**PROJECT REPORT ON C PROGRAM ON TITLE  
 GURU PRABANDHAN**

**CAPITAL COLLEGE AND RESEARCH CENTER**

Koteshwor, Kathmandu, Tel: 5100423, 5100456

Email: [info@ccrc.edu.np](mailto:info@ccrc.edu.np), Web: www.ccrc.edu.np



**DEPARTMENT OF COMPUTER SCIENCE**

**Submitted By:**

Manish Singh Mahato

Class: XII, Sec: P5

Stream: Science

Symbol No.: \_\_\_\_\_\_\_\_\_\_\_

**Submitted To:**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Mr. Ranjan Dulal

HOD

Computer Science Department

***For the partial requirement for fulfilment of the degree of***

**National Examinations Board (NEB) Examination**

**24th Falgun, 2080 B.S.**

**LETTER OF APPROVAL**

This is to certify that this project report prepared by **Mr. Manish Singh Mahato** entitled *“Project Report on C program to manage teachers’ information”* in partial fulfillment of the requirements for the degree of National Examinations Board (NEB) examination, has been well studied. It is satisfactory in scope and quality as in project report for the required degree.

**Evaluation Committee**

………….………………………….... …………..…………………………

**Program Coordinator HOD**

Mr. Rajendra Pokhrel Mr. Ranjan Dulal

Administrative Department (Science) Department of Computer Science

Capital College and Research Center Capital College and Research Center

Koteshwor, Kathmandu Koteshwor, Kathmandu

…………..…………………………

**External Examiner**

Mr. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

N.E.B.

Kathmandu, Nepal

**LETTER OF DECLARATION**

I, Manish Singh Mahato student of Capital College and Research Center (CCRC), hereby declare that the project report entitled **"*****Project Report on C program to manage teachers’ information* "** submitted by me to the College and National Examinations Board (N.E.B.) in partial fulfillment of the requirement for the award of N.E.B. examination is a record of project work carried by me under the guidance of (Program Coordinator) Mr. Rajendra Pokhrel, (HOD) Mr. Ranjan Dulal, and (Lecturers) Mr. Abhishek Dhungel, Mr. Sajan Kharel & Mr. Bivash Kumar Shah sir.

I further declare that this report is the original and self-done work and is not subject to anyone else's work, apart from references.

Name: Manish Singh Mahato

Class: XII, Sec: P5

Stream: Science

Symbol No.: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Capital College and Research Center

Koteshwor, Kathmandu

**ACKNOWLEDGEMENT**

This is my project report on C programming. I would like to express my deepest gratitude to Mr. Rajendra Pokhrel Sir, the Program Coordinator of Science Stream & consider myself to be very fortunate to get an opportunity to study, work and pursue my research under the guidance of my respected teachers (HOD) Mr. Ranjan Dulal, (Lecturers) Mr. Abhishek Dhungel, Mr. Sajan Kharel & Mr. Bivash Kumar Shah sir. and (Computer Lab Teachers) Mr. Umesh Joshi as well as Mr. Sagar Shreshta Sir who guided me throughout the report and helped me to find rooms for improvement which made the report a lot finer. Hence, first and foremost, I express my deepest sense of gratefulness towards them.

I wish to express my sincere gratitude to the principal of Capital College and Research Center, Mr. Hari C. Lamichhane Sir and all the teachers at CCRC, who gave me their utmost platform and cooperation to succeed in completing this learning process.

The aim of this report is to illustrate the project’s subject matter, background knowledge of the subject, development and progress which will be beneficial for individuals who are interested in working on the C programming Concept.

Although this report has been prepared with utmost care and deep routed interest, even then I accept respondent and imperfection.

Thank You.

Name: Manish Singh Mahato

Class: XII, Sec: P5

Stream: Science

Symbol No.: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Capital College and Research Center

Koteshwor, Kathmandu

**TABLE OF CONTENT**

INTRODUCTION TO C PROGRAM AND ITS FEATURES …….…………………. 1

OBJECTIVE ………………..…….………………………………...………………….. 2

SYSTEM REQUIREMENTS …………………………....……......…………………… 3

C PROGRAMMING COMPILATION PROCESS ……….……..…………………….. 4

DEVELOPMENT OF “Guru Parbandhan” …………………………………………….. 5

WORKFLOW ………………………………………………………….....…………… 16

OUTPUT & FINDINGS …………………………………………..…………..……….. 17

CONCLUSION …………………………………………………….…………………. 19  
REFERENCE ………………………………………………………………………… 20

**C Programming & Features**

C programming is a powerful language known for its simplicity, efficiency, and versatility. It provides a robust framework for developing a wide range of software applications, from simple command-line utilities to complex system-level programs and high-performance applications.

