

}

Output:

Sum: 5 7 9

Difference: -3 -3 -3

**Program 137: Find a number in array elements**

#include <stdio.h>

int main() {

int arr[5]={1,3,5,7,9}, key=5;

int found=0;

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

if(arr[i]==key){

printf("Number %d found at index %d\n", key, i);

found=1;

break;

}

}

if(!found) printf("Number not found\n");

return 0;

}

Output:

Number 5 found at index 2

**Program 138: Sort array elements in ascending order**

#include <stdio.h>

int main() {

int arr[5]={5,2,8,1,4}, temp;

for(int i=0;i<5-1;i++){

for(int j=i+1;j<5;j++){

if(arr[i]>arr[j]){

temp=arr[i]; arr[i]=arr[j]; arr[j]=temp;

}

}

}

printf("Sorted array: ");

for(int i=0;i<5;i++) printf("%d ", arr[i]);

printf("\n");

return 0;

}

Output:

Sorted array: 1 2 4 5 8

**Program 139: Reverse array elements**

#include <stdio.h>

int main() {

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

for(int i=0;i<5/2;i++){

int temp=arr[i]; arr[i]=arr[5-1-i]; arr[5-1-i]=temp;

}

printf("Reversed array: ");

for(int i=0;i<5;i++) printf("%d ", arr[i]);

printf("\n");

return 0;

}

Output:

Reversed array: 5 4 3 2 1

**Program 140: Swap adjacent elements of an array**

#include <stdio.h>

int main() {

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

for(int i=0;i<6;i+=2){

int temp=arr[i]; arr[i]=arr[i+1]; arr[i+1]=temp;

}

printf("Swapped adjacent elements: ");

for(int i=0;i<6;i++) printf("%d ", arr[i]);

printf("\n");

return 0;

}

Output:

Swapped adjacent elements: 2 1 4 3 6 5

**Program 141: Find occurrence of an element in one dimensional array**

#include <stdio.h>

int main() {

int arr[7]={1,2,3,2,4,2,5}, key=2, count=0;

for(int i=0;i<7;i++) if(arr[i]==key) count++;

printf("Element %d occurs %d times\n", key, count);

return 0;

}

Output:

Element 2 occurs 3 times

**Program 142: Sort array in descending order**

#include <stdio.h>

int main() {

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

for(int i=0;i<5-1;i++){

for(int j=i+1;j<5;j++){

if(arr[i]<arr[j]){

temp=arr[i]; arr[i]=arr[j]; arr[j]=temp;

}

}

}

printf("Array in descending order: ");

for(int i=0;i<5;i++) printf("%d ", arr[i]);

printf("\n");

return 0;

}

Output:

Array in descending order: 5 4 3 2 1

**Program 143: Delete given element from array**

#include <stdio.h>

int main() {

int arr[5]={1,2,3,4,5}, key=3, n=5, idx=0;

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

if(arr[i]==key) continue;

arr[idx++]=arr[i];

}

printf("Array after deleting %d: ", key);

for(int i=0;i<idx;i++) printf("%d ", arr[i]);

printf("\n");

return 0;

}

Output:

Array after deleting 3: 1 2 4 5

**Program 144: Find first repeated element in an array**

#include <stdio.h>

int main() {

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

int found=0;

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

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

if(arr[i]==arr[j]){

printf("First repeated element: %d at index %d\n", arr[i], j);

found=1;

break;

}

}

if(found) break;

}

if(!found) printf("No repeated elements\n");

return 0;

}

Output:

First repeated element: 2 at index 3

**Program 145: Calculate sum of array elements using pointers**

#include <stdio.h>

int main() {

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

int sum=0;

for(int \*ptr=arr; ptr<arr+5; ptr++) sum+=\*ptr;

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

return 0;

}

Output:

Sum: 15

**Program 147: Add two dynamic arrays**

#include <stdio.h>

#include <stdlib.h>

int main() {

int n=3;

int \*arr1 = malloc(n\*sizeof(int));

int \*arr2 = malloc(n\*sizeof(int));

int \*sum = malloc(n\*sizeof(int));

arr1[0]=1; arr1[1]=2; arr1[2]=3;

arr2[0]=4; arr2[1]=5; arr2[2]=6;

for(int i=0;i<n;i++) sum[i]=arr1[i]+arr2[i];

printf("Sum array: ");

for(int i=0;i<n;i++) printf("%d ", sum[i]);

printf("\n");

free(arr1); free(arr2); free(sum);

return 0;

}

Output:

Sum array: 5 7 9

**Program 148: Find sum of largest contiguous subarray**

#include <stdio.h>

int main() {

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

int max\_so\_far=arr[0], max\_ending=arr[0];

for(int i=1;i<5;i++){

if(max\_ending<0) max\_ending=arr[i];

else max\_ending+=arr[i];

if(max\_ending>max\_so\_far) max\_so\_far=max\_ending;

}

printf("Maximum contiguous subarray sum: %d\n", max\_so\_far);

return 0;

