Laboratory Manual

(Version 12.0)

for

Data and File Structures Lab.

(MCA-162)

MCA - II Semester

Compiled by:

Dr. Sunil Pratap Singh

(Assistant 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

BTL	Bloom's Taxonomy Level
CE	Communication Efficacy
CICP	Conduct Investigations of Complex Computing Problems
CK	Computational Knowledge
CO	Course Outcome
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 Envoirnmental Concern

Declaration

Department	:	Department of Computer Science and Applications
Course, Year and the Semester to which Lab is offered	:	MCA - I Year, II Semester
Name of the Lab Course	:	Data and File Structures Lab.
Course Code	:	MCA-162
Version No.	:	12.0
Name of Course/Lab Teacher(s)	:	Dr. Sunil Pratap Singh
Laboratory Manual Committee	:	1. Mrs. Vaishali Joshi, Chairperson 2. Mrs. Parul Arora, Member 3. Dr. Ritika Wason, Member 4. Mr. Manish Kumar, Member 5. Mr. Uttam Singh Bist, Member 6. Prof. P. S. Grover, Margdarshak 7. Mr. Amit Sharma, Alumni & Industry Expert 8. Dr. Sunil Pratap Singh, Concerned Subject Teacher, Convener
Approved by	:	DAC Date: March, 2022
Approved by	:	IQAC Date: March, 2022

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 produce quality professionals in software development.

2. Mission of the Department

M_1 To produce quality software professionals as per global industry standards.

M_2 To foster innovation, entrepreneurial skills, research capabilities and bring all-round development amongst budding professionals.

M_3 To promote analytical and collaborative life-long learning skills, among students and faculty members.

M_4 To inculcate strong ethical values and professional behaviour while giving equal emphasis to social commitment and nation building.

3. Programme Educational Objectives (PEOs)

The PEOs for the MCA programme are as follows:

PEO_1 Exhibit professional competencies and knowledge for being a successful technocrat.

PEO_2 Adopt creative and innovative practices to solve real-life complex problems.

PEO_3 Be a lifelong learner and contribute effectively to the betterment of the society.

PEO_4 Be effective and inspiring leader for fellow professionals and face the challenges of the rapidly changing multi-dimensional, contemporary world.

4. Programme Outcomes (POs)

PO_1 Computational Knowledge (CK)

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

PO_2 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.

PO_3 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.

PO_4 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.

PO_5 Modern Tool Usage (MTU)

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

PO_6 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.

PO_7 Life-Long Learning (LLL)

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

PO_8 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.

PO_9 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.

PO_{10} Societal and Envoirnmental 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.

PO_{11} Individual & Team Work (I&T)

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

PO_{12} 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 Prcatical File, having prescribed format, as given in Appendix-A. All the students should get the weekly assigntment 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 continuos 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)

CO_1 Illustrate basic data structures - arrays and linked lists. (BTL2)

CO_2 Build stacks and queues using arrays and linked lists. (BTL3)

CO_3 Discover sparse matrix, polynomial arithmetic, searching and sorting techniques and their applications. (BTL4)

CO_4 Appraise binary search tree to perform efficient search operations. (BTL4)

CO_5 Examine and implement graph algorithms. (BTL4)

CO_6 Develop an application making extensive use of binary files. (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	-	-	-	-	-	-	-	-	-	-	-	-
CO_2	-	-	-	-	-	-	-	-	-	-	-	-
CO_3	-	-	-	-	-	-	-	-	-	-	-	-
CO_4	-	-	-	-	-	-	-	-	-	-	-	-
CO_5	-	-	-	-	-	-	-	-	-	-	-	-
CO_6	-	-	-	-	-	-	-	-	-	-	-	-

