

# **Lab Manual**

## **Problem Solving Using C**



Department of Computer Science Engineering & Applications  
Indira Gandhi Institute of Technology, Sarang  
Dhenkanal, Odisha-759146

## Experiments

▪ Section 1: Problem Solving Through Computers & Introduction to C.....	1
▪ Section 2: Tokens & Data Types.....	1
▪ Section 3: Operators & Expressions .....	2
▪ Section 4: Branch Control Structures .....	2
▪ Section 5: Loop & Jump Control Structures .....	3
▪ Section 6: One-Dimensional Arrays.....	3
▪ Section 7: Two-Dimensional Arrays .....	4
▪ Section 8: Strings.....	5
▪ Section 9: Pointers .....	5
▪ Section 10: Functions.....	5
▪ Section 11: Structures.....	6
▪ Section 12: Unions.....	6
▪ Section 13: Dynamic Memory Management.....	7
▪ Section 14: Sparse Matrixes .....	7
▪ Section 15: Single Linked Lists .....	7
▪ Section 16: Double Linked Lists .....	8
▪ Section 17: File Input/Output & Command Line Arguments .....	8
▪ Section 18: Graphics Programming.....	9

## Section 1: Problem Solving Through Computers & Introduction to C

- **Experiment 1.1:** Write an algorithm and draw the flow chart to compute the smallest of three numbers.
- **Experiment 1.2:** Write an algorithm and draw the flow chart to enter the temperature in degree Celsius and display it in Fahrenheit and vice-versa. [NOTE:  $C = (5.0/9.0) \times (F - 32)$ ].
- **Experiment 1.3:** Write an algorithm and draw the flow chart to enter the marks obtained by a student in different subjects and calculate the % of marks obtained in aggregate as per the following conditions:
  - If the aggregate mark is less than 30%, print fail.
  - If the aggregate mark is between 30% to 49%, print 3rd division.
  - If the aggregate mark is between 50% to 69%, print 2nd division.
  - If the aggregate mark is equal to or above 70%, print 1st division.
- **Experiment 1.4:** Write a program to enter the temperature in degree Celsius and display it in Fahrenheit and vice-versa. [NOTE:  $C = (5.0/9.0) \times (F - 32)$ ].
- **Experiment 1.5:** Write a program that takes the radius of a circle and displays its area and perimeter.
- **Experiment 1.6:** Write a program to swap two numbers.
- **Experiment 1.7:** Write a program to swap two numbers without using a third variable.
- **Experiment 1.8:** Write a program to print the following structure using simple `printf` statements:

```
*
* *
* * *
* * * *
```

## Section 2: Tokens & Data Types

- **Experiment 2.1:** Write a program to compute the area of a triangle given its base and height.
- **Experiment 2.2:** Write a program that takes as input, the principal, rate of interest, and time, and then calculates the simple and compound interest.
- **Experiment 2.3:** Write a program that requests two float type numbers from the user and then divides the 1<sup>st</sup> number by the 2<sup>nd</sup> and displays the result along with the numbers.
- **Experiment 2.4:** Write a program to read the price of an item in decimal form (like 15.95) and print the output in paise (like 1595 paise).
- **Experiment 2.5:** Rewrite the program mentioned in Experiment 2.2. However, rename the data types 'float' to 'Decimal' and 'int' to 'integer' by using `typedef` statement.

## Section 3: Operators & Expressions

- **Experiment 3.1:** Write a program to compute the area of a triangle given its three sides.
- **Experiment 3.2:** Write a program to take a number as input and check whether it is even or odd using conditional operator (*do not use if statement*).
- **Experiment 3.3:** Write a program to input a character and check whether it is in upper case or lower case using conditional operator. [Hint: compare against the ASCII range.]
- **Experiment 3.4:** Write a program to input a character and check whether it is a vowel or consonant using conditional operator. [Hint: compare against the ASCII range.]
- **Experiment 3.5:** Write a program to read a four-digit integer and print the sum of its digits. [Hint: Use / and % operators.]

## Section 4: Branch Control Structures

- **Experiment 4.1:** Write a program to input three numbers and find out the smallest.
- **Experiment 4.2:** Write a program to input a number and print “ODD” if the number is odd and “EVEN” if the number is even.
- **Experiment 4.3:** Admission to professional course is subject to the any one of the following conditions:
  - a) Marks in Mathematics  $\geq 60$ , and Marks in Physics  $\geq 50$ , and Marks in Chemistry  $\geq 40$
  - b) Total in all three subjects  $\geq 200$

Write a program that takes as input the marks in three subjects and displays “Admitted” if the student is admitted else displays “Not Admitted”.

- **Experiment 4.4:** Write a program that will read the value of x and evaluate the following function:

$$y = \begin{cases} 1 & \text{for } x > 0 \\ 0 & \text{for } x = 0 \\ -1 & \text{for } x < 0 \end{cases}$$

- **Experiment 4.5:** An electricity board charges the following rates for the use of electricity.
  - For the first 200 units: 80 paise per unit
  - For the next 100 units: Rs. 1.30 per unit
  - Beyond 300 units: Rs. 2.00 per unit

All users are charged a minimum of Rs. 100 as meter charge. If the total amount is more than Rs. 400, then an additional surcharge of 15% of total amount is charged. Write a program to read the name of a consumer and the number of units consumed and print the charge with his/her name.

