

Laboratory Manual

(Version 10.0)

for

Programming in C-Lab

(MCA-153)

MCA - I Semester

Compiled by:

Dr. Ritika Wason

(Associate Professor, BVICAM, New Delhi)



**Bharati Vidyapeeth's
Institute of Computer Applications
and Management (BVICAM)**

A-4, Paschim Vihar, Rohtak Road, New Delhi-63

Visit us at: www.bvicam.in

Index

List of Abbreviations

Declaration

1. Vision of the Department	1
2. Mission of the Department	2
3. Programme Educational Objectives (PEOs)	3
4. Programme Outcomes (POs)	4
5. Institutional Policy for Students' Conduct	5
6. Learning Outcomes of Laboratory Work	6
7. Course/Lab Outcomes (COs)	6
8. Mapping of COs with POs	7
9. Course/Lab Description	8
10. Grading Policy	9
11. Lesson Plan	10
12. Assignments	11

List of Abbreviations

BTL	Bloom's Taxonomy Level
CE	Communication Efficacy
CICP	Conduct Investigations of Complex Computing Problems
CK	Computational Knowledge
CO	Course Outcome
GDB	GNU Debugger
DAC	Departmental Advisory Committee
DDS	Design and Development of Solutions
I&E	Innovation and Entrepreneurship
I&T	Individual & Team Work
IQAC	Internal Quality Assurance Cell
LLL	Life-Long Learning
MTU	Modern Tool Usage
PA	Problem Analysis
PE	Professional Ethics
PEO	Programme Educational Objective
PMF	Project Management and Finance
PO	Programme Outcome
SEC	Societal and Environmental Concern

Declaration

Department	:	Department of Computer Science and Applications
Course, Year and the Semester to which Lab is offered	:	MCA - I Year, I Semester
Name of the Lab Course	:	Programming in C Lab.
Course Code	:	MCA-153
Version No.	:	10.0
Name of Course/Lab Teacher	:	Dr. Ritika Wason
Laboratory Manual Committee	:	<ol style="list-style-type: none"> 1. Mrs. Vaishali Joshi, Chairperson 2. Dr. Anupam Baliyan, Member 3. Dr. Ritika Wason, Member 4. Mrs. Tanya Pathak Garg, Member 5. Mr. Uttam Singh Bist, Member 6. Prof. P. S. Grover, Margdarshak 7. Mr. Amit Sharma, Alumni & Industry Expert 8. Dr. Ritika Wason, Concerned Subject Teacher, Convener
Approved by	:	DAC Date: 03/05/2019
Approved by	:	IQAC Date: 06/05/2019

Signature
(Course Teacher)

Signature
(Head of Department)

Signature
(IQAC Coordinator)

1. Vision of the Department

To become a Centre of excellence in the field of Computer Science and Applications, to contribute effectively in the rapidly changing global economy directed towards national development ensuring prosperity for the mankind.

2. Mission of the Department

- M1** To become a centre of excellence in the field of Computer Science and Applications and produce professionals as per global industry standards.
- M2** To foster innovation, entrepreneurial skills, research capabilities and bring all-round development amongst budding professionals.
- M3** To promote analytical and collaborative life-long learning skills, among students and faculty members involving all stakeholders.
- M4** To inculcate strong ethical values and professional behaviour while giving equal emphasis to social commitment and nation building.

3. Programme Educational Outcomes (PEOs)

The PEO's for the MCA programme are as follows:

- PEO1** Exhibit professional competencies and knowledge for being a successful technocrat.
- PEO2** Adopt creative and innovative practices to solve real-life complex problems.
- PEO3** Be a lifelong learner and contribute effectively to the betterment of the society.
- PEO4** Be effective and inspiring leader for fellow professionals and face the challenges of the rapidly changing multi-dimensional, contemporary world.

4. Programme Outcomes (POs)

PO1: Computational Knowledge (CK)

Demonstrate competencies in fundamentals of computing, computing specialisation, mathematics, and domain knowledge suitable for the computing specialisation to the abstraction and conceptualisation of computing models from defined problems and requirements.

PO2: Problem Analysis (PA)

Identify, formulate, and analyze complex real-life problems in order to arrive at computationally viable conclusions using fundamentals of mathematics, computer sciences, management and relevant domain disciplines.

