**Practical File**

of

**Fundamentals of C Programming**

**(24CSE0107)**

**Batch-2024**

**Bachelor of Engineering (CSE)**

**A black text on a white background

Description automatically generated**

|  |  |
| --- | --- |
| **Submitted By**  **Shifa**  Roll No. 2410991672  Group No. 21, Sem: 2nd, BE (CSE)  Chitkara University, Punjab, 140401  India | **Submitted To**  **Dr. Sangeetha Annam**  Professor, CSE  Chitkara University, Punjab, 140401  India |

**Department of**

**Computer Science and Engineering,**

**Chitkara University School of Engineering and Technology,**

**Chitkara University, Punjab, India**

**Index**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Experiment** | **Page No.** |
| 1 | Install C compiler (GCC/Code::Blocks), set up IDE, compile and run the first "Hello, World!" program. | 3 |
| 2 | Write a Program to show the use to input (Scanf)/output (Printf) statements and block structure of C-program by highlighting the features of "stdio.h". | 4 |
| 3 | Write a program to add two numbers and display the sum. | 5 |
| 4 | Write a program to calculate the area and the circumference of a circle by using radius as the input provided by the user. | 6 |
| 5 | Write a Program to perform addition, subtraction, division and multiplication of two numbers given as input by the user. | 7 |
| 6 | Write a program to evaluate each of the following equations. (i) V = u + at. (ii) S = ut+1/2at2 (iii) T=2\*a+√b+9c (iv) H=√b2+p2 | 8 |
| 7 | Write a program to swap two variables:  a) By using temporary variable.  b) Without using temporary variable | 9-10 |
| 8 | Write a Program to find the greatest among three numbers using:   * Conditional Operator * If-Else statement | 11-12 |
| 9 | Write the following programs using switch case statement:   * To check that an input alphabet is vowel or consonant * To check whether a number is positive, negative or zero | 13-15 |
| 10 | Write a program using while loop to print the sum of first n natural numbers. | 16 |
| 11 | Write a program to check a number is Armstrong or not using For loop. | 17 |
| 12 | Write the program to count the digits in a number and then print the reverse of the number also. | 18-19 |
| 13 | Write a program to generate the Fibonacci series. | 20 |
| 14 | Write a program to print the following patterns:  a)  \*  \* \*  \* \* \*  \* \* \* \*  \* \* \* \* \*  \* \* \* \* \* \*  b)  \*  \* \*  \* \* \*  \* \* \* \*  \* \* \* \* \*  \* \* \* \* \* \* | 21-24 |
| 15 | Write a program to calculate the area and circumference of a circle using functions. | 25-26 |

|  |  |  |
| --- | --- | --- |
| 16. | Write a program to check that the given number is prime, Armstrong or perfect using the concept of functions. | 29-30 |
| 17. | Write a program to calculate the area and circumference of a circle using functions. | 31 |
| 18. | Write a program to swap two variables using the concept of call by value and call by reference. | 32 |
| 19. | Write a program to perform the following operations on 1D-Array:   * Insert * Update * Delete * Display * Search | 33-34 |
| 20. | Write a program to calculate the sum of array elements by passing it to a function. | 35 |
| 21. | Write a program to show the use of passing pointer as arguments to the functions. | 36 |
| 22. | Write a program matrix multiplication using the concept of 2D array | 37 |
| 23. | Write a program to transpose a given matrix. | 38 |
| 24. | Write a program to find the factorial of a number by using the concept of recursion. | 39 |
| 25. | Write a menu driven C program to show the use of in-built string functions like strlen, strcat, strcpy, strcmp, strrev etc. | 40-42 |
| 26. | Write a Program in C to display the total number of appearances of a substring provided as input by the user in a given string. | 43 |
| 27. | Write a program to display the sum of the digits of a number by using the concept of recursion. | 44 |
| 28. | Write a C program to add two distances in inch & feet using the concept of structures. | 45 |
| 29. | Write a C program to add two complex numbers using the concept of structures in C. | 46 |
| 30. | Write a program in C to store the information of five employees using both concepts i.e. array of structure and array within structure. | 47-48 |
| 31. | Write a Program in C to find and replace a specific string in a file and also display the total number of appearances of that string. | 49-51 |

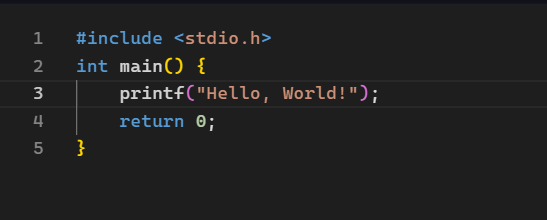
**Experiment No. 1.**

**Aim:** Install C compiler (GCC/Code::Blocks), set up IDE, compile and run the first "Hello, World!" program.

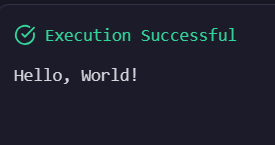
**Concept Used:**

* **C Compiler and IDE Setup:** Understanding how to install a compiler like GCC or Code::Blocks to write and execute C programs.
* **Basic Structure of a C Program:** Introduction to #include <stdio.h>, main(), and printf().
* **Compilation and Execution:** Understanding the process of compiling a C program into machine code and running it.

**Program**:



**Output Screenshot:**



**Experiment No. 2.**

**Aim:** Write a Program to show the use to input (Scanf)/output (Printf) statements and block structure of C-program by highlighting the features of "stdio.h".

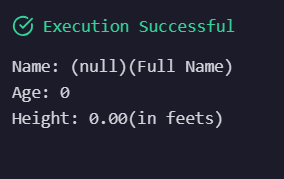
**Concept Used:**

* **Standard Input/Output in C:** Using #include <stdio.h> for input-output functions.
* **Formatted Input (scanf) and Output (printf)** to accept and display values.
* **Block Structure of C Program:** Understanding function blocks, variable declarations, and execution flow.

**Program**:



**Output Screenshot:**

****

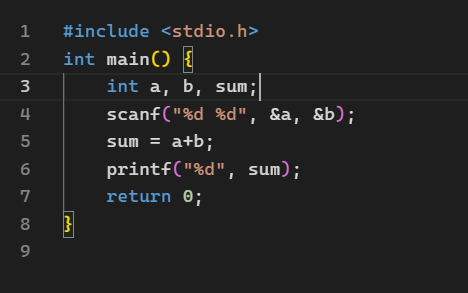
**Experiment No. 3.**

**Aim:** Write a program to add two numbers and display the sum.

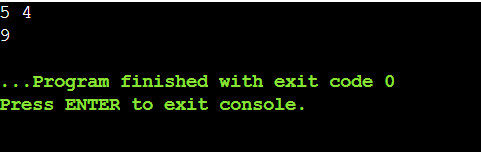
**Concept Used:**

* **Variables and Data Types:** Declaring variables to store integer values.
* **Arithmetic Operators:** Using the + operator for addition.
* **Basic Input/Output Functions:** scanf() to take user input and printf() to display results.

**Program**:



**Output Screenshot:**

****

**Experiment No. 4.**

**Aim:** Write a program to calculate the area and the circumference of a circle by using radius as the input provided by the user.

**Concept Used:**

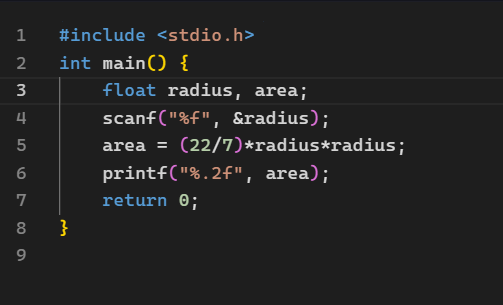
 **Mathematical Expressions:** Using the formulas:

* Area = π × radius²
* Circumference = 2 × π × radius

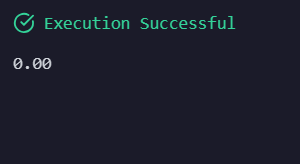
 **Constants in C:** Using #define or const for π = 3.1416.

 **Floating-point Arithmetic:** Handling decimal calculations using float or double.

**Program**:



**Output Screenshot:**

****

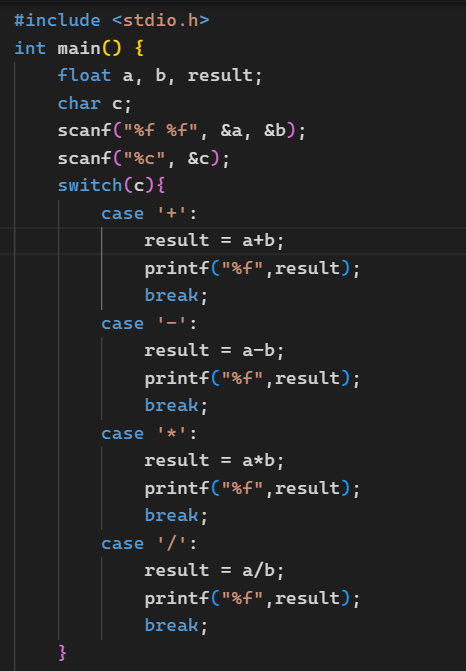
**Experiment No. 5.**

**Aim:** Write a Program to perform addition, subtraction, division and multiplication of two numbers given as input by the user.

