# ✅ Day : Basic Input/Output and Operators (2-8-2025)

1. Write a C program to add two integers.

Input : Two integers (a, b)

Process : Add the two integers: sum = a + b

Output : Display the sum of a + b

Program:

#include <stdio.h>

void main()

{

int a, b, sum;

scanf("%d", &a);

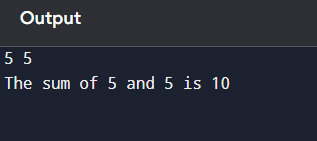
scanf("%d", &b);

sum = a + b;

printf("The sum of %d and %d is %d\n", a, b, sum);

}

Output :



1. Write a program to swap two numbers using a temporary variable.

Input : The program takes two integer values as input from the user, stored in variables a and b.

Process ; swap the values of a and b, a temporary variable temp is used. First, the value of a is assigned to temp. Then, the value of b is assigned to a, and finally, the value stored in temp (original a) is assigned to b.

Output : The program displays the values of a and b after swapping.

Program :

#include <stdio.h>

void main()

{

int a, b, temp;

scanf("%d", &a);

scanf("%d", &b);

temp = a;

a = b;

b = temp;

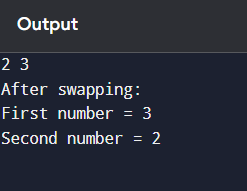
printf("After swapping:\n");

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

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

}

Output :



.

1. Write a program to find the ASCII value of a character.

Input : The program takes two integer values from the user and stores them in variables a and b.

Process : To swap the values without using a temporary variable, arithmetic operations are used:

* First, a is assigned the sum of a and b.
* Then, b is assigned the difference of the new a and original b (which gives original a).
* Finally, a is assigned the difference of the new a and new b (which gives original b).

Output : The program displays the values of a and b after the swap.

Program :

#include <stdio.h>

void main()

{

int a, b;

scanf("%d", &a);

scanf("%d", &b);

a = a + b;

b = a - b;

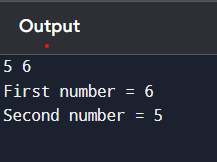
a = a - b;

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

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

}

Output :



1. Write a program to calculate the area and perimeter of a rectangle.

Input : The program accepts the **length** and **breadth** of a rectangle from the user.

Process : t calculates the **area** using the formula area = length × breadth and the **perimeter** using the formula perimeter = 2 × (length + breadth).

Output : The program displays the **area** and **perimeter** of the rectangle.

Program :

#include <stdio.h>

void main()

{

int length, breadth, area, perimeter;

scanf("%d", &length);

scanf("%d", &breadth);

area = length \* breadth;

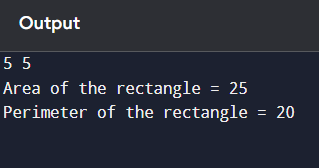
perimeter = 2 \* (length + breadth);

printf("Area of the rectangle = %d\n", area);

printf("Perimeter of the rectangle = %d\n", perimeter);

}

Output :



1. Write a program to compute the simple interest.

Input : The program takes the **principal amount (P)**, **rate of interest (R)**, and **time (T)** in years as input from the user.

Process : It calculates **Simple Interest** using the formula:  
Simple Interest = (P × R × T) / 100.

Output : The calculated **Simple Interest** is displayed to the user.

Program :

#include <stdio.h>

void main()

{

float principal, rate, time, interest;

scanf("%f", &principal);

scanf("%f", &rate);

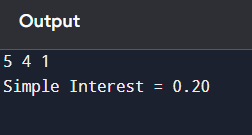
scanf("%f", &time);

interest = (principal \* rate \* time) / 100;

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

}

Output :



1. Write a program to convert temperature from Celsius to Fahrenheit.

Input : The program accepts the **temperature in Celsius** from the user.

Process : It converts the temperature to Fahrenheit using the formula:  
Fahrenheit = (Celsius × 9/5) + 32.

Output : The program displays the **temperature in Fahrenheit**.

Program :

#include <stdio.h>

void main()

{

float celsius, fahrenheit;

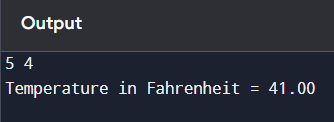
scanf("%f", &celsius);

fahrenheit = (celsius \* 9 / 5) + 32;

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

}

Output :



1. Write a program to find the quotient and remainder of two integers.

Input : The program accepts two integer values from the user — the **dividend** and the **divisor**.

Process : It calculates the **quotient** using the division operator (/) and the **remainder** using the modulus operator (%).

Output : The program displays the **quotient** and **remainder**.

Program :

#include <stdio.h>

void main()

{

int dividend, divisor, quotient, remainder;

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

quotient = dividend / divisor;

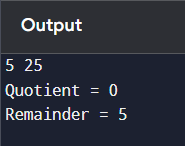
remainder = dividend % divisor;

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

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

}

Output :



1. Write a program to check whether a number is even or odd.

Input : The program accepts an integer number from the user.

Process : The program checks the **remainder** when the number is divided by 2 using the modulus operator (%).

Output : The program displays whether the number is **even** or **odd**.

Program :

#include <stdio.h>

int main() {

int number;

// Input

printf("Enter an integer: ");

scanf("%d", &number);

// Process & Output

if (number % 2 == 0) {

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

} else {

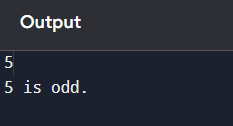
printf("%d is odd.\n", number);

}

return 0;

}

Output :



1. Write a program to calculate the square and cube of a number.

Input : The program takes a single integer number from the user.