PO3: Design and Development of Solutions (DDS)

Design efficient solutions for complex, real-world problems to design systems, components or processes that meet the specifications with suitable consideration to public health, and safety, cultural, societal, and environmental considerations.

PO4: Conduct Investigations of Complex Computing Problems (CICP)

Ability to research, analyze and investigate complex computing problems through design of experiments, analysis and interpretation of data, and synthesis of the information to arrive at valid conclusions.

PO5: Modern Tool Usage (MTU)

Create, select, adapt and apply appropriate technologies and tools to a wide range of computational activities while understanding their limitations.

PO6: Professional Ethics (PE)

Ability to perform professional practices in an ethical way, keeping in mind cyber regulations & laws, responsibilities, and norms of professional computing practices.

PO7: Life-Long Learning (LLL)

Ability to engage in independent learning for continuous self-development as a computing professional.

PO8: Project Management and Finance (PMF)

Ability to apply knowledge and understanding of the computing and management principles and apply these to one's own work, as a member and leader in a team, to manage projects in multidisciplinary environments.

PO9: Communication Efficacy (CE)

Ability to effectively communicate with the technical community, and with society at large, about complex computing activities by being able to understand and write effective reports, design documentation, make effective presentations, with the capability of giving and taking clear instructions.

PO10: Societal and Environmental Concern (SEC)

Ability to recognize and assess societal, environmental, health, safety, legal, and cultural issues within local and global contexts, and the consequential responsibilities applicable to professional computing practices.

PO11: Individual & Team Work (I&T)

Ability to work in multi-disciplinary team collaboration both as a member and leader as per need.

PO12: Innovation and Entrepreneurship (I&E)

Ability to apply innovation to track a suitable opportunity to create value and wealth for the betterment of the individual and society at large.

5. Institutional Policy for Students' Conduct

The following guidelines shall be followed:-

- 5.1 All the students in their introductory Lab. shall be assigned a system, which shall be their workplace for the complete semester. Students can store records of all their Lab. assignments on their individual workstations.
- 5.2 Introductory Lab. shall include an introduction to the appropriate software/tool, followed by a basic Introductory Assignment having Practice Questions. All the students are expected to complete this assignment within a week time, as the same shall be assessed through a lab. test.
- 5.3 Each week the instructor, in parallel to respective topics covered in the theory lecture, shall assign a set of practical problems to the students in form of Assignments (A, B, C,). The problems in these assignments shall be divided into two parts. The first set of Problems shall be compulsory for all the students and its record need to be maintained in the Practical File, having prescribed format, as given in Appendix-A. All the students should get the weekly assignment checked and signed in the Practical File by the respective teacher in the immediate succeeding week. The second set of problems is Advanced Problems and shall be optional. Student may solve these advanced problems for their further practice.
- 5.4 Cellular phones, pagers, CD players, radios and similar devices are prohibited in the classrooms, laboratories and examination halls.
- 5.5 Laptop-size computers / Tablets may be used in lectures for the purpose of taking notes or working on team-projects.
- 5.6 The internal practical exam shall be conducted towards the end of the semester and shall include the complete set of Lab exercises conducted as syllabus. However, students shall be assessed on continuous basis through overall performances in regular lab. tests, both announced and surprise and viva-voce.

- 5.7 The respective faculty shall prepare and submit sufficient number of practical sets of computing problems to the Dean (Examinations), atleast two weeks prior to the actual exam. It is the responsibility of the faculty to ensure that a set should not be repeated for more than 5 students in a given batch.
- 5.8 The exam shall be of 3 hours duration where the student shall be expected to implement solutions to his/her assigned set of problems on appropriate software tools in the lab.
- 5.9 Once implemented, student shall also appropriately document code implemented in the assigned answer sheets, which shall be submitted at the end of the examination. All the students shall also appear for viva-voce examination during the exam.
- 5.10 Co-operate, Collaborate and Explore for the best individual learning outcomes but copying or entering into the act of plagiarism is strictly prohibited.