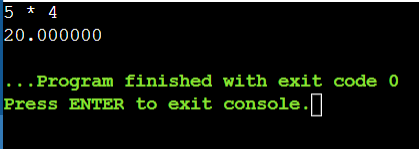
**Concept Used:**

* **Arithmetic Operators:** +, -, \*, and / for basic mathematical operations.
* **Operator Precedence:** Understanding how different operations are evaluated.
* **Handling User Input:** Using scanf() to take values from the user.

**Program**:



**Output Screenshot:**

****

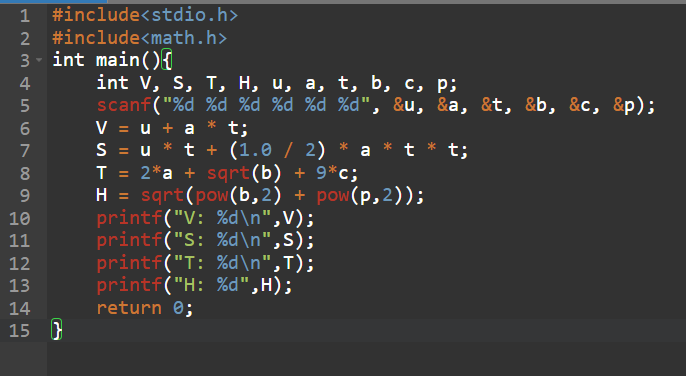
**Experiment No. 6.**

**Aim:** Write a program to evaluate each of the following equations.  
(i) V = u + at. (ii) S = ut+1/2at2 (iii) T=2\*a+√b+9c (iv) H=√b2+p2

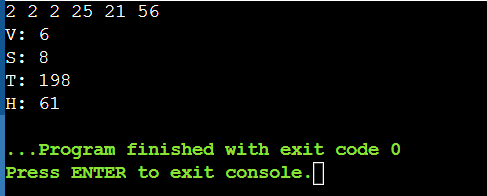
**Concept Used:**

* Mathematical Expressions and Order of Evaluation:
* V = u + at
* S = ut + 1/2at²
* T = 2\*a + √b + 9c
* H = √(b² + p²)
* Using math.h Library: Required for square root (sqrt()).
* Variable Declaration and Arithmetic Operations: Assigning and computing values.

**Program**:

****

**Output Screenshot:**

****

**Experiment No.7.**

**Aim:** Write a program to swap two variables:

a) By using temporary variable.

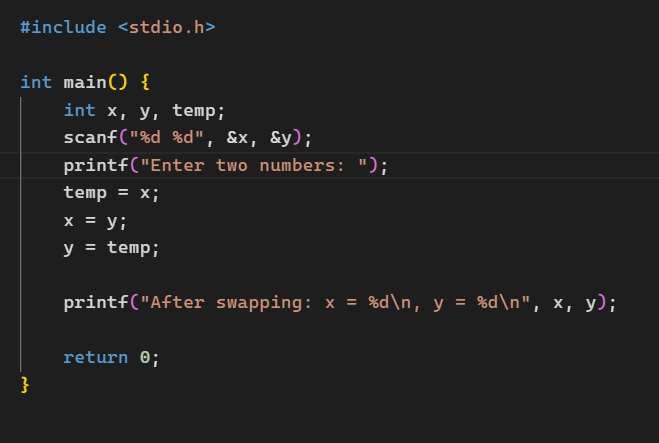
b) Without using temporary variable

**Concept Used:**

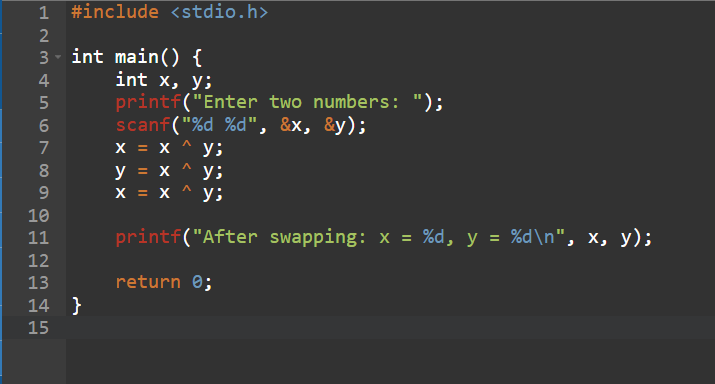
* **Using a Temporary Variable:** Swapping values using an intermediate storage.
* **Without a Temporary Variable:** Swapping using arithmetic operations:
  + a = a + b; b = a - b; a = a - b;
* **Understanding Memory Storage and Assignment Operations.**

**Program**:

1. By using temporary variable

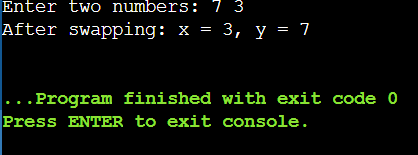


1. By using temporary variable

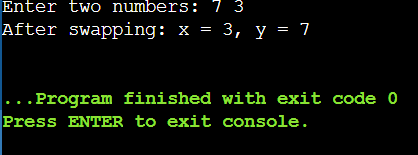


**Output Screenshot:**

1. By using temporary variable

****

1. By using temporary variable

****

**Experiment No. 8.**

**Aim:** Write a Program to find the greatest among three numbers using:

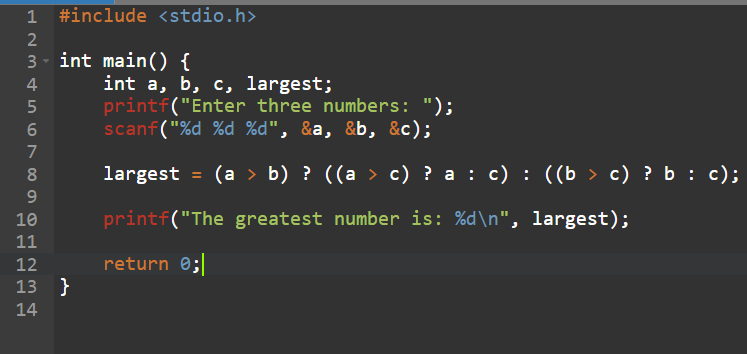
* Conditional Operator
* If-Else statement

**Concept Used:**

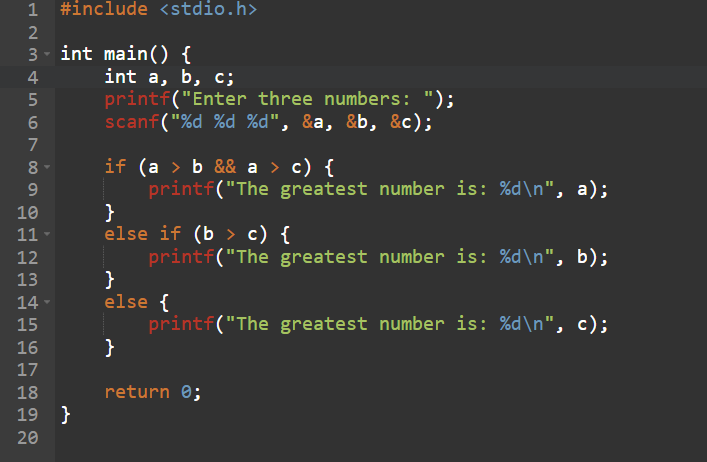
* **Conditional Operator (?:)** for concise decision-making.
* **If-Else Statement:** Basic decision control structure.
* **Comparison Operators (>, <)** to check for the greatest number.

**Program**:

1. Conditional operator

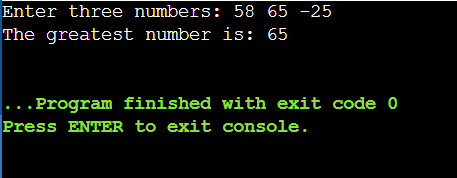


1. If-Else statement

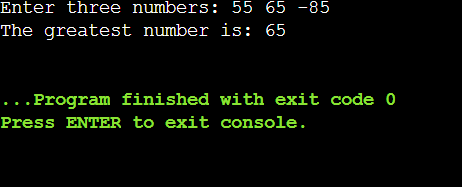


**Output Screenshot:**

1. Conditional operator



1. If-Else statement



**Experiment No. 9.**

**Aim:** Write the following programs using switch case statement:

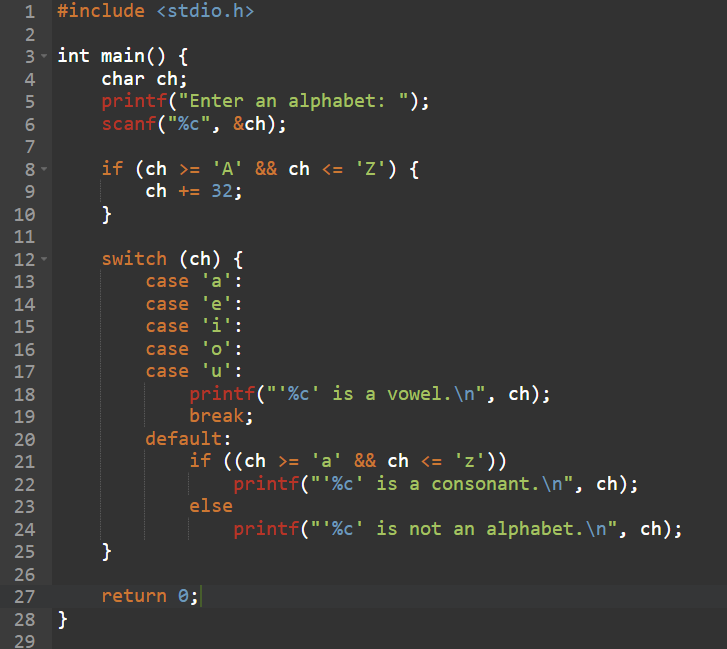
* To check that an input alphabet is vowel or consonant
* To check whether a number is positive, negative or zero