9. Course/Lab Description

Course (Lab) Title	:	Data and File Structures Lab.
Course (Lab) Code	:	MCA-162
Credits	:	01
Pre-requisites	:	Programming in 'C', Basics of Tree/Graph Theory
Academic Session	:	January to June
Contact Hours/Week	:	02 (01 Lab of 02 Hours/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	Reference of Lab Manual
1.	1.	Array (One Dimensional)	AP_1
	2.	Searching and Sorting in Array	AP_2 - AP_5
2.	3.	Sparse Matrix	AA_1
	4.	Operations on Matrix (Two Dimensional)	AA_2
3.	5.	Linked List	BP_1
	6.	Buffer reserved for Revision	-
4.	7.	Polynomial Arithmetic	BP_2
	8.	Stack and Double Stack	CP_1, CA_1, CA_5
5.	9.	Queue (Linear)	CP_2, CA_6
	10.	Queue (Circular and Double Ended)	CP_3, CA_7
6.	11.	Applications of Stack and Queue	CA_2 - CA_4
	12.	Buffer reserved for Revision	-
7.	13.	Binary Tree and Traversal of Tree	DP_1, DA_3
	14.	Priority Queue and Heap Sort	DP_2, DP_4
8.	15.	Binary Search Tree	DP_3, DA_1,
	16.	Graph Implementation using Adjacency Matrix	EP_1 - EP_4
9.	17.	Graph Implementation trough Linked List	EA_1
	18.	Buffer reserved for Revision	-
10.	19.	Graph Traversal (BFS and DFS)	EP_2
	20.	Minimum Cost Spanning Tree	EA_1, EA_3
11.	21.	Impelemenattion of Shortest Path Algorithms	EP_2- EP_4
	22.	File Handling	FP_1 - FP_2, FA_1
12.	23.	File Handling	FP_1 - FP_2, FA_1
	24.	Hashing	FP_3
13.	25.	Buffer reserved for Revision	-

12. Assignments

Assignment Set: A (Searching and Sorting in Array)

Problems:

AP_1 Write a program which takes an array of n integers and displays the frequency of each element present in the array.

AP_2 Write a program which takes an array of n integers and performs searching of an element by implementing linear search and binary search techniques.

AP_3 Write a program which takes an array of n integers and sorts the integers in descending order using bubble sort and selection sort techniques.

AP_4 Write a program which takes an array of n integers and sorts the integers in ascending order using insertion sort technique.

AP_5 Write a program which takes an array of n integers and sorts the integers in ascending order using quick sort technique.

Advanced Problems:

AA_1 Write a program which receives a matrix $a[m][n]$ (represented by a two-dimensional array) of integers and determines whether the given matrix is sparse matrix or not. If it is sparse matrix, represent it through appropriate representation with array to save the memory space.

AA_2 Write a program which receives a matrix $x[n][n]$ (represented by a two-dimensional array) of integers and perform the following operations on the received matrix:

a) Transpose the matrix
b) Determine the row-wise sum
c) Determine the column-wise sum
d) Determine the sum of off-diagnoal elements
e) Display the upper triangular matrix
f) Display the lower triangular matrix
g) Display the diagonal elements only

Assignment Set: B (Linked List)

Problems:

BP_1 Write a menu-driven program which implements a linear linked list with following operations:

a) Insertion of an element at beginning of the list
b) Insertion of an element at specific location of the list
c) Insertion of an element at end of the list
d) Deletion of an element from the beginning of the list
e) Deletion of an element from specific location of the list
f) Deletion of an element from the end of the list
g) Display all elements of the list
h) Search a specific element in the list

BP_2 A polynomial is composed of different terms where each of them holds a coefficient and an exponent. Write a program to represent the following polynomials: $4x^4 + 4x^3 - 2x^2 + x$ and $11x^3 + 7x^2 - 4x$ with linear linked list, and then perform addition of the given polynomials.

Advanced Problems:

BA_1 Write a program to implement doubly linked list of integers, copy the odd integers in one list and the even integers in another list.