6. Learning Outcomes of Laboratory Work

The student shall demonstrate the ability to:

- ☑ Verify and Implement the concepts and theory learnt in class.
- ☑ Code and use Software Tools to solve problems and present their optimal solutions.
- ☑ Apply numerical/statistical formulas for solving problems/questions.
- ☑ Develop and apply critical thinking skills.
- ☑ Design and present Lab as well as project reports.
- ☑ Apply appropriate methods for the analysis of raw data.
- ☑ Perform logical troubleshooting as and when required.

- ☑ Work effectively as a member of a team in varying roles as need be.
- ☑ Communicate effectively, both oral and written.
- ☑ Cultivate ethics, social empathy, creativity and entrepreneurial mindset.

7. Course/Lab Outcomes (COs)

- CO1 Develop programs in C for different algorithms using basic building blocks of the C language.(BTL6)
- CO2 Develop efficient programs using advanced data types, pointers and dynamic memory allocation functions.(BTL6)
- CO3 Implement real-world computing solutions through appropriate usage of the pre-processor as well as file handling on Ubuntu environment, using library functions and system calls.(BTL3)
- CO4 Apply C constructs to programming problems to control, manipulate files, directories and processes on Ubuntu.(BTL3)
- CO5 Work in teams to develop project for real-life cases.(BTL6)

8. Mapping of CO's with PO's

Table 1: Mapping of CO's with PO's

PO/CO	PO ₁	PO ₂	PO ₃	PO ₄	PO ₅	PO ₆	PO ₇	PO ₈	PO ₉	PO ₁₀	PO ₁₁	PO ₁₂
CO ₁	3	3	3	-	3	3	-	-	-	2	-	-
CO ₂	3	3	3	2	-	3	-	-	-	2	-	-
CO ₃	3	3	3	3	-	3	2	-	1	-	2	-
CO ₄	3	3	3	3	-	3	-	-	2	-	-	-
CO ₅	3	3	3	3	3	3	3	2	3	3	3	2

9. Course/Lab Description

Course (Lab) Title : Programming in C Lab.

Course (Lab) Code	: MCA-153
Credits	: 02
Pre-requisites	: A computer with GCC compiler, basics of mathematics, basic logic building skills
Academic Session	: July to December
Contact Hours/Week	: 04 (02 Labs of 02 hours each/Week)
Internal Assessment	: 40 Marks
External Assessment	: 60 Marks

10. Grading Policy

Item	Points	Marks	Remarks
Weekly Lab Assignments including Practical Files	10	10	Closed Book/Open Book
Internal End-Term Practical Examination	20	10	Closed Book
Viva-Voce	10	10	Closed Book
Project	10	10	Innovative Applications of Programming
External End-Term Examinations	60	60	Closed Book (conducted and evaluated by the University)
Total		100	

11. Lesson Plan

Week No.	Lab No.	Topics / Concepts to be Covered	Assignment No.
1.	1.	Syntax and structure of C-program structure on the Ubuntu environment alongwith usage of GCC.	Assignment A, Problems [AP ₁ -AP ₅]
	2.	Basic constructs of C programming along with valid expressions for problem solving.	Assignment A, Problems [AP ₆ -AP ₇], Advanced Problems [AA ₁ -AA ₂]
2.	3.	Usage of decision control, branching and looping structures	Assignment A, Advanced Problems [AA ₃ -AA ₆]
	4.	Usage of constants, operators etc	Assignment B, Problems [BP ₁ -BP ₅]
3.	5.	Pattern Programming	Assignment B, Problems [BP ₆], Advanced Problems [BA ₁ -BA ₂]
	6.	Arrays (Single-dimensional Arrays).	Assignment C, Problems [CP ₁ -CP ₆]
4.	7.	Arrays (Multi-dimensional Arrays).	Assignment C, Problems [CP ₇], Advanced Problems [CA ₁ -CA ₃]
	8.	String Handling	Assignment D, Problems [DP ₂ -DP ₇]
5.	9.	String Manipulation	Assignment D, Advanced Problems [DA ₁ -DA ₄]
	10.	Recursion and Command line parameters.	Assignment D, Problems [DP ₁ ,DP ₅], Advanced Problems [DA ₆]
6.	11.	Buffer reserved for Revision	Assignment A-D