**Concept Used:**

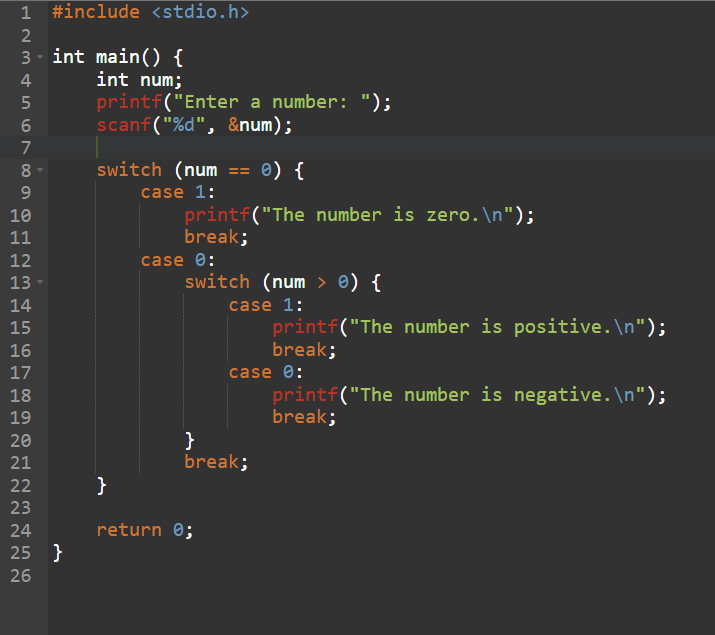
* **Switch-Case Statement:** Alternative to multiple if-else statements for checking conditions.
* **Character and Number Checking:**
* Checking vowels (a, e, i, o, u).
* Checking if a number is positive, negative, or zero

**Program**:

* To check that an input alphabet is vowel or consonant

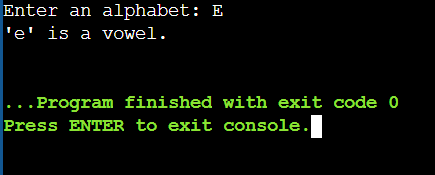


* To check whether a number is positive, negative or zero

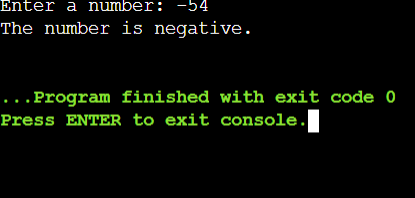


**Output Screenshot:**

* To check that an input alphabet is vowel or consonant

****

* To check whether a number is positive, negative or zero

****

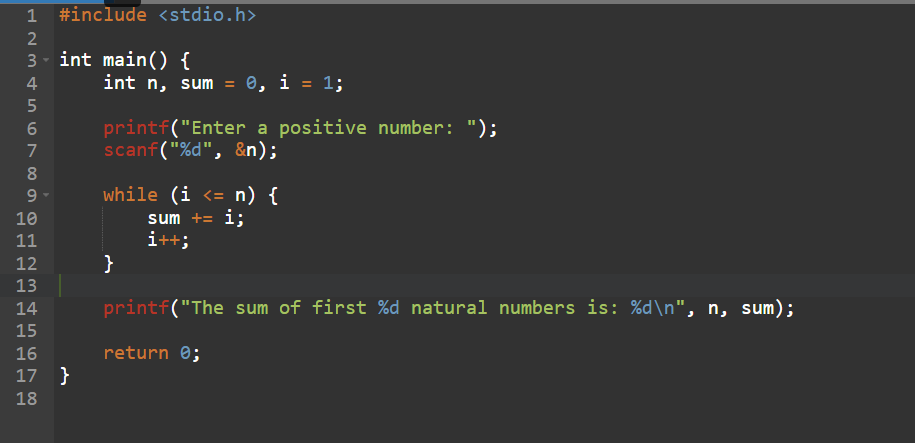
**Experiment No. 10.**

**Aim:** Write a program using while loop to print the sum of first n natural numbers.

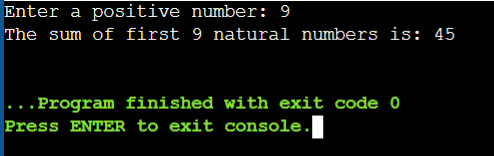
**Concept Used:**

* **Loops in C (While Loop):** Repeatedly adding numbers from 1 to N.
* **Iteration and Accumulation:** Using a variable to store the sum dynamically.

**Program**:



**Output Screenshot:**

****

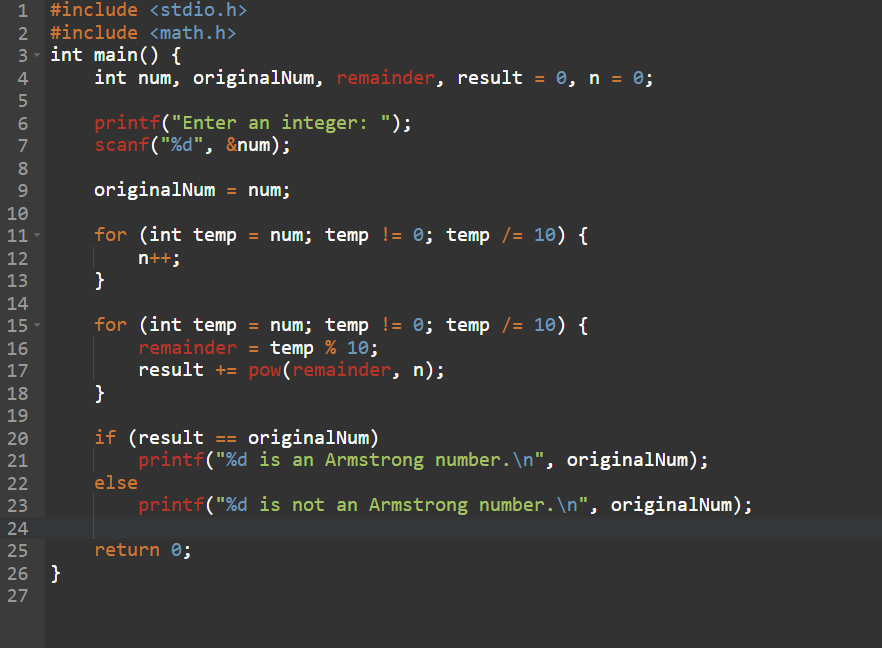
**Experiment No. 11.**

**Aim:** Write a program to check a number is Armstrong or not using For loop.

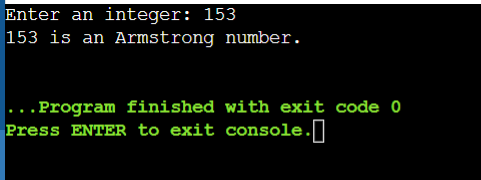
**Concept Used:**

* **For Loop for Iteration:** Iterating through digits of a number.
* **Mathematical Computation:** Checking if the sum of cube of digits equals the original number.
* **Modulus (%) and Division (/) Operations:** Extracting digits.

**Program**:



**Output Screenshot:**

****

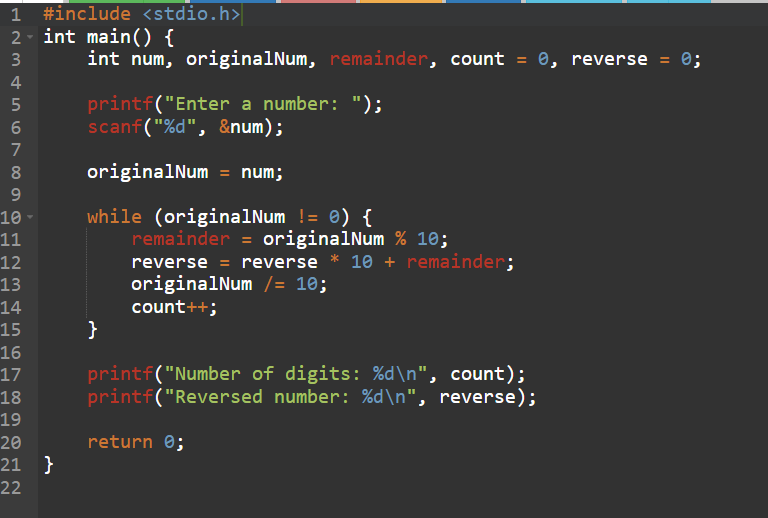
**Experiment No. 12.**

**Aim:** Write the program to count the digits in a number and then print the reverse of the number also.

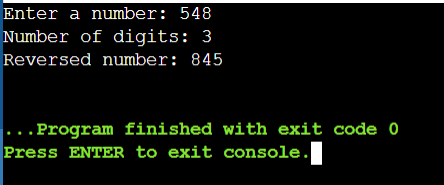
**Concept Used:**

* **Looping (while loop)** – Runs until originalNum becomes 0.
* **Modulus Operator (%)** – Extracts the last digit (remainder = originalNum % 10).
* **Reversing a Number** – Builds the reversed number using reverse = reverse \* 10 + remainder.
* **Counting Digits** – count++ increments the counter for each extracted digit.
* **Integer Division (/)** – originalNum /= 10; removes the last digit to continue the process.