One of the key features of C programming is its close-to-hardware abstraction, allowing developers to access low-level system resources and optimize performance.

Additionally, C offers a rich set of built-in functions and libraries, providing developers with the tools they need to accomplish various tasks efficiently. Its syntax is concise and flexible, making it easy to write clear and concise code.

Furthermore, C is highly portable, allowing programs written in C to run on different platforms with minimal modification. Overall, C programming offers a balance of power, flexibility, and simplicity, making it a preferred choice for developing software in diverse domains such as embedded systems, operating systems, games, and more.

The C programming language boasts a robust standard library that offers developers a comprehensive suite of tools for building efficient and versatile software. This library encompasses a wide array of functionalities, including input/output operations, string manipulation, memory management, mathematical computations, date and time handling, utility functions, character manipulation, and error handling mechanisms. For instance, developers can leverage functions like **printf()** and **scanf()** for formatted input and output, **strcpy()** and **strcmp()** for string manipulation, and **malloc()** and **free()** for dynamic memory allocation

**OBJECTIVE**

To create a menu-driven C program for teacher management.

To handle the file on disk.

To implement the switch case for menu.

**SYSTEM TOOLS AND SOFTWARE REQUIREMENTS**

**Tools and Technologies Used:**

**IDE: -** Dev c++

**Compiler: -** GCC (GNU Compiler Collection)

**Operating System: -** Windows 7,10,11

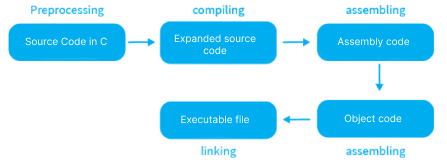
**System Requirements:**

**Processor: -** Any 64-bit processor

**RAM: -** 4 GB or higher

**Storage: -** A few gigabytes of free storage

**C Compilation Process**



The compilation process in C programming involves several steps that transform human-readable source code into machine-executable binary code.

Firstly, a preprocessor scans the source code, handling directives like #include and performing macro substitutions. Next, the preprocessed code is passed to the compiler, which translates it into assembly code specific to the target architecture.

The assembler then converts the assembly code into object code, consisting of machine instructions and data. Subsequently, the linker combines multiple object files, resolves external references, and generates an executable file.

During this process, libraries containing precompiled object code may be linked to the executable.

Finally, the loader loads the executable into memory, making it ready for execution by the CPU. Any errors encountered during compilation, such as syntax errors or undefined references, are reported to the user, who can then revise the source code accordingly. Overall, the compilation process in C programming is a crucial series of steps that translate source code into executable binaries, enabling the execution of C programs on diverse computing platforms.

**DEVELOPMENT ENGAGEMENT**

**Q. 1. Menu Driven C Program to manage teacher information with user**

**Authentication**

**CODE:**

#include <stdio.h>

#include <ctype.h>

#include <string.h>

#include <stdlib.h>

#include <windows.h>

**// Structure declaration**

struct teacher

{

int id, salary;

char name[100], sub[50];

} temp, tech[100];

**// Functions declartaion**

int menu(struct teacher tech[], struct teacher temp);

int input(struct teacher tech[], struct teacher temp);

int chk\_id(struct teacher tech[], int, int);

void display(struct teacher tech[], struct teacher temp);

void save(struct teacher tech[], int);

void query(struct teacher tech[], struct teacher temp);

void reset(struct teacher tech[], struct teacher temp);

void report(struct teacher tech[], struct teacher temp);

int verify(struct teacher tech[], struct teacher temp, char[]);

void verify\_mod();

int loading();

**// main**

void main()

{

system("color 09");

char pwd[50];

printf("\nWelcome to Guru Prabandhan C Program...\n");

printf("\nEnter 4-digit PIN to start or enter 1111 if your'e new ");

scanf("%s", &pwd);

verify(tech, temp, pwd);

}

**// menu for the user**

int menu(struct teacher tech[], struct teacher temp)