Week No.	Lab No.	Topics / Concepts to be Covered	Assignment No.
	12.	New data types like structures and union	Assignment E, Problems [EP ₁ -EP ₄], Advanced Problems [EA ₁ -EA ₂]
7.	13.	Usage of Enums, bitfields or typedef	Assignment E, Problems [EP ₅ -EP ₆], Advanced Problems [EA ₃ -EA ₄]
	14.	Pointers.	Assignment F, Problems [FP ₁ ,FP ₇], Advanced Problems [FA ₂]
8.	15.	Dynamic memory allocation	Assignment F, Problems [FP ₂ -FP ₄],
	16.	File Handling and Manipulation.	Assignment F, Problems [FP ₅ -FP ₆], Advanced Problems [FA ₁ -FA ₂]
9.	17.	File Handling and Manipulation.	Assignment F, Advanced Problems [FA ₃ -FA ₆]
	18.	Buffer reserved for Revision	Assignment E-F
10.	19.	C Pre-processor	Assignment G, Problems [GP ₁ -GP ₅]
	20.	Usage of library functions	Assignment G, Problems [GP ₆], Advanced Problems [GA ₁ -GA ₃]
11.	21.	Buffer reserved for Revision	Assignment G
	22.	Directory Control and Manipulation through System Calls.	Assignment H, Problems [HP ₁ -HP ₃ , HP ₅], Advanced Problems [HA ₃ -HA ₄]
12.	23.	Process Control	Assignment H, Problems [HP ₄],

Week No.	Lab No.	Topics / Concepts to be Covered	Assignment No.
			Advanced Problems [HA ₁]
	24.	Thread Creation	Assignment H, Problems [HP ₆], Advanced Problems [HA ₂]
13.	25.	Miscellaneous Exercises	Assignment I, Problems [IP ₁ -IP ₈]
	26.	Buffer reserved for Revision	Assignment H-I

12. Assignments

Assignment Set: A

Objectives of the Assignment:

Familiarize with syntax and structure of C-programming.

Familiarize with problem solving using basic constructs of C Programming.

Test and execute the programs and correct syntax and logical errors.

CO/BTL Covered: CO₁/BTL₂ & BTL₃

Problems:

- AP₁* Calculate and display the volume of a CUBE. Test and execute the program for cubes having its side =10cm and 4 inch and 8cm respt.
- AP₂* Read in two integers from the user and display one as A% of the other. Your output should look like: 30 is 50% of 60.
- AP₃* Calculate the average of two numbers input by the user. Also test and print their standard deviation.
- AP₄* The distance between two cities (in KM) is input through the keyboard. Convert and print this distance in meters, feet, inches and centimetres.
- AP₅* Accept temperature in fahrenheit and convert it into equivalent

temperature in Celsius.

AP₆ Calculate and display the area for an unspecified number of circles. The calculations should continue until a value of zero is entered for the radius.

AP₇ Read in two integers from the user and calculate and display their sum without using the + operator.

AP₈ The length and breadth of a rectangle and radius of a circle are input through the keyboard. Calculate and display the area and perimeter of a rectangle and area and circumference of a circle

Advanced Problems:

AA₁ Design and develop a flowchart or an algorithm that takes three coefficients (a, b, and c) of a Quadratic equation ($ax^2+bx+c=0$) as input and compute all possible roots. Implement a C program for the developed flowchart/algorithm and execute the same to output the possible roots for a given set of coefficients with appropriate messages.

AA₂ Design and develop an algorithm that can be developed as a function that prints all the numbers between 1 and 100, with 10 numbers on each line. Use two for loops. All columns should be aligned.

AA₃ If a five digit number is input through the keyboard, write a program to print a new number by adding one to each of its digits. For Ex: if 12391 is input, output should be 23502.

AA₄ Calculate the bill amount for an item given its quantity sold, value, discount and tax.

Assignment Set: B

Objectives of the Assignment:

- Familiarize with problem solving using basic constructs of C Programming.
- Implement conditional branching, decision control and iteration.
- Decompose a problem into functions and synthesize a complete program using divide and conquer approach.

CO/BTL Covered: CO₁/BTL₂ & BTL₃

Problems:

- BP₁* Print an uppercase character to lowercase using ASCII value and vice versa.
- BP₂* Find the largest of two numbers entered by user without using the IF keyword.
- BP₃* Enter a number and check if the number is even or odd without using Arithmetic Operations.
- BP₄* Check if a number entered by the user is perfect or not.
- BP₅* Count the total number of digits in a number.
- BP₆* Define a c function that perform a logical XOR operation using the logical operators AND (&&), OR (| |) and NOT (!).