## Section 5: Loop & Jump Control Structures

- **Experiment 5.1:** Write programs to print the following structure:

```
      *
     * * *
    * * * * *
   * * * * * * *
```

- **Experiment 5.2:** Write a program to print the following structure:

```
4 3 2 1 2 3 4
 3 2 1 2 3
  2 1 2
   1
```

- **Experiment 5.3:** Write a program to count the number of digits in a number.
- **Experiment 5.4:** Write a program that takes as input an integer 'n' and computes the following series:  $1 - (2/3!) + (3/4!) - (4/5!) + \dots \pm (n/(n+1)!)$ .
- **Experiment 5.5:** Write a program to find the number of and sum of all integers greater than 100 and less than 200 that are divisible by 7.
- **Experiment 5.6:** Write a program to find out the HCF and LCM of two numbers.
- **Experiment 5.7:** Write a program to input a number and check whether it is palindrome or not (*a palindrome number is a number that remains the same when its digits are reversed, e.g., 16461*).
- **Experiment 5.8:** Write a program to input a number and check whether it is Armstrong or not (*An Armstrong number is an integer such that the sum of the cubes of its digits is equal to the number itself. For example, 371 is an Armstrong number since  $3^3 + 7^3 + 1^3 = 371$* ).
- **Experiment 5.9:** Write a program to input a number and check whether it is perfect or not (*A number is called perfect if the sum of its divisors is equal to the number itself. For example, 28 is a perfect number since  $1+2+4+7+14=28$* ).
- **Experiment 5.10:** Write a program that takes an integer 'n' as input and prints the Fibonacci series up to 'n'. (Hint: Fibonacci series: 1 1 2 3 5 8 13 21 ...)

## Section 6: One-Dimensional Arrays

- **Experiment 6.1:** Write a program to input 'n' numbers and sort them in descending order.
- **Experiment 6.2:** Write a program to input 'n' numbers and print the 2<sup>nd</sup> largest and 2<sup>nd</sup> smallest number.
- **Experiment 6.3:** Write a program to read 'n' characters and then delete the duplicate characters.

- **Experiment 6.4:** Write a program to read 'n' real numbers and then insert a new real number at a given position (after insertion, the array size should increase by 1).
- **Experiment 6.5:** Write a program to read 'n' numbers and find the frequency of occurrence of each number.
- **Experiment 6.6:** Write a menu-based program that takes two linear arrays A and B as input and displays the results of the following operations based on user's choice.
  1. Traverse A.
  2. Traverse B.
  3. Insert a new element in A (after insertion, the size of A should increase by 1).
  4. Insert a new element in B (after insertion, the size of B should increase by 1).
  5. Delete an element from A.
  6. Delete an element from B.
  7. Search an element in A.
  8. Search an element in B.
  9. Sort A (in ascending order).
  10. Sort B (in ascending order).
  11. Merge A and B to form a new array C.

## Section 7: Two-Dimensional Arrays

- **Experiment 7.1:** Write a program to input a square matrix (a matrix of order  $n \times n$ ) and check whether it is symmetric or not (A matrix  $A = (a_{ij})$  is symmetric if  $a_{ij} = a_{ji}$ , for all i and j).
- **Experiment 7.2:** Write a program to input a square matrix (a matrix of order  $n \times n$ ) and then print its upper triangular matrix. e.g.

	5	7	9	6		5	7	9	6
For the matrix	2	4	3	1		4	3	1	
	3	5	7	9	Its upper triangular form is:		7	9	
	8	6	9	7				7	

- **Experiment 7.3:** Write a program that finds the sum of each row and sum of each column of a matrix.
- **Experiment 7.4:** Write a program that finds the sum of diagonal elements of a square matrix.

- **Experiment 7.5:** Write a menu-based program that takes two matrixes A and B as input and displays the results of the following operations based on user's choice.
  1.  $A + B$
  2.  $A - B$
  3.  $B - A$
  4.  $A \times B$

## Section 8: Strings

- **Experiment 8.1:** Write a program to find out the length of a string without using the `strlen()` function.
- **Experiment 8.2:** Write a program that will count the number of occurrences of a specific character in a given line of text.
- **Experiment 8.3:** Write a program that takes as input a string and two numbers n1 and n2 and find the substring between these two positions. For example, let the string is "Welcome" and the numbers are n1=2 and n2=5 then the substring will be: "lcom".
- **Experiment 8.4:** Write a program that reads two strings and then compares them without using the `strcmp()` function.
- **Experiment 8.5:** Write a program which reads a string and prints it in alphabetical order. For example, the word WELCOME should be displayed as CEELMOW.

## Section 9: Pointers

- **Experiment 9.1:** Write a program using pointers to input a 4X5 matrix and then display it.
- **Experiment 9.2:** Write a program using pointers to read an array of integers and print its elements in reverse order.