**Program**:



**Output Screenshot:**

****

**Experiment No. 13.**

**Aim:** Write a program to generate the Fibonacci series.

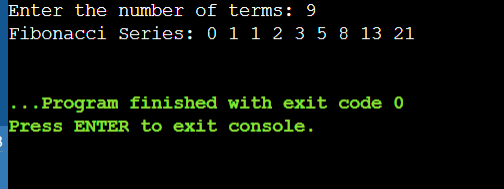
**Concept Used:**

* **Recursion vs Iteration:** Understanding the two approaches.
* **Looping (for loop)** – Iterates n times to generate the Fibonacci sequence.
* **Arithmetic Operations** – Addition (next = first + second;) to compute Fibonacci numbers.
* **Variable Swapping** – Updates first and second for the next Fibonacci term.

**Program**:



**Output Screenshot:**

****

**Experiment No. 14.**

**Aim:** Write a program to print the following patterns:

a)

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

\* \* \* \* \* \*

b)

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

\* \* \* \* \* \*

**Concept Used:**

* **Nested Loops (for loop)** –
* Outer loop (i) controls the number of rows.
* Inner loop (j) controls the number of \* printed in each row.
* **Loop Control & Iteration –**
* The number of \* increases as i increases.
* printf("\n") moves to the next line after each row.

**Program**:

a)

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

\* \* \* \* \* \*



b)

\*

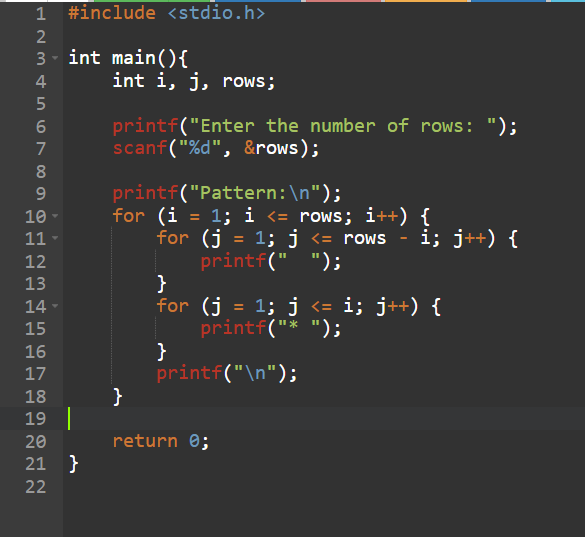
\* \*

\* \* \*

\* \* \* \*

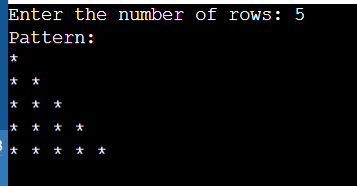
\* \* \* \* \*

\* \* \* \* \* \*

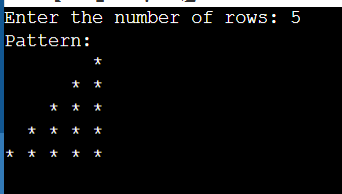


**Output Screenshot:**

**a)**

****

**b)**

****

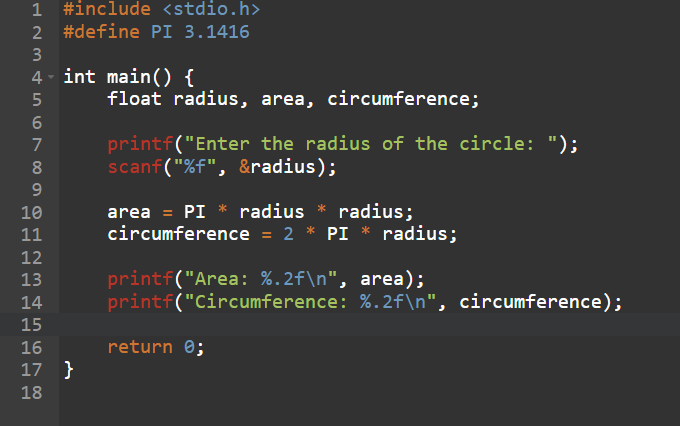
**Experiment No. 15.**

**Aim:** Write a program to calculate the area and circumference of a circle using functions.

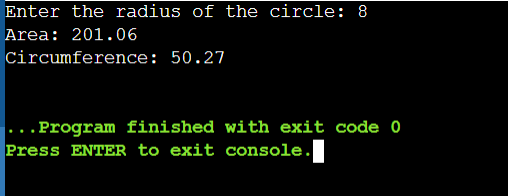
**Concept Used:**

* **Functions in C:** Creating reusable functions for area and circumference calculations.
* **Function Prototypes and Definitions:** Understanding function calls and return values.
* **Preprocessor Directive (#define)** – Defines PI as constant (3.1416).
* **Variables & Data Types** – float for handling decimal values.
* **User Input/Output (** scanf , printf **) –** Takes user input and displays output,
* **Mathematical Expressions –** 
  + Area = pi \* r \* r
  + Circumference = 2 \* pi \*r
* **Arithmetic Operations** – Multiplication (\*) used for calculations.
* **Structured Programming** – Code follows a sequential flow inside main().

**Program**:



**Output Screenshot:**

****

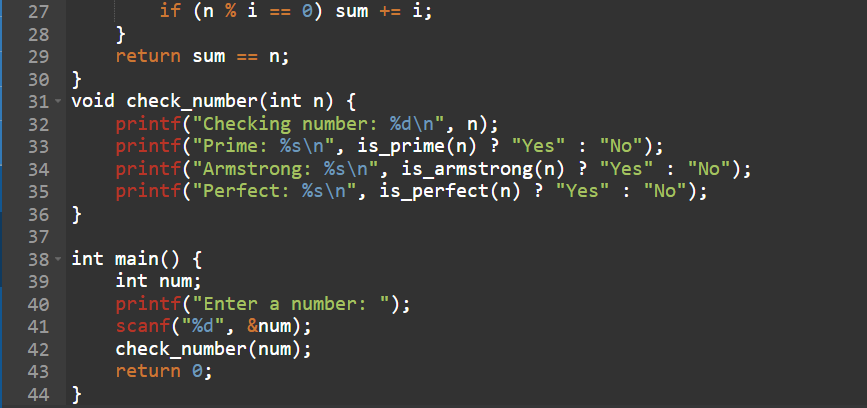
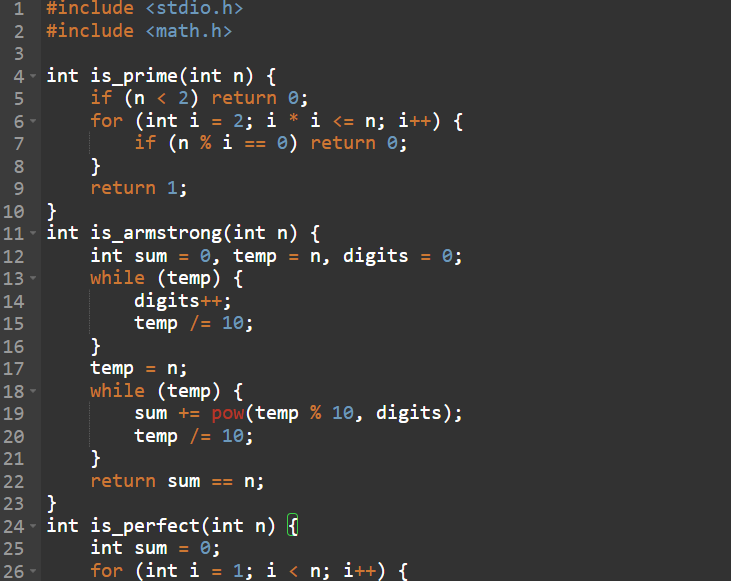
**Experiment No. 16**

**Aim:** Write a program to check that the given number is prime, Armstrong or perfect using the concept of functions.

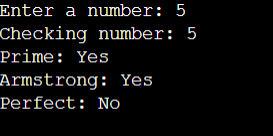
**Concept Used:**

* **Functions in C:** To modularize the logic for prime, Armstrong, and perfect number checks.
* **Mathematical Operations:** Use of modulus (%), loops, and number theory.
* **Structured Programming:** Separate functions for each type of number check.