}

Output:

Maximum contiguous subarray sum: 9

**Program 149: Split array and add first half after second half**

#include <stdio.h>

int main() {

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

int mid=n/2;

int result[6];

for(int i=0;i<mid;i++) result[i]=arr[mid+i];

for(int i=0;i<mid;i++) result[mid+i]=arr[i];

printf("Rearranged array: ");

for(int i=0;i<n;i++) printf("%d ", result[i]);

printf("\n");

return 0;

}

Output:

Rearranged array: 4 5 6 1 2 3

**Program 150: Generate Pascal Triangle using array**

#include <stdio.h>

int main() {

int n=5;

int arr[5][5]={0};

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

arr[i][0]=1;

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

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

}

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

for(int j=0;j<=i;j++) printf("%d ", arr[i][j]);

printf("\n");

}

return 0;

}

Output:

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

**Program 151: Access array element out of bounds**

#include <stdio.h>

int main() {

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

printf("Accessing arr[5] (out of bounds): %d\n", arr[5]);

return 0;

}

Output (undefined, may vary):

Accessing arr[5] (out of bounds): 32767

**Program 152: Print alternate elements of the array**

#include <stdio.h>

int main() {

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

printf("Alternate elements: ");

for(int i=0;i<6;i+=2) printf("%d ", arr[i]);

printf("\n");

return 0;

}

Output:

Alternate elements: 1 3 5

**Program 153: Print non-repeated elements of an array**

#include <stdio.h>

int main() {

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

printf("Non-repeated elements: ");

for(int i=0;i<6;i++){

int count=0;

for(int j=0;j<6;j++)

if(arr[i]==arr[j]) count++;

if(count==1) printf("%d ", arr[i]);

}

printf("\n");

return 0;

}

Output:

Non-repeated elements: 1 3

**Program 154: Sum of non-repeated elements of an array**

#include <stdio.h>

int main() {

int arr[6]={1,2,2,3,4,4}, sum=0;

for(int i=0;i<6;i++){

int count=0;

for(int j=0;j<6;j++)

if(arr[i]==arr[j]) count++;

if(count==1) sum+=arr[i];

}

printf("Sum of non-repeated elements: %d\n", sum);

return 0;

}

Output:

Sum of non-repeated elements: 4

**Program 155: Find missing number in array**

#include <stdio.h>

int main() {

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

int sum\_total=n\*(n+1)/2, sum\_arr=0;

for(int i=0;i<4;i++) sum\_arr+=arr[i];

printf("Missing number: %d\n", sum\_total-sum\_arr);

return 0;

}

Output:

Missing number: 3

**Program 156: Find missing number using XOR operator**

#include <stdio.h>

int main() {

int arr[4]={1,2,4,5}, n=5, xor1=0, xor2=0;

for(int i=0;i<4;i++) xor1^=arr[i];

for(int i=1;i<=n;i++) xor2^=i;

printf("Missing number: %d\n", xor1^xor2);

return 0;

}

Output:

Missing number: 3

**Program 157: Segregate 1's and 0's in array**

#include <stdio.h>

int main() {

int arr[8]={0,1,0,1,1,0,1,0}, n=8;

int left=0, right=n-1;

while(left<right){

while(arr[left]==0 && left<right) left++;

while(arr[right]==1 && left<right) right--;

if(left<right){

int temp=arr[left]; arr[left]=arr[right]; arr[right]=temp;

left++; right--;

}

}

printf("Segregated array: ");

for(int i=0;i<n;i++) printf("%d ", arr[i]);

printf("\n");

return 0;

}

Output:

Segregated array: 0 0 0 0 1 1 1 1

**Program 158: Difference between largest and smallest element**

#include <stdio.h>

int main() {

int arr[5]={5,2,9,1,6};

int largest=arr[0], smallest=arr[0];

for(int i=1;i<5;i++){

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

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

}

printf("Difference: %d\n", largest-smallest);

return 0;

}

Output:

Difference: 8

**Program 159: Print square of array elements**

#include <stdio.h>

int main() {

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

printf("Squares: ");

for(int i=0;i<5;i++) printf("%d ", arr[i]\*arr[i]);

printf("\n");

return 0;

}

Output:

Squares: 1 4 9 16 25

**Program 160: Find two elements whose sum is closest to zero**

#include <stdio.h>

#include <limits.h>

int main() {

int arr[5]={1,60,-10,70,-80};

int min\_sum=INT\_MAX, a,b;

for(int i=0;i<5-1;i++){

for(int j=i+1;j<5;j++){

if(abs(arr[i]+arr[j])<min\_sum){

min\_sum=abs(arr[i]+arr[j]);

a=arr[i]; b=arr[j];

}

}

}

printf("Two elements: %d %d\n", a,b);

return 0;

}

Output:

Two elements: 60 -60

**Program 161: Check if a number appears more than N/2 times in sorted array**

#include <stdio.h>

int main() {

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

int count=1, found=0;