Advanced Problems:

BA₁ Print the following pattern:

```
1
1 2
3 5 8
13 21 34 55
89 144 233 377 610
```

BA₂ Print the following pattern:

```
1
232
34543
4567654
567898765
```

Assignment Set: C

Objectives of the Assignment:

- Familiarize with problem solving using basic constructs of C Programming.
- Create and Implement effective data handling through arrays.

- Decompose a problem into functions and synthesize a complete program using divide and conquer approach.

CO/BTL Covered: CO₁ & CO₂/BTL₂ & BTL₃

Problems:

- CP₁* Create a small simulation in c of the guess the magic number game.
- CP₂* Design a function to calculate the average of n numbers input by the user. Then compute the deviation of each number about the average.
- CP₃* Design a function that takes an integer array as parameter and returns a modified array that contains running sum of its original values. For example, if the original array had the values [3 2 4 7], after running the function the array would contain [3 5 9 6], and so on. Also develop a complete C solution to use this function.
- CP₄* Design and develop a solution to find the sum of a subset of the elements of an array. The starting and the ending limits are to be accepted from the user.
- CP₅* Three floating-point arrays are used to store marks secured by students in three class tests. The final marks secured by a student are to be calculated as the average of best-two. Write a program that calculates the final marks and stores them in another array Results.
- CP₆* Modify the above program so that the individual marks are stored in a 2D array, rather than three independent 1D arrays.
- CP₇* If a student is absent in a class, then the marks secured by him/her are set to -1. In such a case the marks of the student are calculated on the basis of those tests in which the student appeared. Modify the above program to incorporate this condition.

Advanced Questions:

- CA₁* Design and develop a flowchart or an algorithm for a function that accepts an integer parameter n. It then creates and returns an array of n arrays of integers, where array for row 0 has length 0; array for row 1 has length 1 and so on. Values in the array may remain uninitialized.

Implement a C program for the same and test for all possible inputs with appropriate messages

CA₂ A company store the information regarding the sales made for three of its products (PA, PB and PC) by five of its salespersons (SP1, SP2, SP3, SP4, SP5) in four different zones (N, E, S, W) in a 3D array. Here, the first dimension represents the products, the second dimension represents the salespersons and the third dimension represents the zones. Write down an interactive, menu driven solution that generates:

- a) Total sales made.
- b) Zone wise sales.
- c) Product wise sales.
- d) Sales made by each salesperson.
- e) Sales made by a particular salesperson for each product. (Get salesperson number form user)
- f) Sales made by a particular salesperson in each zone. (Get salesperson number form user)

Sales made by a particular salesperson for a product in a zone. (Get salesperson number, product & zone form user)

CA₃ Design and develop appropriate functions that:

- a) Counts the number of 0's in an integer array. The number of 0's is returned by the function.
- b) Prints an array of integers. The integers are printed on one line, enclosed in curly brackets and separated by commas.
- c) Triples the value of all elements in an array of integers.

All functions should take two parameters, the array of integers and the number of elements in the array.

Also develop a main function that asks the user to input 10 integers and stores them in an array. Use your other functions to print the initial array, the number of zero-valued elements in the array and the contents of the array when all elements have been tripled.

Assignment Set: D

Objectives of the Assignment:

- Familiarize with problem solving using basic constructs of C Programming.
- Create and Implement effective data handling through strings.
- Problem Solving through recursion.

CO/BTLCovered: CO₁ & CO₂/BTL₂ & BTL₃

Problems:

DP₁ Design and develop a function to accept a positive integer through the keyboard and find the binary equivalent of the same using recursion. Implement a C program for the same and test for all possible inputs with appropriate messages.

DP₂ Write a method that accepts an array of string objects and returns a single string that is resultant of concatenating all strings in the array separated by space. Do not add a space after the last element, nor before the first one. Implement a C program for the same and test for all possible inputs with appropriate messages.

DP₃ Sorts words in a string in lexicographical order. Implement a C program for the same and test for all possible inputs with appropriate messages.

DP₄ Convert a number to a word statement e.g. 1245 should get converted to One Thousand Two Hundred and Forty-Five. Implement a C program for the same and test for all possible inputs with appropriate messages.