BA_2 Write a program to implement the following operations in a linear linked list:

a) Remove all occurance of a given element
b) Remove all duplicate elements
c) Sort the elements in ascending order
d) Reverse the list

BA_3 There is a need to store information about the smart watches that are sold by an online e-commerce company. To do so, a list of products' records is maintained, each of which contains information about a single type of product, including how many quantities of it are available in stock. There is no need to change the types of watches that are sold, and thus watches

are seldom added or removed from the list. However, there is need to update a watch's record whenever inventory of that watch changes due to purchase by someone. A watch's position in the list is also its Id, and therefore updating a record involves using the watch Id to get the corresponding record and modifying its contents.

Assignment Set: C (Stack and Queue)

Problems:

CP_1 Write a menu-driven program which implements a stack (*using one-dimensional array*) with following operations:

a) Push (insert an element)
b) Pop (delete an element)
c) Display (print all the elements of stack)

CP_2 Write a menu-driven program which implements a linear queue (*using one-dimensional array*) with following operations:

a) Enqueue (insert an element)
b) Dequeue (delete an element)
c) Display (print all the elements of queue)

CP_3 Write a menu-driven program which implements a circular queue (*using one-dimensional array*) with following operations:

a) Enqueue (insert an element)
b) Dequeue (delete an element)
c) Display (print all the elements of queue)

Advanced Problems:

CA_1 Write a program that implements two stacks (double stack) in a single array. Both stacks should grow dynamically in opposite directions, one from lower index to higher index and second from higher to lower index.

CA_2 Write a program that accepts a string of characters and prints the characters in reverse order. The program should make use of appropriate operations of stack.

CA_3 Write a program that accepts a number in decimal and converts it to its equivalent binary number. The program should make use of appropriate operations of stack.

CA_4 Write a program that accepts a postfix expression of the form **ab+cd-*ab/** and evaluate it after accepting the values of a, b, c and d. The program should make use of appropriate stack operations.

CA_5 Write a menu-driven program which implements a stack (*using linear linked list*) with following operations:

a) Push (Insert an element)
b) Pop (Delete an element)
c) Display (Print all elements of stack)

CA_6 Write a menu-driven program which implements a linear queue (*using linear linked list*) with following operations:

a) Enqueue (Insert an element)
b) Dequeue (Delete an element)
c) Display (Print all elements of queue)

CA_7 Write a menu-driven program which implements a double ended queue (*using one-dimensional array*) with all possible insertion and deletion operations.

Assignment Set: D (Tree)

Problems:

DP_1 Write a menu-driven program which implements a binary tree (*using linked list*) with following operations:

a) Insertion of a node
b) Deletion of a node
c) Preorder traversal
d) Inorder traversal
e) Postorder traversal
f) Determine total numer of leaf nodes

DP_2 Write a menu-driven program which implements a heap (*using one-dimensional array*) with following operations:

a) Insertion of a node
b) Deletion of a node
c) Display (print all the elements of heap)

DP_3 Write a menu-driven program which implements a binary search tree (*using linked list*) with following operations:

a) Insertion of a node
b) Deletion of a node
c) Preorder traversal
d) Inorder traversal
e) Postorder traversal

DP_4 Write a program which takes an array of n integers and sorts the integers in ascending order using heap sort technique.

Advanced Problems:

DA_1 Write a menu-driven program that implements a binary search tree with following operations:

a) Total number of nodes in the tree
b) Total number of leaf nodes in the tree
c) Total number of non-leaf nodes in the tree
d) Smallest node of the tree
e) Largest node of the tree
f) Search a node in the tree
g) Sum of all nodes of the tree
h) Display the nodes in ascending order

DA_2 Write a program to implement priority queue (*using one dimensional array*). For deletion of a node, the node having smallest data value will be considered to have highest priority.

DA_3 Write a program to implement a binary tree and perform preorder, inorder and postorder traversal without performing recursive operations.