**Program:**

****

**Output Screenshot:**

****

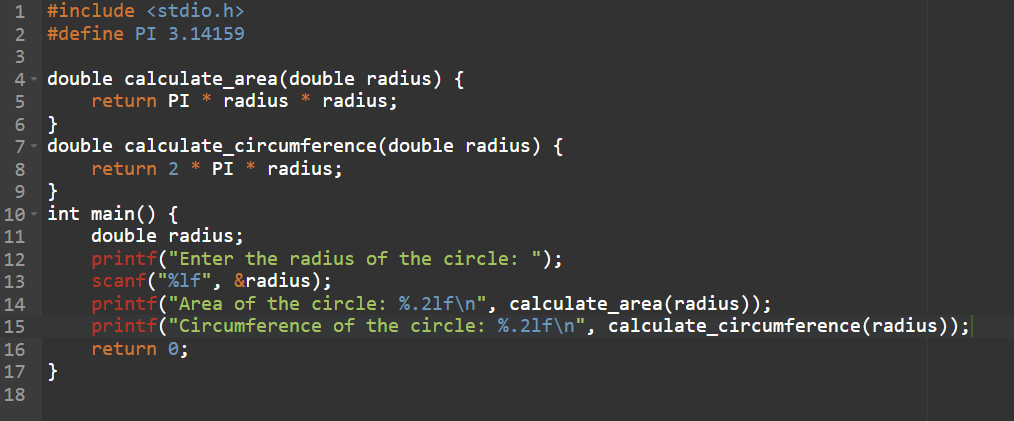
**Experiment No. 17**

**Aim:** Write a program to calculate the area and circumference of a circle using functions.

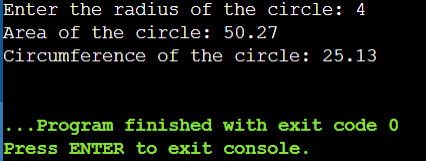
**Concept Used:**

* **Functions:** Separate reusable functions for area and circumference.
* **Math Expressions:** Area = π × r², Circumference = 2πr.
* **Preprocessor Directive:** #define PI 3.1416
* **User Input & Output:** Use of scanf and printf.

**Program:**

****

**Output Screenshot:**

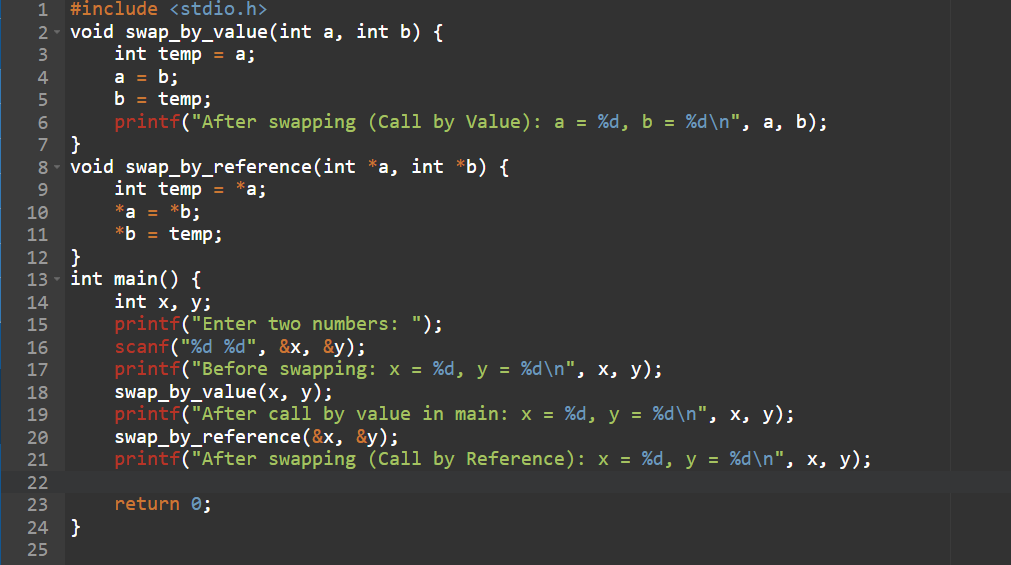
****

**Experiment No. 18**

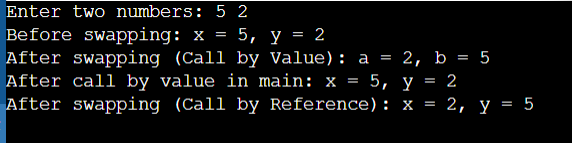
**Aim:** Write a program to swap two variables using the concept of call by value and call by reference.

**Concept Used:**

* **Call by Value:** Function receives a copy of the data.
* **Call by Reference:** Function receives address using pointers.
* **Understanding Variable Scope and Memory Access**

**Program:  
  
**

**Output Screenshot:**

****

**Experiment No. 19**

**Aim:** Write a program to perform the following operations on 1D-Array:

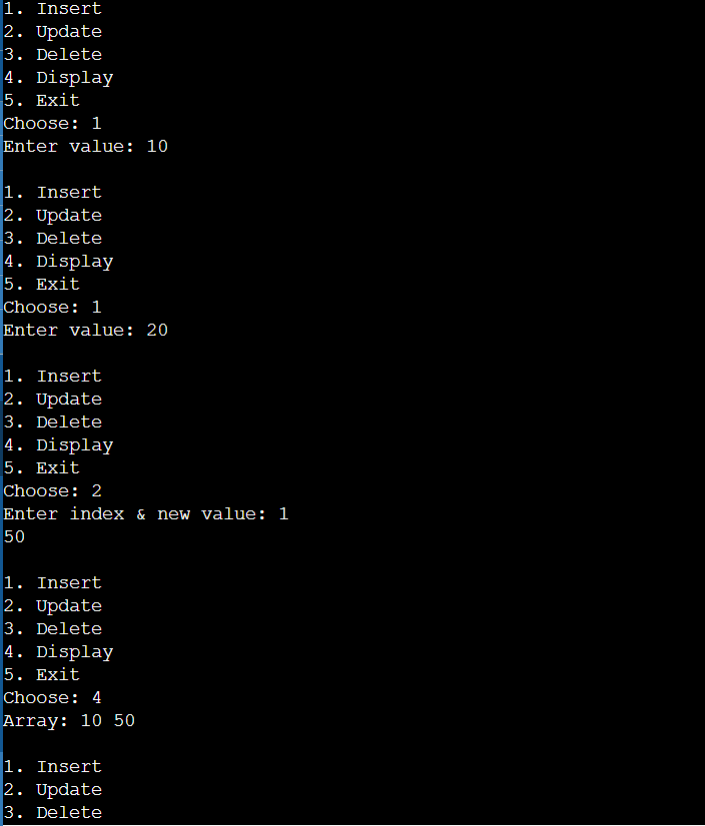
* Insert
* Update
* Delete
* Display
* Search

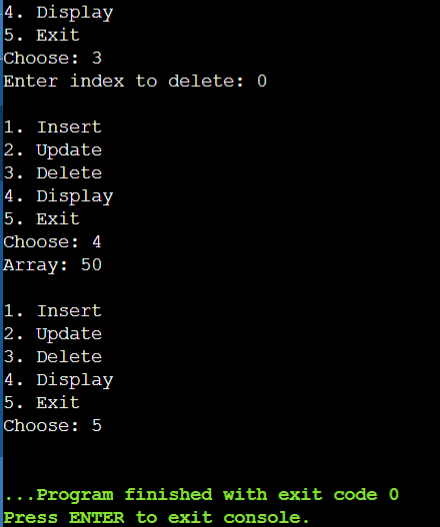
**Concept Used:**

* **Arrays:** Fixed-size data storage.
* **Functions & Menu-driven Programs**
* **Looping and Conditionals:** To perform different operations.
* **Index Manipulation:** For shifting elements during insert/delete.

**Program:  
 **

**Output Screenshot:**

****

****

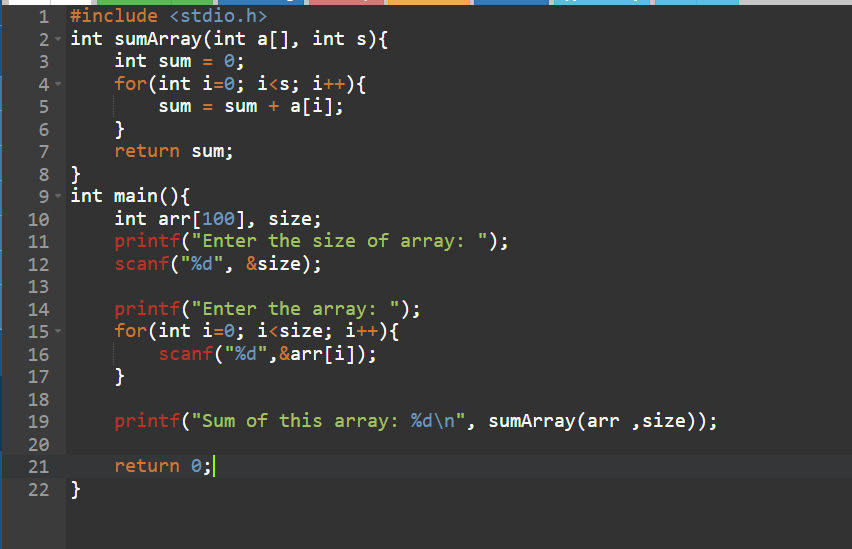
**Experiment No. 20**

**Aim:** Write a program to calculate the sum of array elements by passing it to a function.

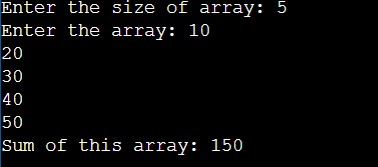
**Concept Used:**

* **Function with Array Parameter**
* **Loops:** For traversal and summation.
* **Array Handling:** Passing array to function.