## Section 10: Functions

- **Experiment 10.1:** Write a program using function to determine whether a year is a leap year or not.
- **Experiment 10.2:** Write a program using function to take a string as input and then convert all lowercase characters to its uppercase equivalent.
- **Experiment 10.3:** Use recursive function calls to evaluate the following series:  

$$x - (x^3/3!) + (x^5/5!) - (x^7/7!) + \dots$$

- **Experiment 10.4:** Write a program using function to input a m x n matrix (of integers) and then transpose it.
- **Experiment 10.5:** Write a program using function to input a m x n matrix (of integers) and then find the largest, the 2<sup>nd</sup> largest, the smallest, and the 2<sup>nd</sup> smallest numbers in it.
- **Experiment 10.6:** Using pointers, write a function that receives a character string and a character as arguments and deletes all occurrence of this character in the string. The function should return the corrected string with no holes.
- **Experiment 10.7:** Write a function using pointers to add two matrices and return the resultant matrix. Use this function within the main function to add two matrices.
- **Experiment 10.8:** Write a function using pointers that compares two integer arrays to see whether they are identical. The function returns 1 if they are identical, 0 otherwise.

## Section 11: Structures

- **Experiment 11.1:** A student is represented by the following attributes:
  - Roll number (int)
  - Name (string)
  - Mark in subject 1 (float)
  - Mark in subject 2 (float)
  - Mark in subject 3 (float)

Write a program using structure to create data for 50 students and then find the total marks for each student and average mark of all students.

- **Experiment 11.2:** Write a program to create a structure COMPLEX to represent a complex number. Using the structure write a program to add two complex numbers and display their sum.

## Section 12: Unions

- **Experiment 12.1:** An item is represented by the following attributes:
  - Either a model number (string) or an item code (string)
  - Name (string)
  - Price (float)

Write a program using union to read the information for 5 items from the keyboard and print the same on the screen.



## Section 13: Dynamic Memory Management

- **Experiment 13.1:** Write a program to input 'n' numbers (the value of 'n' is specified at runtime) in an array and display their sum and average.
- **Experiment 13.2:** Write a program to input 'n' numbers (the value of 'n' is specified at runtime) in an array and display the largest and smallest among them.
- **Experiment 13.3:** Write a menu-base program that creates two one-dimensional arrays at runtime and performs the following operations as chosen by the user. Separate functions should be written for each of the following operations.
  1. Insert an element in the 1<sup>st</sup> array (if size exceeds the allocated size, use 'realloc()')
  2. Sort the 1<sup>st</sup> array.
  3. Sort the 2<sup>nd</sup> array.
  4. Merge the two arrays.

## Section 14: Sparse Matrixes

- **Experiment 14.1:** Write a program to convert a sparse matrix given in regular form to 3-tuple form.
- **Experiment 14.2:** Write a program to find the transpose of a sparse matrix given in 3-tuple form.

## Section 15: Single Linked Lists

- **Experiment 15.1:** Write a menu-based program to implement the following operations on a single linked list based on user's choice.
  1. Create a single linked list.
  2. Traverse a single linked list.
  3. Insert a node at the beginning.
  4. Insert a node at the end.
  5. Insert a node at an intermediate position after a given node.
  6. Delete the first node.
  7. Delete the last node.
  8. Delete an intermediate node after a given node.
  9. Sorting the node values.

## Section 16: Double Linked Lists

- **Experiment 16.1:** Write a menu-based program to implement the following operations on a double linked list based on user's choice.
  1. Create a single linked list.
  2. Traverse a single linked list.
  3. Insert a node at the beginning.
  4. Insert a node at the end
  5. Insert a node at an intermediate position after a given node.
  6. Delete the first node.
  7. Delete the last node.
  8. Delete an intermediate node after a given node.

## Section 17: File Input/Output & Command Line Arguments

- **Experiment 17.1:** An employee is represented by the following attributes:
  - Employee ID
  - Name
  - Date of Joining (DD MM YY)
  - Salary

Write a program to read the information of five employees from the keyboard and write it in a file "EmployeesInfo.txt", and again read the same information from the file "EmployeesInfo.txt" and display it on the screen.

- **Experiment 17.2:** Write a program that copies the content of one file to another, where the file names are supplied to the program on command line.
- **Experiment 17.3:** Write a program that takes a line of text from command line and prints characters present in the line in alphabetical order. For example, if we type the line of text "India is my country" on command line the output should be "acdliimnnorstuyy".
- **Experiment 17.4:** Re-write the program as given in Experiment 17.1 where the file name (that stores the employees information) is given on the command line.

## Section 18: Graphics Programming

- **Experiment 18.1:** Write a menu-based program to draw the following shapes based on user's choice (the user should choose what to draw and give the required data, i.e., for drawing a circle the user should give its radius, and the color of the circle).
  1. Line
  2. Circle
  3. Arc
  4. Ellipse
  5. Rectangle
  6. Polygon
  7. Bar
- **Experiment 18.2:** Write a program to draw a moving car.

\*\*\*\*\*