Process : It calculates the **square** by multiplying the number by itself (number × number), and the **cube** by multiplying the number three times (number × number × number).

**Output : The program displays the square and cube of the entered number.**

Program :

#include <stdio.h>

void main()

{

int number, square, cube;

scanf("%d", &number);

square = number \* number;

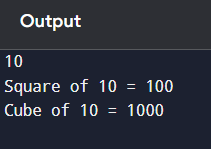
cube = number \* number \* number;

printf("Square of %d = %d\n", number, square);

printf("Cube of %d = %d\n", number, cube);

}

Output :



1. Write a program to swap number using two numbers without using a temporary varaiable

Input : The program takes two integer values from the user, stored in variables a and b.

Process : It swaps the values of a and b using arithmetic operations, without using a third (temporary) variable.

Output : The program displays the values of a and b after the swap.

Program :

#include <stdio.h>

void main()

{

int a, b;

scanf("%d", &a);

scanf("%d", &b);

a = a + b;

b = a - b;

a = a - b;

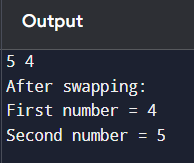
printf("After swapping:\n");

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

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

}

Output :



# ✅ Day : Conditional Statements (4-8-2025)

1. Write a program to check if a number is positive, negative, or zero.
2. Write a program to find the largest among three numbers.
3. Write a program to check if a year is a leap year.
4. Write a program to check whether a character is a vowel or consonant.
5. Write a program to assign grades based on marks.
6. Write a program to check whether a number is divisible by 5 and 11.
7. Write a program to find the absolute value of a number.
8. Write a menu-driven program to perform +, -, \*, / operations.
9. Write a program to find roots of a quadratic equation.
10. Write a program to find the number of digits in a number.

# ✅ Day : Loops and Iterations (5-8-2025)

1. Write a program to print numbers from 1 to 100.
2. Write a program to print even numbers from 1 to 50.
3. Write a program to find the factorial of a number.
4. Write a program to calculate the sum of digits of a number.
5. Write a program to reverse a number.
6. Write a program to check whether a number is a palindrome.
7. Write a program to print multiplication table of a number.
8. Write a program to count the number of digits in a number.
9. Write a program to print the Fibonacci series up to n terms.
10. Write a program to calculate the sum of the first n natural numbers.

# ✅ Day : Arrays (6-8-2025)

1. Write a program to read and print elements of an array.
2. Write a program to find the sum of elements of an array.
3. Write a program to find the maximum and minimum element in an array.
4. Write a program to reverse an array.
5. Write a program to search for an element in an array (linear search).
6. Write a program to sort an array in ascending order.
7. Write a program to insert an element in an array.
8. Write a program to delete an element from an array.
9. Write a program to find the frequency of elements in an array.
10. Write a program to merge two arrays.

# ✅ Day : Strings (7-8-2025)

1. Write a program to find the length of a string without using strlen().
2. Write a program to copy one string to another.
3. Write a program to concatenate two strings.
4. Write a program to compare two strings.
5. Write a program to count vowels and consonants in a string.
6. Write a program to convert lowercase to uppercase and vice versa.
7. Write a program to check if a string is palindrome.
8. Write a program to reverse a string.
9. Write a program to count words in a string.
10. Write a program to find the frequency of each character in a string.

# ✅ Day : Functions (8-8-2025)

1. Write a function to find the factorial of a number.
2. Write a function to check whether a number is prime.
3. Write a function to calculate power using recursion.
4. Write a function to check palindrome number using recursion.
5. Write a function to calculate nCr (combinations).
6. Write a program to demonstrate call by value and call by reference.
7. Write a program using function to swap two numbers.
8. Write a recursive function to find the nth Fibonacci number.
9. Write a program to find GCD and LCM using functions.
10. Write a program to demonstrate global and local variables.

# ✅ Day : Pointers (9-8-2025)

1. Write a program to print the address of a variable using pointer.
2. Write a program to access array elements using pointers.
3. Write a program to swap two numbers using pointers.
4. Write a program to add two numbers using pointers.
5. Write a program to find the length of a string using pointers.
6. Write a program to reverse a string using pointers.
7. Write a program to count vowels using pointer.
8. Write a program to demonstrate pointer to pointer.
9. Write a program to allocate memory using malloc() and free it.
10. Write a program to sort an array using pointer notation.

# ✅ Day : Structures and Unions (11-8-2025)

1. Define a structure for student record and print details.
2. Write a program to store and display employee details using structures.
3. Write a program to pass a structure to a function.
4. Write a program to store multiple student records using array of structures.
5. Write a program to demonstrate nested structures.
6. Write a program to calculate total and average marks using structures.
7. Write a program to find the highest marks among students.
8. Write a program to sort student records by name using structure.
9. Write a program using union to store data of different types.
10. Compare and contrast structure vs union with a sample program.

# ✅ Day : File Handling (12-8-2025)

1. Write a program to create and write to a text file.
2. Write a program to read contents of a file and display.
3. Write a program to count number of lines in a file.
4. Write a program to copy contents from one file to another.
5. Write a program to append text to a file.
6. Write a program to count vowels in a file.
7. Write a program to read integers from a file and find the sum.
8. Write a program to read a structure from a file.
9. Write a program to sort names stored in a file.
10. Write a program to search for a word in a file.