{

loading();

system("cls");

system("color 0a");

int prg\_no;

printf("\nEnter program number to execute\n");

printf("\nPress 1 to add teachers information");

printf("\nPress 2 to display teachers information");

printf("\nPress 3 to search for teachers by ID");

printf("\nPress 4 to generate summary");

printf("\nPress 5 to reset the database");

printf("\nPress 6 to change the password");

printf("\nPress 7 to exit\n");

scanf("%d", &prg\_no);

switch (prg\_no)

{

case 1:

system("color 03");

input(tech, temp);

break;

case 2:

system("color 1");

display(tech, temp);

break;

case 3:

query(tech, temp);

break;

case 4:

system("color 8");

report(tech, temp);

break;

case 5:

system("color 04");

reset(tech, temp);

break;

case 6:

system("color c");

printf("\nRequesting...");

verify\_mod();

break;

case 7:

system("cls");

system("color 06");

printf("Thank you for using Guru Prabandhan. \nIf you have any query or suggestion

feel free to reach me on contact@msmahato.com.np\n");

exit(0);

break;

default:

system("cls");

printf("Enter the valid program\n");

menu(tech, temp);

break;

}

}

**// to take the information from user when password is correct**

int input(struct teacher tech[], struct teacher temp)

{

int n, i, prg\_no, temp\_id;

printf("Enter the no of teachers \n");

scanf("%d", &n);

for (i = 0; i < n; i++)

{

printf("\nEnter the details of teacher:%d", i + 1);

printf("\nID :");

scanf("%d", &temp\_id);

tech[i].id = chk\_id(tech, temp\_id, n);

printf("\nname :");

fflush(stdin);

scanf("%[^\n]", &tech[i].name);

printf("\nSalary :");

scanf("%d", &tech[i].salary);

printf("\nSubject :");

fflush(stdin);

scanf("%[^\n]", &tech[i].sub);

}

save(tech, n);

menu(tech, temp);

}

**// to save the entered information from structure to disk**

void save(struct teacher tech[], int n)

{

FILE \*info;

info = fopen("info.txt", "r+");

int old\_no\_teacher, i;

fscanf(info, "%d", &old\_no\_teacher);

fseek(info, 0, SEEK\_SET);

fprintf(info, "%d\n", old\_no\_teacher + n);

fclose(info);

info = fopen("info.txt", "a");

for (i = 0; i < n; i++)

{

fprintf(info, "%d %s\n%d %s\n", tech[i].id, tech[i].name, tech[i].salary, tech[i].sub);

}

fclose(info);

printf("\n Information was saved");

}

**// to display teacher information in proper table format**

void display(struct teacher tech[], struct teacher temp)

{

FILE \*info;

info = fopen("info.txt", "r");

int id, salary, no\_of\_std, n = 0, i, j;

char name[100], sub[50];

system("cls");

fscanf(info, "%d", &no\_of\_std);

for (n = 0; n < no\_of\_std; n++)

{

fscanf(info, "%d %[^\n]\n%d %[^\n]\n", &tech[n].id, &tech[n].name,

&tech[n].salary,&tech[n].sub);

}

for (i = 0; i < no\_of\_std; i++)

{

for (j = i + 1; j < no\_of\_std; j++)

{

if (tech[i].id > tech[j].id)

{

temp = tech[i];

tech[i] = tech[j];

tech[j] = temp;

}

}

}

if (no\_of\_std >= 1)

{

printf("Displaying the sorted list of teachers \n");

printf("+------+------------------+-------------+-------------------+\n");

printf("| ID | NAME | SALARY | SUBJECT |\n");

printf("+------+------------------+-------------+-------------------+\n");

for (n = 0; n < no\_of\_std; n++) {

printf("| %-4d | %-16s | %-11d | %-17s |\n", tech[n].id, tech[n].name, tech[n].salary, tech[n].sub);

printf("+------+------------------+-------------+-------------------+\n");

}

}

else

{

printf("\n we couldn't find any data.");

}

fclose(info);

printf("Enter any key to return home page \n");

getch();

menu(tech, temp);

}

**// to serach for the teacher according to the teacher ID**

void query(struct teacher tech[], struct teacher temp)

{

int no\_of\_std, n, flag = 0, get\_id;

FILE \*info;

info = fopen("info.txt", "r");

fscanf(info, "%d", &no\_of\_std);

for (n = 0; n < no\_of\_std; n++)

{

fscanf(info, "%d %[^\n]\n%d %[^\n]\n", &tech[n].id, &tech[n].name,

&tech[n].salary,&tech[n].sub);