DP₅ Display the countdown from a number entered by the user till 1. The program should only display the countdown if the user enters the string “display” at runtime. Use command line arguments to accept a number and a string from the user.

DP₆ Design a function to delete all vowels from a sentence. Assume that the sentence is not more than 80 characters long. Implement a C program for the same and test for all possible inputs with appropriate messages.

DP₇ Design a function that will read a line and delete from it all occurrences

of the word 'the'. Implement a C program for the same and test for all possible inputs with appropriate messages.

Advanced Problems

DA₁ Design and develop a function to count the number of occurrences of any two vowels in succession in a line of text. For, examples, in the sentence "Please read this application and give me gratuity" such occurrences are ea, ea, ui.

DA₂ Without using any library functions, design a C function

```
void append(char* str1, char* str2) { ... }
```

that takes as argument two strings str1, str2 and appends str2 to str1. After calling append, the pointer str1 is supposed to point to the concatenation of (the original) str1 and str2. The caller of append has to make sure that enough memory for the result of concatenation is available at the memory address that str1 points to.

DA₃ Design and develop a solution with appropriate library functions to check if the string input by the user is a valid IPv4 address or not.

DA₄ Design an algorithm that finds the maximum occurring character in a string input by the user. Develop appropriate functions to implement the above algorithm. Test the functions on atleast 5 different string inputs from the user. Identify the inputs where this algorithm fails.

DA₆ Design a recursive function to accept an input string from the user and find the first capital letter in the string.

Assignment Set: E

Objectives of the Assignment:

- Create and Implement effective data handling through use of definition of new data types like structures, union, Enums, bitfields or typedef as required.

CO/BTLCovered: CO₁ & CO₂/BTL₂ & BTL₃

Problems:

EP₁ Design a menu driven solution that depicts the working of a library. The menu options should be:

- (a) Add book Information
- (b) Display book information
- (c) List all books of given author
- (d) List the title of specified book
- (e) List the count of books in the library
- (f) List the books in the order of accession number
- (g) Exit

Create a structure called library to hold accession number, title of the book, author name, price of the book, and flag indicating whether book is issued or not.

EP₂ Create a structure to specify data on students enlisted below: Roll number, Name, Department, Course, Year of joining. Assuming that there are not more than 100 students in the college.

- (a) Write a function to print names of all students who joined in a particular year.
- (b) Write a function to print the data of a student whose roll number is given.

EP₃ Design a function to compare two given dates by the user. To store a date, use a structure that contains three members namely date, month and year. If the dates are equal, then display message as "Equal" otherwise "Unequal".

EP₄ Define two structures Rectangle and Oval. Use these to define a union shape. Write a function CalculateArea() that takes a parameter of type Shape and returns area of appropriate Shape.

EP₅ Accept/ receive an integer as an input. The output should be a sentence that gives the English name for the weekday associated with the integer. If the user enters an integer not associated with the weekday the program prints, "This is not a day of the week."

EP₆ Optimize the date struct using bitfields. Also display the memory

optimization above.

Advanced Problems

- EA₁* Write a menu driven program to perform Create, Read, Update and Delete (CRUD) operations on an array of Student structures defined and created in question *EP2* above.
- EA₂* Create a structure Distance that can accept a distance in inches and feet. Design appropriate functions that add two distances in inch and feet and display the sum.
- EA₃* Create a structure to represent a complex number. Design functions that accept two complex numbers as arguments and returns their sum. Test the function appropriately.
- EA₄* Create a structure to represent time. Design a function named differenceBetweenTime that displays the difference between two time periods.

Assignment Set: F

Objectives of the Assignment:

- Create and Implement efficient solutions to real world problems using pointers and dynamic memory allocation.
- Solve real world problems through application of C programming constructs to store and manage data appropriately in required files.

CO/BTLCovered: CO₁ to CO₃/BTL₂ to BTL₆

Problems:

- FP₁* Create an effective solution to analyze the number of words, digits and vowels in a string entered by the user using pointers.
- FP₂* Compose appropriate functions to implement following operations of dynamic Stack.
- (a) Push the Element
 - (b) Pop the Element
 - (c) Display

(d) Exit

FP₃ Write functions to create two dimensional arrays dynamically using:

(a) Pointer to an Array

(b) Array of Pointers

(c) Pointer to a Pointer

The number of rows and columns as well as their values should be accepted from the user.