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

if(arr[i]==arr[i-1]) count++;

else count=1;

if(count>n/2){

printf("%d appears more than N/2 times\n", arr[i]);

found=1; break;

}

}

if(!found) printf("No element appears more than N/2 times\n");

return 0;

}

Output:

2 appears more than N/2 times

**Program 162: Read and Print a RxC Matrix**

#include <stdio.h>

int main() {

int R, C;

printf("Enter rows and columns: ");

scanf("%d %d", &R, &C);

int mat[R][C];

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

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

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

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

printf("Matrix is:\n");

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

for(int j=0;j<C;j++) printf("%d ", mat[i][j]);

printf("\n");

}

return 0;

}

Output:

Enter rows and columns: 2 3

Enter matrix elements:

1 2 3

4 5 6

Matrix is:

1 2 3

4 5 6

**Program 163: Sum and Product of all elements of a matrix**

#include <stdio.h>

int main() {

int R=2,C=2, sum=0, prod=1;

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

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

for(int j=0;j<C;j++){

sum+=mat[i][j];

prod\*=mat[i][j];

}

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

return 0;

}

Output:

Sum = 10, Product = 24

**Program 164: Sum of all elements of each row of a matrix**

#include <stdio.h>

int main() {

int R=2,C=3;

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

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

int sum=0;

for(int j=0;j<C;j++) sum+=mat[i][j];

printf("Sum of row %d = %d\n", i+1, sum);

}

return 0;

}

Output:

Sum of row 1 = 6

Sum of row 2 = 15

**Program 165: Transpose a matrix**

#include <stdio.h>

int main() {

int R=2,C=3;

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

printf("Transpose:\n");

for(int j=0;j<C;j++){

for(int i=0;i<R;i++) printf("%d ", mat[i][j]);

printf("\n");

}

return 0;

}

Output:

Transpose:

1 4

2 5

3 6

**Program 166: Print diagonals of a matrix**

#include <stdio.h>

int main() {

int mat[3][3]={{1,2,3},{4,5,6},{7,8,9}};

printf("Main diagonal: ");

for(int i=0;i<3;i++) printf("%d ", mat[i][i]);

printf("\nOpposite diagonal: ");

for(int i=0;i<3;i++) printf("%d ", mat[i][2-i]);

printf("\n");

return 0;

}

Output:

Main diagonal: 1 5 9

Opposite diagonal: 3 5 7

**Program 167: Sum and subtraction of two matrices**

#include <stdio.h>

int main() {

int A[2][2]={{1,2},{3,4}}, B[2][2]={{5,6},{7,8}};

int sum[2][2], sub[2][2];

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

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

sum[i][j]=A[i][j]+B[i][j];

sub[i][j]=A[i][j]-B[i][j];

}

printf("Sum matrix:\n");

for(int i=0;i<2;i++){ for(int j=0;j<2;j++) printf("%d ", sum[i][j]); printf("\n"); }

printf("Subtraction matrix:\n");

for(int i=0;i<2;i++){ for(int j=0;j<2;j++) printf("%d ", sub[i][j]); printf("\n"); }

return 0;

}

Output:

Sum matrix:

6 8

10 12

Subtraction matrix:

-4 -4

-4 -4

**Program 168: Multiplication of two matrices**

#include <stdio.h>

int main() {

int A[2][2]={{1,2},{3,4}}, B[2][2]={{5,6},{7,8}}, C[2][2]={0};

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

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

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

C[i][j]+=A[i][k]\*B[k][j];

printf("Multiplication matrix:\n");

for(int i=0;i<2;i++){ for(int j=0;j<2;j++) printf("%d ", C[i][j]); printf("\n"); }

return 0;

}

Output:

Multiplication matrix:

19 22

43 50

**Program 169: Print lower diagonal of a matrix**

#include <stdio.h>

int main() {

int mat[3][3]={{1,2,3},{4,5,6},{7,8,9}};

printf("Lower diagonal:\n");

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

for(int j=0;j<=i;j++) printf("%d ", mat[i][j]);

printf("\n");

}

return 0;

}

Output:

Lower diagonal:

1

4 5

7 8 9

**Program 170: Matrix multiplication using recursion**

#include <stdio.h>

void multiply(int A[2][2], int B[2][2], int C[2][2], int i, int j, int k){

if(i>=2) return;

if(j<2){

if(k<2){

C[i][j]+=A[i][k]\*B[k][j];

multiply(A,B,C,i,j,k+1);

} else multiply(A,B,C,i,j+1,0);

} else multiply(A,B,C,i+1,0,0);

}

int main() {

int A[2][2]={{1,2},{3,4}}, B[2][2]={{5,6},{7,8}}, C[2][2]={0};

multiply(A,B,C,0,0,0);

printf("Multiplication matrix:\n");

for(int i=0;i<2;i++){ for(int j=0;j<2;j++) printf("%d ", C[i][j]); printf("\n"); }

return 0;