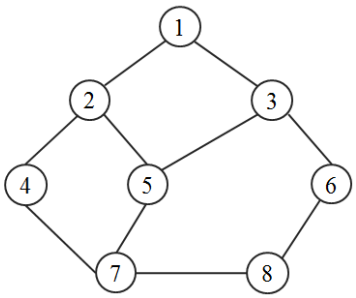
Assignment Set: E (Graph)

Problems:

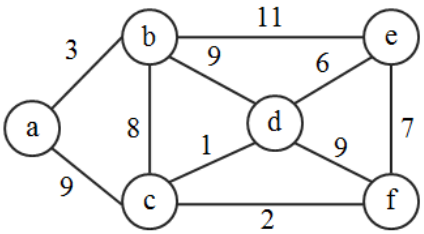
EP$_1$ Write a menu-driven program which implements a graph (*using adjacency matrix*) with following operations:

g) Insertion of a vertex
h) Insertion of an edge
i) Deletion of a vertex
j) Deletion of an edge
k) Calculation of degree of each vertex
l) Calculation of number of self-loops in the graph

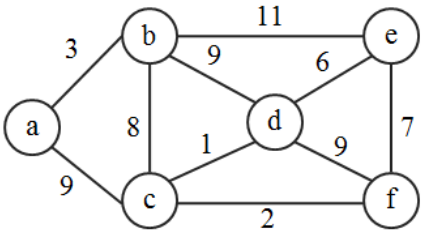
EP$_2$ Write a program to travers the following graph using breadth first search and depth first search techniques.



EP$_3$ Write a program to determine shortest path from *a* to *f* (using Dijkstra's algorithm) in the following graph.



EP$_4$ Write a program to determine shortest paths between every pair of vertices (using Floyd Warshell's algorithm) in the following graph.

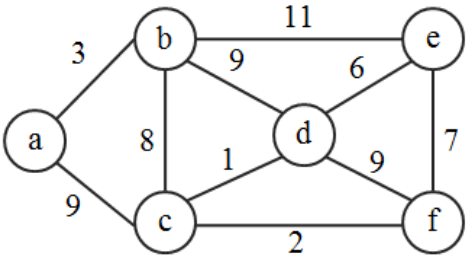


Advanced Problems:

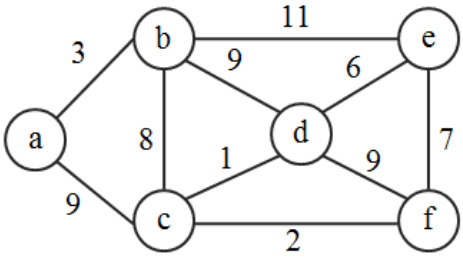
EA$_1$ Write a menu-driven program which implements a graph (*using adjacency list*) with following operations:

a) Insertion of a vertex
b) Insertion of an edge
c) Deletion of a vertex
d) Deletion of an edge
e) Calculation of degree of each vertex

EA$_2$ Write a program to determine the minimum cost spanning tree (using Prim's algorithm) in the following graph.



EA$_3$ Write a program to determine the minimum cost spanning tree (using Kruskal's algorithm) in the following graph.



Assignment Set: F (File Handling)

Problems:

FP$_1$ Write a program that generates *n* random integers and stores them in a text file, named as "All.txt". Then, retrieve the stored integers from this file and copy to "Odd.txt" and „Even.txt" based upon the type of number, i.e. if the retrieved integer is odd number then store in "Odd.txt" file or if the retrieved integer is even then store in "Even.txt" file. Finally, display the contents of all three files.

FP_2 A text file contains student's grade, followed by student's name. Sample data is following:

8.3 Gautam

9.4 Jasleen

6.7 Gaurav

9.4 Naman

5.7 Ishika

7.5 Rakesh

Write a program to find the the highest grade, and list all the students who have highest grade. Also, list the details of students' having 3rd highest grade.

FP_3 Write a program to implement 20 integers (with some duplicate integers) in a hash table (with separate chaining for collision resolution).

Advanced Problems:

FA_1 Consider the list (file) of books maintained in a library system. When a user wants to check whether a particular book is available, a search operation is called for. If the book is available and is issued to the user, a delete operation can be performed to remove this book from the set of available books. When the user returns the book, it can be inserted back into the set of available books. Write a C program to solve the problem by using appropriate data structures. It is essential that the above mentioned operations should be performed efficiently as possible since these operations are performed frequently.