FP₄ Develop functions to perform following operations on a Linked List:

(a) Add a node at any position

(b) Search for some data value

(c) Delete a node from any position

FP₅ Design and develop solution to read a file and display its contents along with line numbers before each line.

FP₆ Develop question *EP1* above to store and retrieve records of books of a library in suitable file.

FP₇ Design and develop appropriate functions to create and populate two arrays from the user. Once the arrays are displayed their contents should be swapped. Pointers should be used to achieve the above scenario.

Advanced Problems

FA₁ Develop question *EP2* above to store and retrieve student's record in suitable file. Fetch each record containing name and year of joining a student display them in sorted order by name and year of joining.

FA₂ Design the function `threeColorsSort()` that takes as input an array of integers in the range of 0 and 2 (0, 1 and 2 only), and arranges them in an increasing order:

```
void threeColorsSort(int * theArray, int arraySize)
```

Your solution should have linear runtime in the parameter `arraySize`. Then, develop a program that asks the user for how many numbers to input, and then for the actual numbers. The program should then output the same numbers in ascending order. Make sure to free up your

allocated memory.

FA₃ Design and develop appropriate function to encrypt the contents of a text file.

FA₄ Design a solution to count and display the number of characters in a text file input by the user.

FA₅ Design a file format to store a person's name, address, and other information. Write a program to read this file and produce a set of mailing labels.

FA₆ Design a solution to delete a specific line of text from a file.

Assignment Set: G

Objectives of the Assignment:

- Implement real-world, optimized computing solutions through appropriate usage of the pre-processor Ubuntu environment, using library functions.

CO/BTLCovered: CO₁ to CO₃/BTL₂ to BTL₆

Problems:

GP₁ Demonstrate the use of callback functions through library function `qsort()` and `bsearch()` to search for a student in an array of student structure on basis of rollno.

GP₂ Design a solution to fetch and display the current system time using appropriate macro.

GP₃ Design a solution to ensure calling a function before terminating a program.

GP₄ Design a solution to accept a float value from the user and round it off to the nearest integer using Macros.

GP₅ Design a solution that prints the error message from any function on compilation alongwith the function name and line number of the error.

GP₆ Design a solution to print the name of the currently executing function in

given code.

Advanced Problems

- GA₁** Design a solution that disallows the use of puts in code. The compiler should display an error message wherever puts is used.
- GA₂** Design a solution to find the factorial of a number entered by the user. Debug your code step by step using the gnu debugger. Record your steps.
- GA₃** Design a makefile to automate code compilation of all code files above.

Assignment Set: H

Objectives of the Assignment:

Implement real-world computing solutions through appropriate usage of the pre-processor as well as file manipulation on Ubuntu environment, using system calls.

CO/BTLCovered: CO₁ to CO₄/BTL₂ to BTL₆

Problems:

- HP₁** Use system calls to read an inode of a file whose name is passed as command line parameter. The program should display the rights, the owner, group members and others have on the file.
- HP₂** Provide appropriate menu options to change the ownership, group ownership and permissions on the file. The required data can be accepted interactively from the user.
- HP₃** Design and develop a solution that changes the current working directory of the currently executing process to its root.
- HP₄** Create a child process and displays the process id and parent process ids of both the parent and the child process. The parent process should display its data after only the child process has terminated.
- HP₅** Displays the contents of the directory (using system) that is passed as command line parameter. The program should then display a menu

providing options to copy, move or delete the file. The desired operation should be carried through a child process whose memory area is overlaid with the required shell command.

HP₆ Create a thread to calculate sum. Accept the integer number for which sum has to be calculated as a command line argument.

Advanced Problems:

HA₁ Demonstrate zombie process creation through forking a child process, that terminates immediately and then the parent goes to sleep for a minute, without ever cleaning up the child process.

HA₂ Create three threads such that they print the sequence 1 2 3 infinitely.

HA₃ Design a solution to print the details of all the variables of the GNU-enabled execution environment.

HA₄ Design and develop a solution to write a memory buffer to a temporary file (so that memory can be freed or reused) and then read it back later.