**Program:**

****

**Output Screenshot:**

****

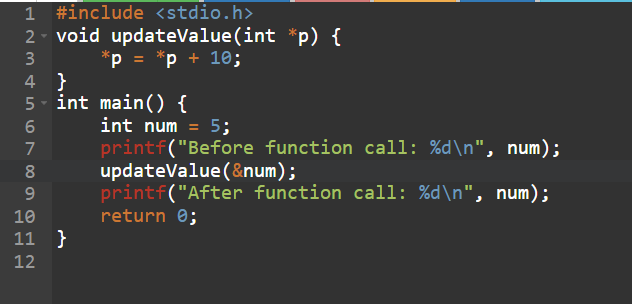
**Experiment No. 21**

**Aim:** Write a program to show the use of passing pointer as arguments to the functions.

**Concept Used:**

* **Pointers:** Direct memory access.
* **Function Arguments:** Modifying values using pointer dereferencing.
* **Call by Reference**

**Program:**

****

**Output Screenshot:**

****

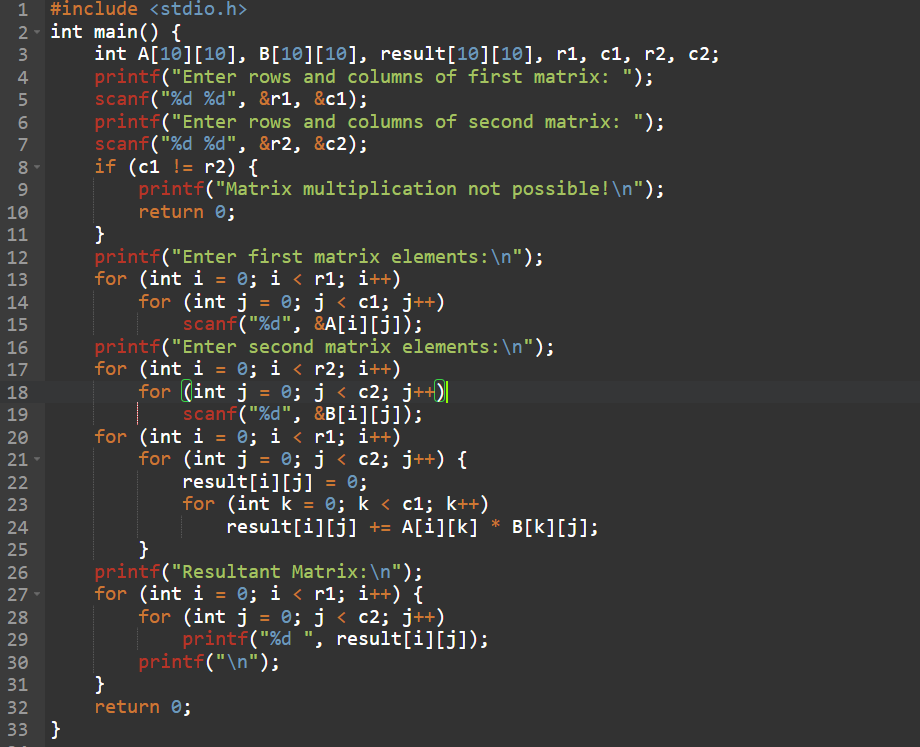
**Experiment No. 22**

**Aim:** Write a program matrix multiplication using the concept of 2D array

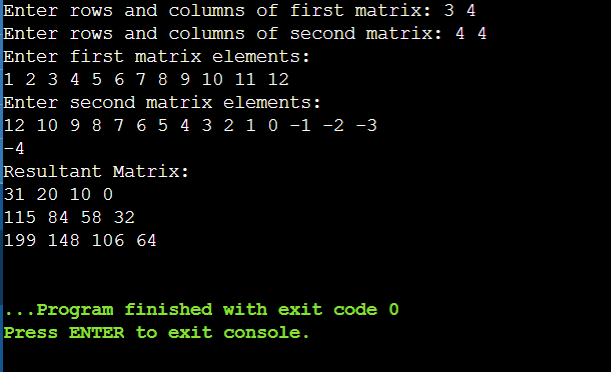
**Concept Used:**

* **2D Arrays:** For storing matrices.
* **Nested Loops:** For row-column multiplication.
* **Matrix Concepts:** Dot product computation.

**Program:**

****

**Output Screenshot:**

****

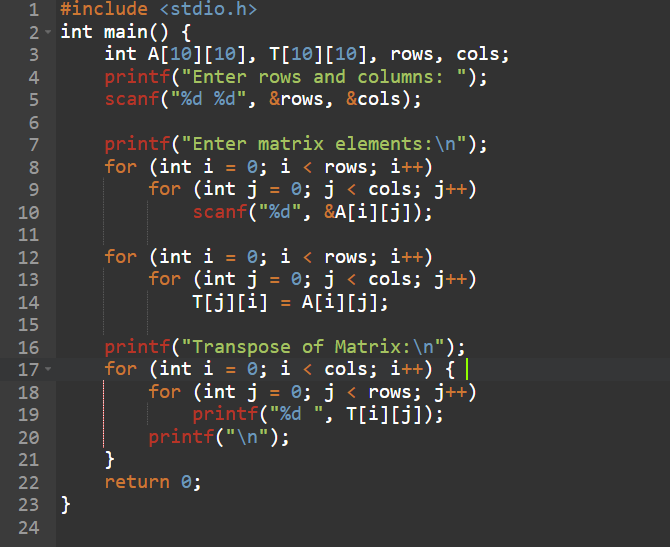
**Experiment No. 23**

**Aim:** Write a program to transpose a given matrix.

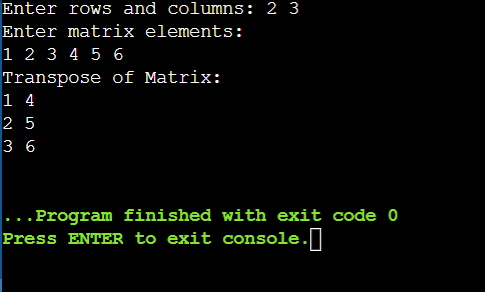
**Concept Used:**

* **2D Arrays**
* **Swapping Rows with Columns**
* **Loops:** Nested loop for accessing matrix elements.

**Program:**

****

**Output Screenshot:**

****

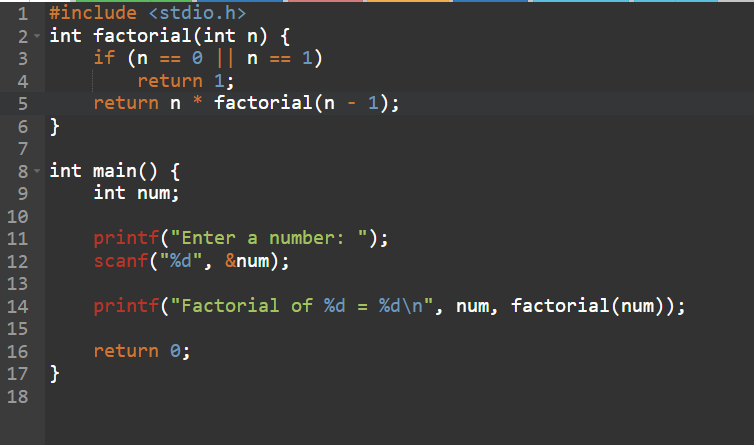
**Experiment No. 24**

**Aim:** Write a program to find the factorial of a number by using the concept of recursion.

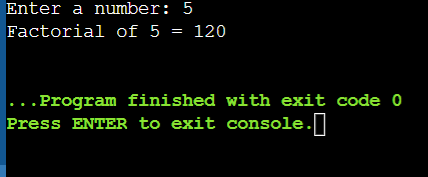
**Concept Used:**

* **Recursion:** A function calling itself.
* **Base Case & Recursive Case**
* **Mathematics:** n! = n × (n-1)!