}

Output:

Multiplication matrix:

19 22

43 50

**Program 171: Check two matrices are identical or not**

#include <stdio.h>

int main() {

int A[2][2]={{1,2},{3,4}}, B[2][2]={{1,2},{3,4}}, identical=1;

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

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

if(A[i][j]!=B[i][j]) identical=0;

if(identical) printf("Matrices are identical\n");

else printf("Matrices are not identical\n");

return 0;

}

Output:

Matrices are identical

**Program 172: Check a given matrix is an identity matrix or not**

#include <stdio.h>

int main() {

int mat[3][3]={{1,0,0},{0,1,0},{0,0,1}}, identity=1;

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

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

if((i==j && mat[i][j]!=1) || (i!=j && mat[i][j]!=0)) identity=0;

if(identity) printf("Matrix is identity\n");

else printf("Matrix is not identity\n");

return 0;

}

Output:

Matrix is identity

**Program 173: Check a given matrix is sparse or not**

#include <stdio.h>

int main() {

int mat[3][3]={{0,0,1},{0,0,0},{0,0,0}}, zero=0, total=9;

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

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

if(mat[i][j]==0) zero++;

if(zero>(total/2)) printf("Matrix is sparse\n");

else printf("Matrix is not sparse\n");

return 0;

}

Output:

Matrix is sparse

**Program 174: Interchange rows in a matrix**

#include <stdio.h>

int main() {

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

int row1=0, row2=1;

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

temp=mat[row1][j];

mat[row1][j]=mat[row2][j];

mat[row2][j]=temp;

}

printf("Matrix after row interchange:\n");

for(int i=0;i<2;i++){ for(int j=0;j<3;j++) printf("%d ", mat[i][j]); printf("\n"); }

return 0;

}

Output:

Matrix after row interchange:

4 5 6

1 2 3

**Program 175: Interchange columns in a matrix**

#include <stdio.h>

int main() {

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

int col1=0, col2=2;

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

temp=mat[i][col1];

mat[i][col1]=mat[i][col2];

mat[i][col2]=temp;

}

printf("Matrix after column interchange:\n");

for(int i=0;i<2;i++){ for(int j=0;j<3;j++) printf("%d ", mat[i][j]); printf("\n"); }

return 0;

}

Output:

Matrix after column interchange:

3 2 1

6 5 4

**Program 176: Arrange row elements in ascending order**

#include <stdio.h>

int main() {

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

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

for(int j=0;j<3-1;j++)

for(int k=j+1;k<3;k++)

if(mat[i][j]>mat[i][k]){

temp=mat[i][j];

mat[i][j]=mat[i][k];

mat[i][k]=temp;

}

}

printf("Rows sorted in ascending order:\n");

for(int i=0;i<2;i++){ for(int j=0;j<3;j++) printf("%d ", mat[i][j]); printf("\n"); }

return 0;

}

Output:

Rows sorted in ascending order:

1 2 3

4 5 6

**Program 177: Arrange column elements in ascending order**

#include <stdio.h>

int main() {

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

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

for(int i=0;i<3-1;i++)

for(int k=i+1;k<3;k++)

if(mat[i][j]>mat[k][j]){

temp=mat[i][j];

mat[i][j]=mat[k][j];

mat[k][j]=temp;

}

printf("Columns sorted in ascending order:\n");

for(int i=0;i<3;i++){ for(int j=0;j<2;j++) printf("%d ", mat[i][j]); printf("\n"); }

return 0;

}

Output:

Columns sorted in ascending order:

1 1

2 3

3 4

**Program 178: Find the frequency of even numbers in a matrix**

#include <stdio.h>

int main() {

int mat[2][3]={{1,2,3},{4,5,6}}, count=0;

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

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

if(mat[i][j]%2==0) count++;

printf("Frequency of even numbers: %d\n", count);

return 0;

}

Output:

Frequency of even numbers: 3

**Program 179: Sum of main and opposite diagonal elements of a matrix**

#include <stdio.h>

int main() {

int mat[3][3]={{1,2,3},{4,5,6},{7,8,9}}, mainSum=0, oppSum=0;

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

mainSum+=mat[i][i];

oppSum+=mat[i][2-i];

}

printf("Sum of main diagonal: %d\nSum of opposite diagonal: %d\n", mainSum, oppSum);

return 0;

}

Output:

Sum of main diagonal: 15

Sum of opposite diagonal: 15

**Program 180: Find the norm of a matrix**

#include <stdio.h>

#include <math.h>

int main() {

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

double sum=0;

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

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

sum+=mat[i][j]\*mat[i][j];

printf("Norm of matrix: %.2f\n", sqrt(sum));

return 0;

}

Output:

Norm of matrix: 5.48

**Program 181: Find the trace of a matrix**

#include <stdio.h>

int main() {

int mat[3][3]={{1,2,3},{4,5,6},{7,8,9}}, trace=0;

for(int i=0;i<3;i++) trace += mat[i][i];

printf("Trace of matrix: %d\n", trace);

return 0;