}

printf("Enter the valid ID to search in directory\n");

scanf("%d", &get\_id);

for (n = 0; n < no\_of\_std; n++)

{

if (tech[n].id == get\_id)

{

printf("%d\t %s\t %d\t %s", tech[n].id, tech[n].name, tech[n].salary, tech[n].sub);

printf("\n-----------------------------------------------------------\n");

flag++;

}

}

if (flag < 1)

{

printf("Couldn't find the teacher of ID: %d\n", get\_id);

}

printf("\nEnter any key to return home page\n");

getch();

menu(tech, temp);

}

**// to generate summary of all the informations saved in raw file**

void report(struct teacher tech[], struct teacher temp)

{

int n = 0, sum = 0, no\_of\_std, high\_salary = 0, least\_salary = 100000;

FILE \*info;

info = fopen("info.txt", "r");

fscanf(info, "%d", &no\_of\_std);

FILE \*summary;

summary = fopen("summary.txt", "w");

if (no\_of\_std >= 1)

{

for (n = 0; n < no\_of\_std; n++)

{

fscanf(info, "%d %[^\n]\n%d %[^\n]\n", &tech[n].id, &tech[n].name,

&tech[n].salary,

&tech[n].sub);

}

fclose(info);

fprintf(summary, "Guru Prabandhan C\nSummary of data saved on disk\n");

fprintf(summary, "Name of teacher & their subject\n");

for (n = 0; n < no\_of\_std; n++)

{

sum = sum + tech[n].salary;

}

for (n = 0; n < no\_of\_std; n++)

{

if (tech[n].salary > high\_salary)

{

high\_salary = tech[n].salary;

}

if (tech[n].salary < least\_salary)

{

least\_salary = tech[n].salary;

}

fprintf(summary, "\n%d. %s(%s) Salary: %d", tech[n].id, tech[n].name, tech[n].sub,

tech[n].salary);

fprintf(summary, "\n----------------------------------");

}

fprintf(summary, "\nThe total expenditure on salary is %d\n Where highest paid

salary is %d and least paid salary is %d", sum, high\_salary, least\_salary);

fclose(summary);

printf("The summary has been exported... Press any key to return home\n");

getch();

menu(tech, temp);

}

else

{

fprintf(summary, "%s", "No any data is saved in directory!");

fclose(summary);

printf("No any data is saved in directory!\n");

printf(" Press any key to return home \n");

getch();

menu(tech, temp);

}

}

**// to reset the database**

void reset(struct teacher tech[], struct teacher temp)

{

FILE \*a;

a = fopen("info.txt", "w");

fclose(a);

system("cls");

printf("The content of the file has been erased\nPress any key to return home");

getch();

menu(tech, temp);

}

**// to check password & handle the security feature like resetting database for too**

**many wrong attempt**

int verify(struct teacher tech[], struct teacher temp, char pass[])

{

int prg\_no;

char pwd[50];

static int count = 0;

char response;

FILE \*auth;

auth = fopen("auth.sec", "r");

fscanf(auth, "%s", &pwd);

if (strcmp(pwd, pass) == 0)

{

system("color 09");

menu(tech, temp);

}

else

{

system("color 04");

printf("\nAuthentication failed!\nEnter correct password: ");

fflush(stdin);

scanf("%s", pass);

count++;

if (count > 3 && count < 5)

{

printf("Too many attempt were made (2 left), Do you want to reset

databse?\n(CAUTION! All data created will be lost)\nPress any key to retry

password or press Y to reset\n ");

response = getch();

if (tolower(response) == 'y')

{

FILE \*b;

b = fopen("auth.sec", "w");

fprintf(b, "%s", "1111");

fclose(b);

reset(tech, temp);

}

else

{

verify(tech, temp, pass);

}

}

if (count > 5)

{

printf("We couldn't log you in.. Sorry!");

exit(0);

}

verify(tech, temp, pass);

}

}

**// function to change password**

void verify\_mod()

{

int flag = 0, i;

char old\_pwd[50], new\_pwd[50], temp[50];

FILE \*auth;

auth = fopen("auth.sec", "r");

fflush(stdin);

fscanf(auth, "%s", temp);

fclose(auth);

printf("Enter your correct old 4-digit password\n");

scanf("%s", old\_pwd);

if (strcmp(temp, old\_pwd) != 0)

{

printf("Incorrect password \n");

verify\_mod();