Assignment Set: I

Objectives of the Assignment:

Apply all C programming constructs and mechanisms to solve problems in an optimized manner.

CO/BTLCovered: CO₁ to CO₄/BTL₂ to BTL₆

Problems:

IP₁ Create a structure to specify data of customers in a bank. The data to be stored is: Account number, Name, Balance in account. Assume maximum of 200 customers in the bank.

(a) Design a function to print the Account number and name of each customer with balance below Rs. 100.

(b) If a customer request for withdrawal or deposit, it is given in the form: Acct. no, amount, code (1 for deposit, 0 for withdrawal). Design a program to give a message, "The balance is insufficient for the specified withdrawal".

- IP₂* An automobile company has serial number for engine parts starting from AA0 to FF9. The other characteristics of parts to be specified in a structure are: Year of manufacture, material and quantity manufactured.
- (a) Specify a structure to store information corresponding to a part.
- (b) Write a program to retrieve information on parts with serial numbers between BB1 and CC6
- IP₃* Design and develop a function that merges lines alternately from two files and writes the results to new file. If one file has less number of lines than the other, the remaining lines from the larger file should be simply copied into the target file. Design a complete solution using the above function.
- IP₄* Design a program to carry out the following:
- To read a text file “TRIAL.TXT” consisting of a maximum of 50 lines of text, each line with a maximum of 80 characters.
 - Count and display the number of words contained in the file.
 - Display the total number of four letter words in the text file. Assume that the end of a word may be a space, comma or a full-stop followed by one or more spaces or a newline character
- IP₅* Create an array of four function pointers. Each pointer should point to a different function. Each of these functions should receive two integers and return a float. Through a loop call each of the functions through the addresses stored in the array.
- IP₆* Write a function that receives a variable number of arguments, which are the coordinates of a point. Based on the number of arguments received, the function should classify the type of shape like a point, line, triangle etc.
- IP₇* Design a function that accepts information about a date in an appropriate structure containing three members namely day, month and year. Ensure through bitfields that the day is stored in first five bits of day, month in first 4 bits of month and year in first 12 bits of year. Design a complete solution to accept the date of joining of 10 employees and display them

in ascending order of year.

IP₈ Design a program that can be used at command prompt as a calculating utility. The usage of the program is shown below. C> calc <switch> <n> <m> Where, n and m are two integer operands. Switch can be any one of the arithmetic or comparison operators. If arithmetic operator is supplied, the output should be the result of the operation. If comparison operator is supplied then the output should be True or False

Minor Project in C

Objectives of the Assignment: Solve/Simulate a real-world problem through an efficient solution using all features of the C programming language.

CO/BTLCovered: CO₁ to CO₅/BTL₂ to BTL₆

S. No.	Suggested Topics for Project
--------	------------------------------

- | | |
|-----|---|
| 1. | Hangman Game |
| 2. | MCD Multi-level Parking & Billing Utility |
| 3. | AC Feasibility Calculator |
| 4. | Alarm Clock Utility |
| 5. | Login-Logout Tracker |
| 6. | See your C Tester |
| 7. | Power Vocab |
| 8. | Sales Data Forecaster |
| 9. | Automatic Traffic Challan Generator Utility |
| 10. | Network Mapper |
| 11. | Snake Mania Game |
| 12. | Metro Route Tracer Utility |
| 13. | File Encryption-Decryption Utility |
| 14. | Horoscope Utility |
| 15. | Sudoku Game |

S. No.	Suggested Topics for Project
--------	------------------------------

- | | |
|-----|----------------------------|
| 16. | Client Server Chat Utility |
| 17. | Password Manager |
| 18. | Food Ordering Utility |
| 19. | Typing Game |
| 20. | KBC Quiz Game Utility |

APPENDIX-A

Template for the Index of Lab File

WEEK NO.	PROBLEMS WITH DESCRIPTION		PAGE NO.	SIGNATURE OF THE TEACHER WITH DATE
1	AP ₁			
	AP ₂			
	AP ₃			
2	AA ₁			
	AA ₂			
	AA ₃			
3	BP ₁			
	BP ₂			
	BP ₃			
	BP ₄			

Note: The students should use Header and Footer mentioning their roll no. & name in header and page no. in footer