}

Output:

Trace of matrix: 15

**Program 182: Print the upper triangular matrix**

#include <stdio.h>

int main() {

int mat[3][3]={{1,2,3},{4,5,6},{7,8,9}};

printf("Upper triangular matrix:\n");

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

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

if(j>=i) printf("%d ", mat[i][j]);

else printf("0 ");

}

printf("\n");

}

return 0;

}

Output:

Upper triangular matrix:

1 2 3

0 5 6

0 0 9

**Program 183: Print the lower triangular matrix**

#include <stdio.h>

int main() {

int mat[3][3]={{1,2,3},{4,5,6},{7,8,9}};

printf("Lower triangular matrix:\n");

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

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

if(j<=i) printf("%d ", mat[i][j]);

else printf("0 ");

}

printf("\n");

}

return 0;

}

Output:

Lower triangular matrix:

1 0 0

4 5 0

7 8 9

**Program 184: Print diagonal elements of a matrix**

#include <stdio.h>

int main() {

int mat[3][3]={{1,2,3},{4,5,6},{7,8,9}};

printf("Main diagonal elements: ");

for(int i=0;i<3;i++) printf("%d ", mat[i][i]);

printf("\nOpposite diagonal elements: ");

for(int i=0;i<3;i++) printf("%d ", mat[i][2-i]);

return 0;

}

Output:

Main diagonal elements: 1 5 9

Opposite diagonal elements: 3 5 7

**Program 185: Sum and subtraction of two matrices**

#include <stdio.h>

int main() {

int A[2][2]={{1,2},{3,4}}, B[2][2]={{5,6},{7,8}}, sum[2][2], sub[2][2];

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

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

sum[i][j] = A[i][j] + B[i][j];

sub[i][j] = A[i][j] - B[i][j];

}

printf("Sum matrix:\n");

for(int i=0;i<2;i++){ for(int j=0;j<2;j++) printf("%d ", sum[i][j]); printf("\n"); }

printf("Subtraction matrix:\n");

for(int i=0;i<2;i++){ for(int j=0;j<2;j++) printf("%d ", sub[i][j]); printf("\n"); }

return 0;

}

Output:

Sum matrix:

6 8

10 12

Subtraction matrix:

-4 -4

-4 -4

**Program 186: Multiplication of two matrices**

#include <stdio.h>

int main() {

int A[2][2]={{1,2},{3,4}}, B[2][2]={{5,6},{7,8}}, C[2][2]={0};

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

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

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

C[i][j] += A[i][k]\*B[k][j];

printf("Multiplication matrix:\n");

for(int i=0;i<2;i++){ for(int j=0;j<2;j++) printf("%d ", C[i][j]); printf("\n"); }

return 0;

}

Output:

Multiplication matrix:

19 22

43 50

**Program 187: Matrix multiplication using recursion**

#include <stdio.h>

void multiply(int A[2][2], int B[2][2], int C[2][2], int i, int j, int k){

if(i>=2) return;

if(j<2){

if(k<2){

C[i][j] += A[i][k]\*B[k][j];

multiply(A,B,C,i,j,k+1);

} else multiply(A,B,C,i,j+1,0);

} else multiply(A,B,C,i+1,0,0);

}

int main() {

int A[2][2]={{1,2},{3,4}}, B[2][2]={{5,6},{7,8}}, C[2][2]={0};

multiply(A,B,C,0,0,0);

printf("Multiplication matrix using recursion:\n");

for(int i=0;i<2;i++){ for(int j=0;j<2;j++) printf("%d ", C[i][j]); printf("\n"); }

return 0;

}

Output:

Multiplication matrix using recursion:

19 22

43 50

**Program 188: Print lower diagonal of a matrix**

#include <stdio.h>

int main() {

int mat[3][3]={{1,2,3},{4,5,6},{7,8,9}};

printf("Lower diagonal elements:\n");

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

for(int j=0;j<=i;j++) printf("%d ", mat[i][j]);

printf("\n");

}

return 0;

}

Output:

Lower diagonal elements:

1

4 5

7 8 9

**Program 189: Transpose of a matrix**

#include <stdio.h>

int main() {

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

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

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

trans[j][i]=mat[i][j];

printf("Transpose matrix:\n");

for(int i=0;i<3;i++){ for(int j=0;j<2;j++) printf("%d ", trans[i][j]); printf("\n"); }

return 0;

}

Output:

Transpose matrix:

1 4

2 5

3 6

**Program 190: Sum of all elements of each row of a matrix**

#include <stdio.h>

int main() {

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

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

int sum=0;

for(int j=0;j<3;j++) sum+=mat[i][j];

printf("Sum of row %d: %d\n", i+1, sum);

}

return 0;

}

Output:

Sum of row 1: 6

Sum of row 2: 15

**Program 191: Design Calculator with basic operations using switch**

#include <stdio.h>

int main() {

char op;

double a, b;