}

else

{

re:

flag = 0;

printf("Enter new 4-digit password password \n");

scanf("%s", new\_pwd);

for (i = 0; new\_pwd[i] != '\0'; i++)

{

flag++;

}

printf("%d", flag);

if (flag == 4)

{

FILE \*auth;

auth = fopen("auth.sec", "w");

fprintf(auth, "%s", new\_pwd);

printf("\nYour password was changed");

exit(0);

}

else

{

printf("\nnew password must be of 4 digit");

goto re;

}

}

}

**// to check for entered teacher's ID, Whether it is inputted previously or not (It check**

**in both structure & file)**

int chk\_id(struct teacher tech[], int temp\_id, int ar\_size)

{

struct teacher

{

int id, salary;

char name[100], sub[50];

} temp[100];

int no\_of\_std, n, flag = 0;

FILE \*info;

info = fopen("info.txt", "r");

fscanf(info, "%d", &no\_of\_std);

for (n = 0; n < ar\_size; n++)

{

if (tech[n].id == temp\_id)

{

flag++;

}

printf("%d %d checking array for id\n", n, flag);

}

for (n = 0; n < no\_of\_std; n++)

{

fscanf(info, "%d %[^\n]\n%d %[^\n]\n", &temp[n].id, &temp[n].name,

&temp[n].salary,

&temp[n].sub);

}

for (n = 0; n < no\_of\_std; n++)

{

if (temp[n].id == temp\_id)

{

flag++;

}

}

if (flag > 0)

{

printf("This ID has already been taken\n Enter new ID:");

scanf("%d", &temp\_id);

chk\_id(tech, temp\_id, ar\_size);

}

else

{

return temp\_id;

}

}

**// Display loading screen while switching function.**

int loading()

{

char loading[10] = "Loading";

int i;

for (i = 0; loading[i] != '\0'; i++)

{

Sleep(300);

printf("%c", loading[i]);

}

for (i = 0; i < 15; i++)

{

printf(".");

}

Sleep(900);

for (i = 0; i < 10; i++)

{

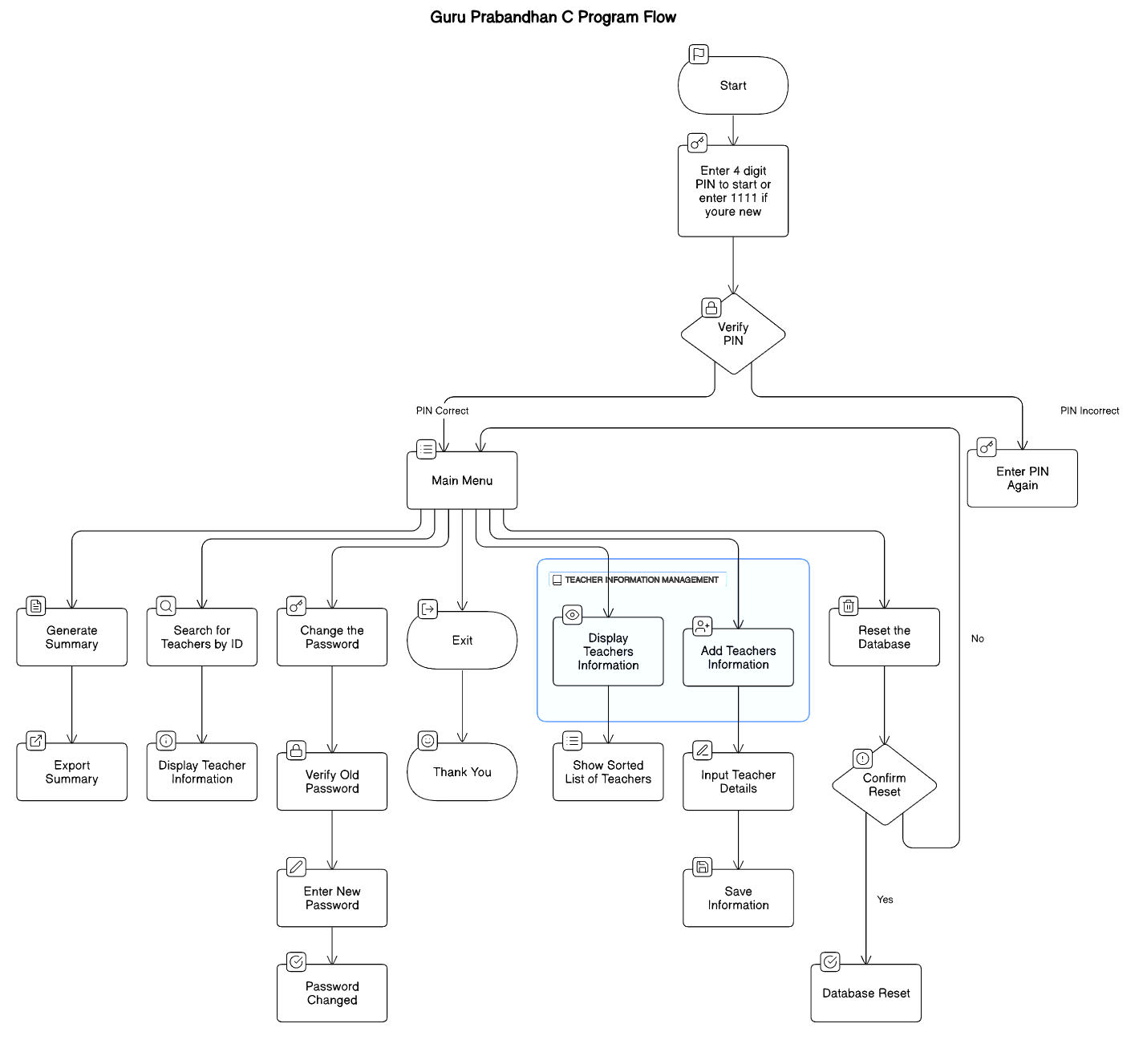
printf("..");