**Program:**

****

**Output Screenshot:**

****

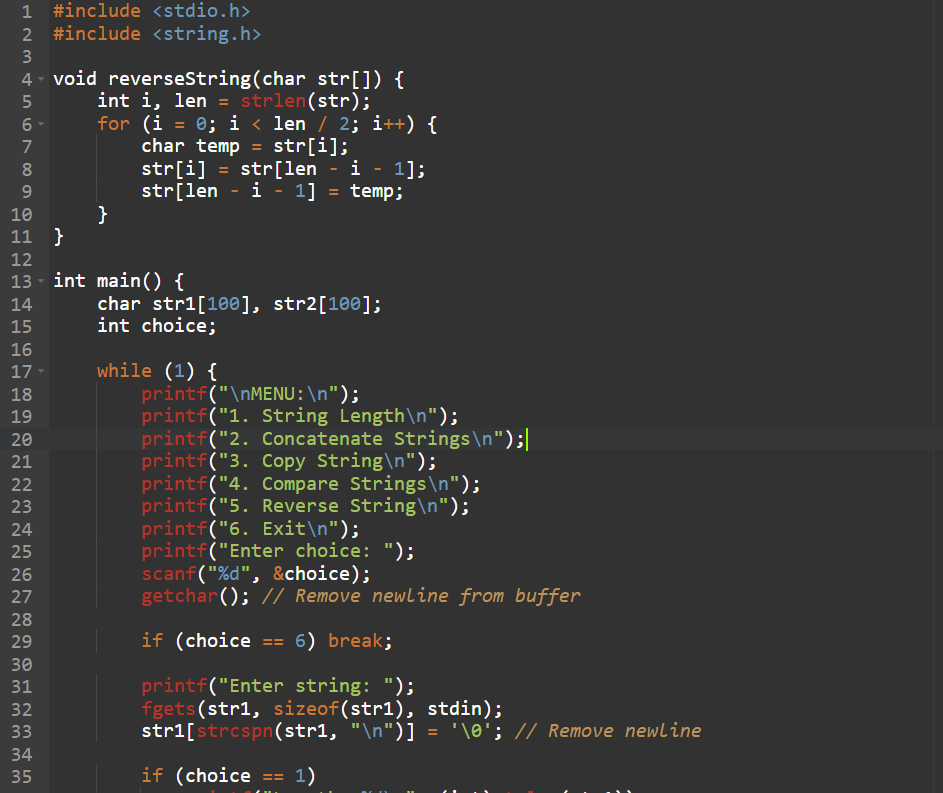
**Experiment No. 25**

**Aim:** Write a menu driven C program to show the use of in-built string functions like strlen, strcat, strcpy, strcmp, strrev etc.

**Concept Used:**

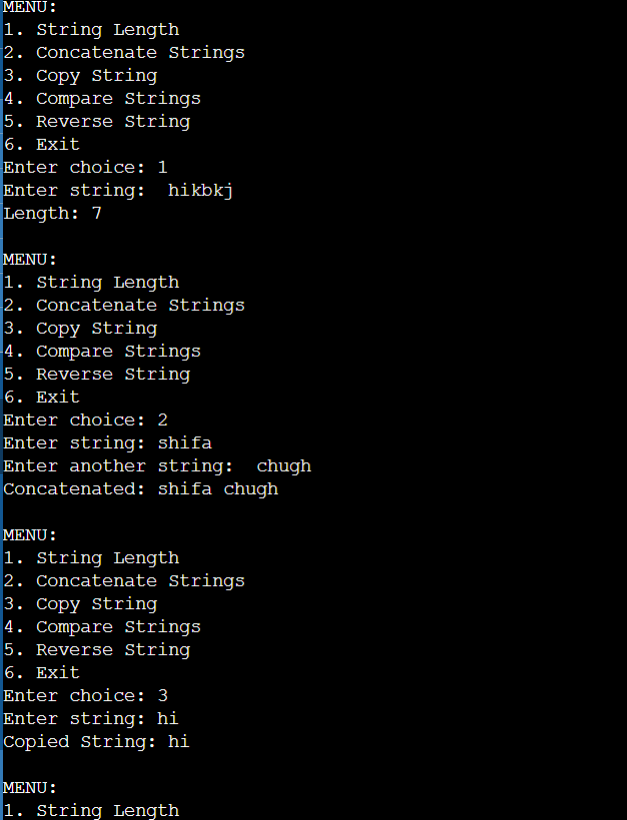
* **String Functions:** strlen(), strcpy(), strcat(), strcmp(), strrev()
* **Menu-Driven Logic**
* **Strings and Character Arrays**

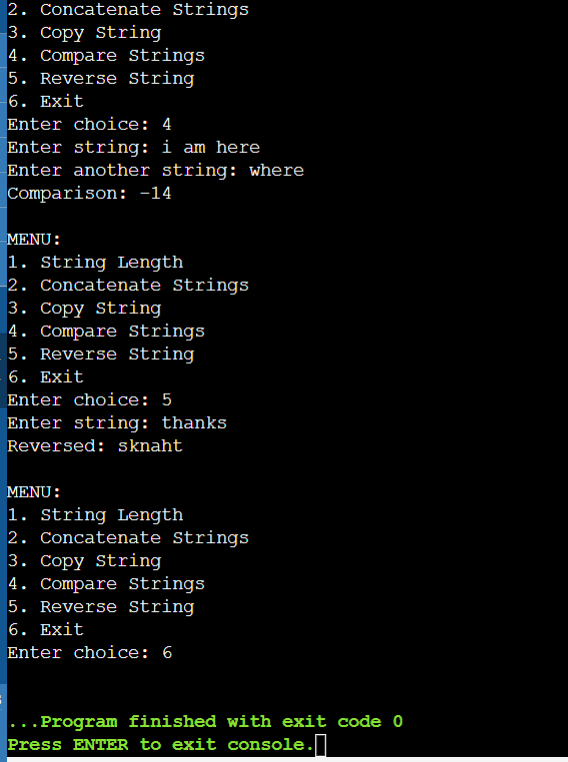
**Program:**

****

****

**Output Screenshot:**

****

****

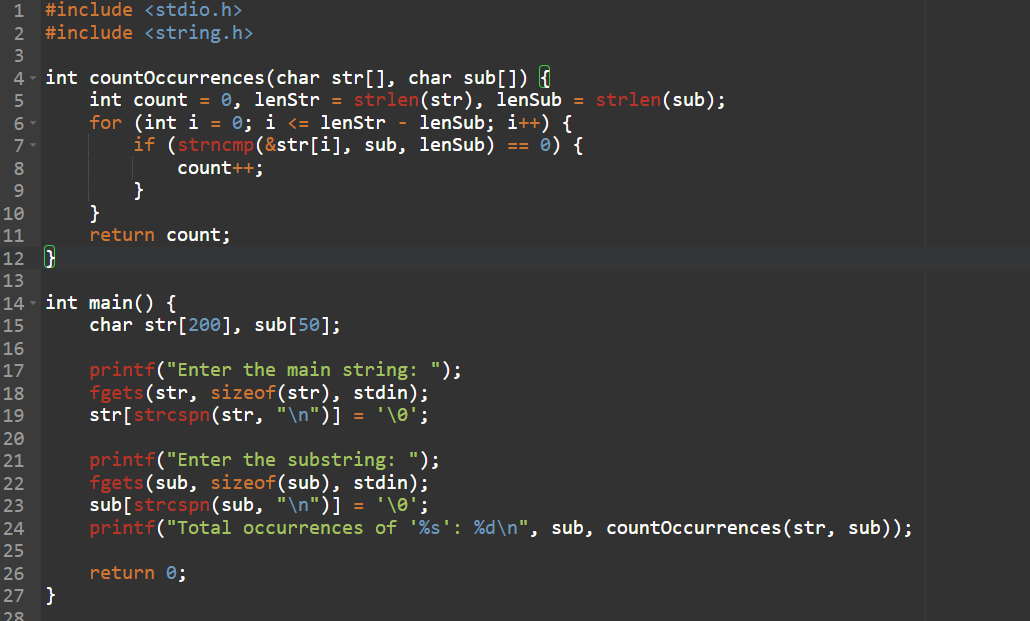
**Experiment No. 26**

**Aim:** Write a Program in C to display the total number of appearances of a substring provided as input by the user in a given string.

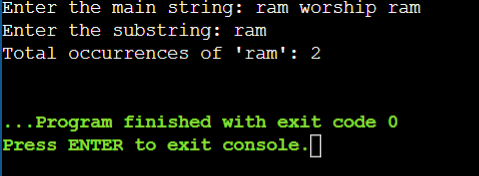
**Concept Used:**

* **String Search:** Manual or using strstr().
* **Looping:** Traverse the string and count matches.
* **Pointer/String Manipulation**

**Program:**

****

**Output Screenshot:**

****

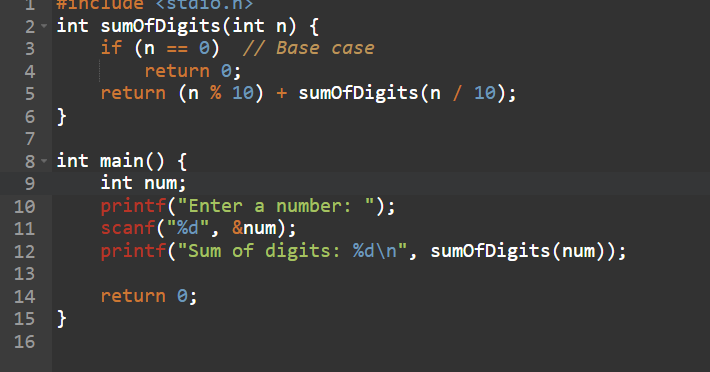
**Experiment No. 27**

**Aim:** Write a program to display the sum of the digits of a number by using the concept of recursion.

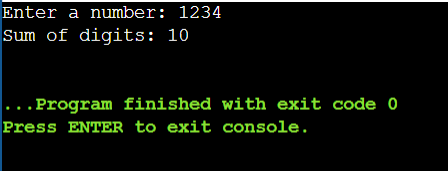
**Concept Used:**

* **Recursion:** For repetitive digit sum calculation.
* **Base Case:** If number is 0, return 0.
* **Mathematics:** Sum = last digit + recursive call on remaining digits.