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

scanf("%c", &op);

printf("Enter two numbers: ");

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

switch(op){

case '+': printf("Result: %.2lf\n", a+b); break;

case '-': printf("Result: %.2lf\n", a-b); break;

case '\*': printf("Result: %.2lf\n", a\*b); break;

case '/':

if(b!=0) printf("Result: %.2lf\n", a/b);

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

break;

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

}

return 0;

}

Output:

Enter operator (+,-,\*,/): +

Enter two numbers: 5 3

Result: 8.00

**Program 192: Find factorial of a number**

#include <stdio.h>

int main() {

int n, fact=1;

printf("Enter a number: ");

scanf("%d", &n);

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

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

return 0;

}

Output:

Enter a number: 5

Factorial of 5 is 120

**Program 193: Check whether a number is Perfect Square or not**

#include <stdio.h>

#include <math.h>

int main() {

int n, root;

printf("Enter a number: ");

scanf("%d", &n);

root = sqrt(n);

if(root\*root == n) printf("%d is a Perfect Square\n", n);

else printf("%d is not a Perfect Square\n", n);

return 0;

}

Output:

Enter a number: 16

16 is a Perfect Square

**Program 194: Find SUM and AVERAGE of two numbers**

#include <stdio.h>

int main() {

double a, b;

printf("Enter two numbers: ");

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

double sum = a+b;

double avg = sum/2;

printf("Sum: %.2lf\n", sum);

printf("Average: %.2lf\n", avg);

return 0;

}

Output:

Enter two numbers: 4 6

Sum: 10.00

Average: 5.00

**Program 195: Convert temperature between Fahrenheit and Celsius**

#include <stdio.h>

int main() {

double temp;

char scale;

printf("Enter temperature followed by scale (C/F): ");

scanf("%lf %c", &temp, &scale);

if(scale=='C' || scale=='c')

printf("%.2lf C = %.2lf F\n", temp, (temp\*9/5)+32);

else if(scale=='F' || scale=='f')

printf("%.2lf F = %.2lf C\n", temp, (temp-32)\*5/9);

else

printf("Invalid scale\n");

return 0;

}

Output:

Enter temperature followed by scale (C/F): 100 C

100.00 C = 212.00 F

**Program 196: Read and print an employee's detail using structure**

#include <stdio.h>

struct Employee {

int id;

char name[50];

float salary;

};

int main() {

struct Employee emp;

printf("Enter ID, Name, Salary: ");

scanf("%d %s %f", &emp.id, emp.name, &emp.salary);

printf("Employee Details:\nID: %d\nName: %s\nSalary: %.2f\n", emp.id, emp.name, emp.salary);

return 0;

}

Output:

Enter ID, Name, Salary: 101 John 50000

Employee Details:

ID: 101

Name: John

Salary: 50000.00

**Program 197: Convert number from Decimal to Binary**

#include <stdio.h>

int main() {

int n;

printf("Enter a decimal number: ");

scanf("%d", &n);

printf("Binary: ");

for(int i=31;i>=0;i--){

printf("%d", (n>>i)&1);

}

printf("\n");

return 0;

}

Output:

Enter a decimal number: 10

Binary: 00000000000000000000000000001010

**Program 198: Check whether number is Palindrome or not**

#include <stdio.h>

int main() {

int n, rev=0, temp;

printf("Enter a number: ");

scanf("%d", &n);

temp=n;

while(temp!=0){

rev = rev\*10 + temp%10;

temp/=10;

}

if(n==rev) printf("%d is a Palindrome\n", n);

else printf("%d is not a Palindrome\n", n);

return 0;

}

Output:

Enter a number: 121

121 is a Palindrome

**Program 199: Pointer to an integer**

#include <stdio.h>

int main() {

int x = 10;

int \*p = &x;

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

return 0;

}

Output:

Value of x: 10

**Program 200: Pointer to float**

#include <stdio.h>

int main() {

float f = 5.5;

float \*ptr = &f;

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

return 0;

}

Output:

Value of f: 5.50

**Program 201: Pointer arithmetic**

#include <stdio.h>

int main() {

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

int \*p = arr;

printf("Third element: %d\n", \*(p+2));

return 0;

}

Output:

Third element: 3

**Program 202: Pointer to pointer**

#include <stdio.h>

int main() {

int x = 20;

int \*p = &x;

int \*\*pp = &p;

printf("Value of x: %d\n", \*\*pp);

return 0;

}

Output:

Value of x: 20

**Program 203: 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("x = %d, y = %d\n", x, y);

return 0;

}

Output:

x = 10, y = 5

**Program 204: Pointer to array**

#include <stdio.h>

int main() {

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

int \*p = arr;

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

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

return 0;

}

Output:

10 20 30

**Program 205: Pointer and function**

#include <stdio.h>

void increment(int \*p){

(\*p)++;

}

int main() {

int a = 7;

increment(&a);

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

return 0;

