C Programming Lab Manual – Summer Project Report

Name: Soumya Jain

Roll Number: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Course: M.Tech CSE

Institute: IIIT Bangalore

Professor/Mentor: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date of Submission: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Certificate

This is to certify that Soumya Jain has successfully completed the summer project titled 'C Programming Lab Manual' under my guidance and supervision. This project has been submitted in partial fulfillment of the requirements for the summer project.

Signature of Guide

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_

# Abstract

This project involved designing and compiling a comprehensive lab manual for the C Programming language. It includes concept-based examples, macro usage, structure manipulation, pointer operations, and logical operator precedence. All programs are compiled and managed using a portable Makefile to enable ease of execution on any Linux system.

# Objective

- To reinforce core C programming concepts through code.  
- To create reusable and organized code examples.  
- To automate compilation via a system-independent Makefile.  
- To prepare students for practical lab exams.

# Tools and Technologies Used

- GCC Compiler via WSL Ubuntu  
- Bash / Makefile  
- Linux File System  
- Online compilers (for verification)

# Project Description

The project includes a set of organized C programs demonstrating core topics of the C language. Programs are separated by concept such as arrays, loops, structures, pointers, macros, and operators. A Makefile is provided to compile all programs independently into executables using GCC. Folder structure includes:  
  
- `src/`: Contains all `.c` and `.h` files  
- `bin/`: Stores compiled executables  
- `Makefile`: Automates compilation  
- `README.txt`: Contains compilation and execution instructions

# Sample Output Screenshots

Screenshots or outputs of at least 2 programs should be added here by the student.

# Conclusion

This project provided hands-on experience in writing, managing, and compiling multiple C programs. It strengthened understanding of Makefile automation, program modularity, and core programming concepts. The lab manual serves as a strong base for future academic and interview preparation.

# Future Scope

- Add problem statements or questions for each code.  
- Include error explanation and handling in comments.  
- Add test scripts for verifying output correctness.

# Appendix

Sample program list:  
- `arrays.c` – Demonstrates basic arrays  
- `pointers.c` – Pointer arithmetic  
- `structures.c` – Nested structures and access  
- `macros.c` – Use of macros and PRINT wrappers  
- `logic\_ops.c` – Logical operator precedence  
- ... etc.

Include screenshots of folder structure and Makefile usage if needed.