**Program:**

****

**Output Screenshot:**

****

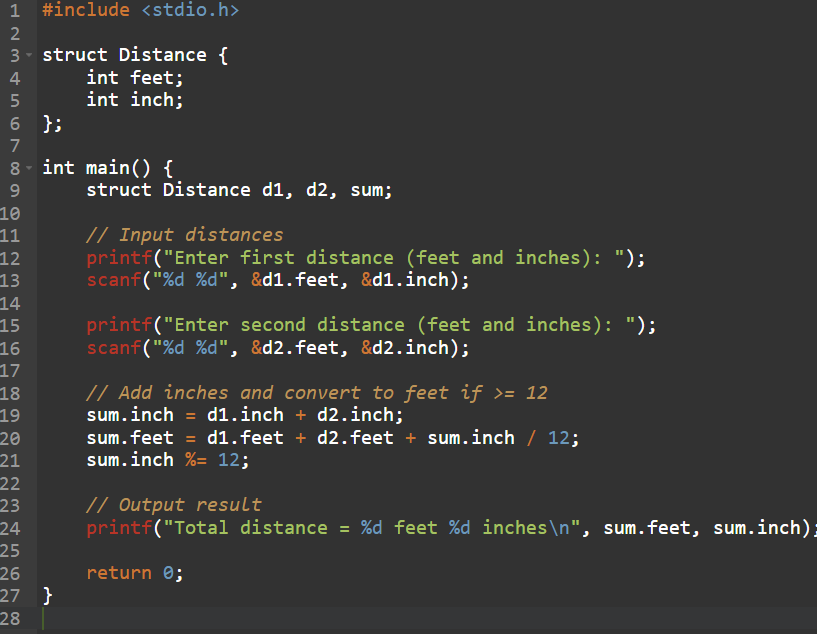
**Experiment No. 28**

**Aim:** Write a C program to add two distances in inch & feet using the concept of structures.

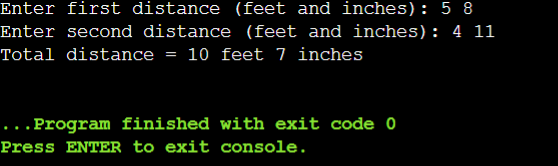
**Concept used:**

* **Structure in C:** Used to group feet and inches as one unit.
* **User-defined data type:** A structure named Distance holds both fields.
* **Addition logic:** Add inches separately and convert to feet if inches ≥ 12.

**Program:**

****

**Output Screenshot:**

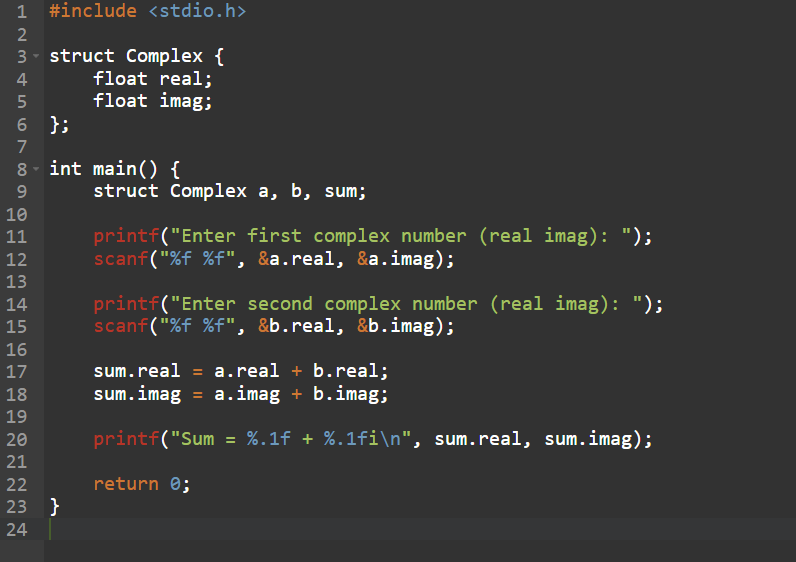
****

**Experiment No. 29**

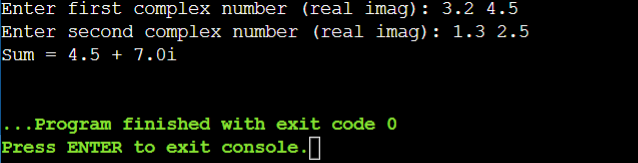
**Aim:** Write a C program to add two complex numbers using the concept of structures in C.

**Concept used:**

* **Structure:** Used to represent a complex number with real and imaginary parts.
* **Encapsulation:** Real and imaginary parts combined in one structure.
* **Arithmetic operation:** Real parts and imaginary parts are added separately.

**Program:  
**

**Output Screenshot:**

****

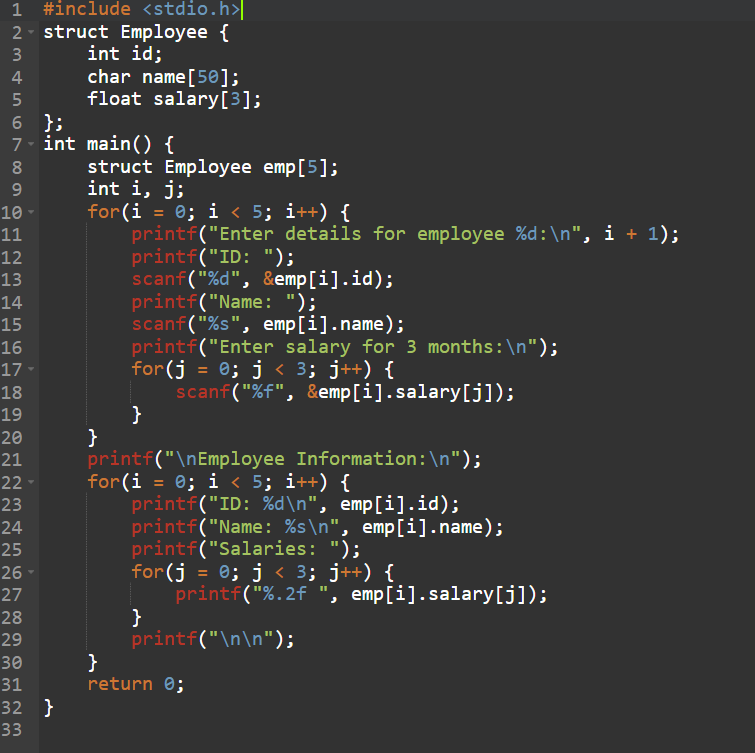
**Experiment No. 30**

**Aim:** Write a program in C to store the information of five employees using both concepts i.e. array of structure and array within structure

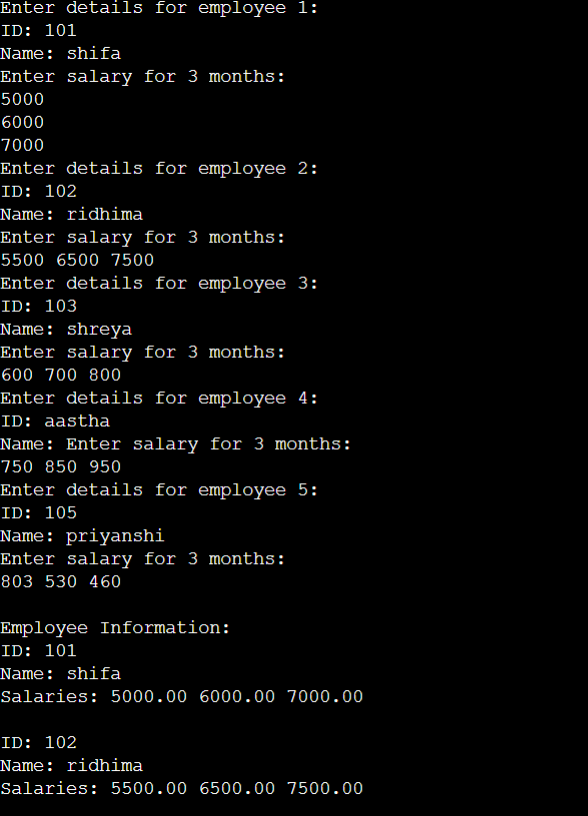
**Concept used:**

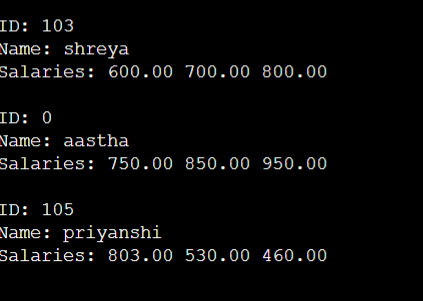
* **Array of Structures:** To store data of multiple employees.
* **Array within a Structure:** Used for storing character arrays like names or departments.
* **Data Handling:** Structure contains fields like ID, name, salary, etc., and stored in an array to manage multiple records.

**Program:**

****

**Output Screenshot:**

****

****

**Experiment No. 31**

**Aim:** Write a Program in C to find and replace a specific string in a file and also display the total number of appearances of that string.

**Concept used:**

**Program:**

**Output Screenshot:**