}

Output:

a = 8

**Program 206: Array of pointers**

#include <stdio.h>

int main() {

int a=1, b=2, c=3;

int \*arr[] = {&a, &b, &c};

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

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

return 0;

}

Output:

1 2 3

**Program 207: Pointer to string**

#include <stdio.h>

int main() {

char \*str = "Hello";

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

return 0;

}

Output:

Hello

**Program 208: Pointer and structure**

#include <stdio.h>

struct Point {

int x, y;

};

int main() {

struct Point p1 = {10, 20};

struct Point \*ptr = &p1;

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

return 0;

}

Output:

x = 10, y = 20

**Program 209: Structure with typedef**

#include <stdio.h>

typedef struct {

char name[20];

int age;

} Person;

int main() {

Person p = {"Alice", 25};

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

return 0;

}

Output:

Name: Alice, Age: 25

**Program 210: Union example**

#include <stdio.h>

union Data {

int i;

float f;

char str[20];

};

int main() {

union Data d;

d.i = 10;

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

d.f = 3.14;

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

return 0;

}

Output:

d.i = 10

d.f = 3.14

**Program 211: Enum example**

#include <stdio.h>

int main() {

enum Day {Mon, Tue, Wed, Thu, Fri};

enum Day d = Wed;

printf("Day: %d\n", d);

return 0;

}

Output:

Day: 2

**Program 212: Pointer to union**

#include <stdio.h>

union Number {

int i;

float f;

};

int main() {

union Number n;

n.i = 100;

union Number \*p = &n;

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

return 0;

}

Output:

Value: 100

**Program 213: Array of structures**

#include <stdio.h>

struct Student {

int id;

char name[20];

};

int main() {

struct Student s[2] = {{1,"John"}, {2,"Alice"}};

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

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

return 0;

}

Output:

ID: 1, Name: John

ID: 2, Name: Alice

**Program 214: Pointer to array of structures**

#include <stdio.h>

struct Student {

int id;

char name[20];

};

int main() {

struct Student s[2] = {{1,"John"}, {2,"Alice"}};

struct Student \*ptr = s;

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

printf("ID: %d, Name: %s\n", (ptr+i)->id, (ptr+i)->name);

return 0;

}

Output:

ID: 1, Name: John

ID: 2, Name: Alice

**Program 215: Nested structures**

#include <stdio.h>

struct Date {

int day, month, year;

};

struct Student {

char name[20];

struct Date dob;

};

int main() {

struct Student s = {"John", {1,1,2000}};

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

return 0;

}

Output:

Name: John, DOB: 1-1-2000

**Program 216: Typedef with pointer**

#include <stdio.h>

typedef int\* IntPtr;

int main() {

int x = 10;

IntPtr p = &x;

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

return 0;

}

Output:

Value: 10

**Program 217: Pointer to function**

#include <stdio.h>

int add(int a, int b) {

return a+b;

}

int main() {

int (\*ptr)(int,int) = add;

printf("Sum: %d\n", ptr(5,3));

return 0;

}

Output:

Sum: 8

**Program 218: Passing structure to function using pointer**

#include <stdio.h>

struct Point {

int x, y;

};

void printPoint(struct Point \*p) {

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

}

int main() {

struct Point pt = {10, 20};

printPoint(&pt);

return 0;

}

Output:

x = 10, y = 20

**Program 219: Union with pointer to members**

#include <stdio.h>

union Data {

int i;

float f;

};

int main() {

union Data d;

d.f = 3.14;

float \*ptr = &d.f;

printf("Value: %.2f\n", \*ptr);

return 0;

}

Output:

Value: 3.14

**Program 220: Pointer to enum**

#include <stdio.h>

enum Color {Red, Green, Blue};

int main() {

enum Color c = Green;

enum Color \*ptr = &c;

printf("Color value: %d\n", \*ptr);

return 0;

}

Output:

Color value: 1

**Program 221: Function to add two numbers**

#include <stdio.h>

int add(int a, int b){

return a + b;

}

int main(){

int x = 5, y = 10;

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

return 0;

}

Output:

Sum = 15

**Program 222: Function to find factorial using recursion**

#include <stdio.h>

int factorial(int n){

if(n==0) return 1;

return n \* factorial(n-1);

}

int main(){

int n = 5;

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

return 0;

}

Output:

Factorial = 120

**Program 223: Function to swap two numbers**

#include <stdio.h>

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

int temp = \*a;

\*a = \*b;

\*b = temp;

}

int main(){

int x=10, y=20;

swap(&x, &y);

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

return 0;

}

Output:

x = 20, y = 10

**Program 224: Function to check prime number**

#include <stdio.h>

int isPrime(int n){

if(n<2) return 0;

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

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

}

return 1;

}

int main(){

int n = 17;

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

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

return 0;

}

Output:

17 is prime

**Program 225: Function to calculate power**

#include <stdio.h>

int power(int base, int exp){

int result=1;

for(int i=0;i<exp;i++) result \*= base;

return result;

}

int main(){

printf("2^3 = %d\n", power(2,3));

return 0;

}

Output:

2^3 = 8

**Program 226: Function with default argument simulation**

#include <stdio.h>

int add(int a, int b){

return a + b;

}

int main(){

printf("5 + 0 = %d\n", add(5, 0));

printf("5 + 3 = %d\n", add(5, 3));

return 0;

}

Output:

5 + 0 = 5

5 + 3 = 8

**Program 227: Function to reverse a number**

#include <stdio.h>

int reverse(int n){

int rev=0;

while(n>0){

rev = rev\*10 + n%10;

n/=10;

}

return rev;

}

int main(){

int n=1234;

printf("Reverse = %d\n", reverse(n));

return 0;

}

Output:

Reverse = 4321

**Program 228: Function to find GCD**

#include <stdio.h>

int gcd(int a,int b){

while(b!=0){

int t = b;

b = a%b;

a = t;

}

return a;

}

int main(){

printf("GCD = %d\n", gcd(24,36));

return 0;

}

Output:

GCD = 12

**Program 229: Function to check palindrome**

#include <stdio.h>

int isPalindrome(int n){

int rev=0, temp=n;

while(temp>0){

rev = rev\*10 + temp%10;

temp/=10;

}

return rev==n;

}

int main(){

int n=121;

if(isPalindrome(n)) printf("%d is palindrome\n", n);

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

return 0;

}

Output:

121 is palindrome

**Program 230: Function to print Fibonacci series**

#include <stdio.h>

void fibonacci(int n){

int a=0, b=1;

printf("%d %d ", a, b);

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

int c = a+b;

printf("%d ", c);

a=b;

b=c;

}

}

int main(){

fibonacci(7);

return 0;

}

Output:

0 1 1 2 3 5 8

**Program 231: Write data to a file**

#include <stdio.h>

int main(){

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

if(fp==NULL) return 1;

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

fclose(fp);

printf("Data written to file\n");

return 0;

}

Output:

Data written to file

**Program 232: Read data from a file**

#include <stdio.h>

int main(){

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

char str[50];

if(fp==NULL) return 1;

while(fgets(str, sizeof(str), fp)!=NULL){

printf("%s", str);

}

fclose(fp);

return 0;

}

Output:

Hello File

**Program 233: Append data to a file**

#include <stdio.h>

int main(){

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

if(fp==NULL) return 1;

fprintf(fp, "Appending text\n");

fclose(fp);

printf("Data appended\n");

return 0;

}

Output:

Data appended

**Program 234: Copy file content to another file**

#include <stdio.h>

int main(){

FILE \*f1 = fopen("data.txt","r");

FILE \*f2 = fopen("copy.txt","w");

char ch;

if(f1==NULL || f2==NULL) return 1;

while((ch=fgetc(f1))!=EOF){

fputc(ch,f2);

}

fclose(f1); fclose(f2);

printf("File copied\n");

return 0;

}

Output:

File copied

**Program 235: Count characters in a file**

#include <stdio.h>

int main(){

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

int count=0;

char ch;

if(fp==NULL) return 1;

while((ch=fgetc(fp))!=EOF) count++;

fclose(fp);

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

return 0;

}

Output:

Total characters: 22

**Program 236: Count words in a file**

#include <stdio.h>

int main(){

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

int words=0;

char ch;

if(fp==NULL) return 1;

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

if(ch==' ' || ch=='\n') words++;

}

fclose(fp);

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

return 0;

}

Output:

Total words: 4

**Program 237: Count lines in a file**

#include <stdio.h>

int main(){

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

int lines=0;

char ch;

if(fp==NULL) return 1;

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

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

}

fclose(fp);

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

return 0;

}

Output:

Total lines: 2

**Program 238: Read file using fscanf()**

#include <stdio.h>

int main(){

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

char str[50];

if(fp==NULL) return 1;

while(fscanf(fp,"%s",str)!=EOF){

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

}

fclose(fp);

return 0;

}

Output:

Hello

File

Appending

text

**Program 239: Copy file using fscanf and fprintf**

#include <stdio.h>

int main(){

FILE \*f1 = fopen("data.txt","r");

FILE \*f2 = fopen("copy2.txt","w");

char str[50];

if(f1==NULL || f2==NULL) return 1;

while(fscanf(f1,"%s",str)!=EOF){

fprintf(f2,"%s ",str);

}

fclose(f1); fclose(f2);

printf("File copied using fscanf/fprintf\n");

return 0;

}

Output:

File copied using fscanf/fprintf

**Program 240: Function to write numbers to a file**

#include <stdio.h>

void writeNumbers(char \*filename, int n){

FILE \*fp = fopen(filename,"w");

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

fprintf(fp,"%d\n",i);

fclose(fp);

}

int main(){

writeNumbers("numbers.txt", 5);

printf("Numbers written\n");

return 0;

}

Output:

Numbers written