}

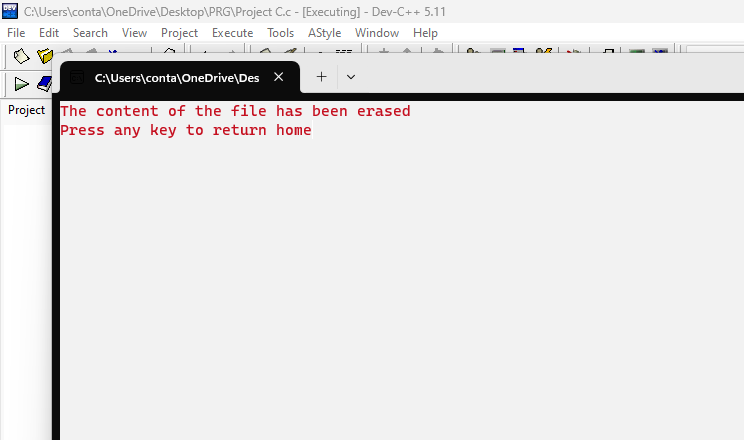
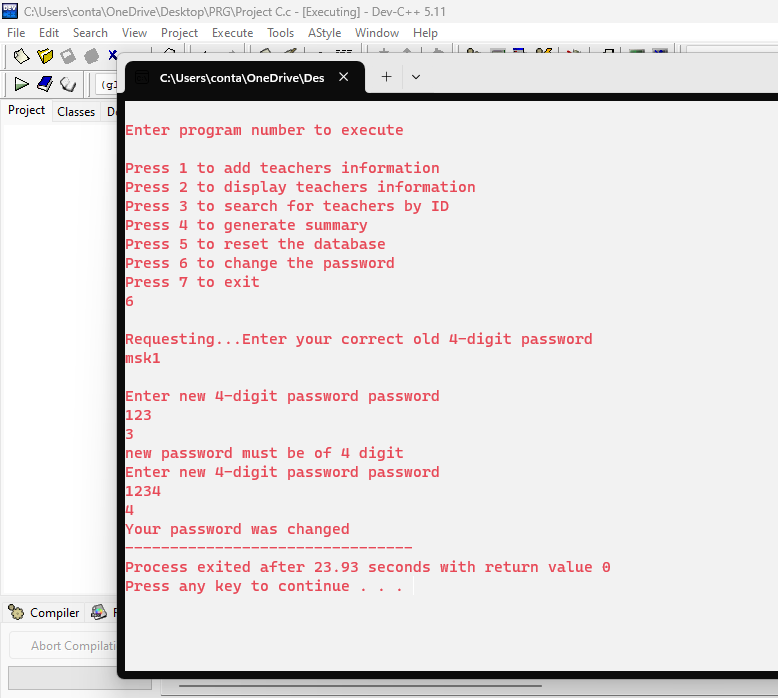
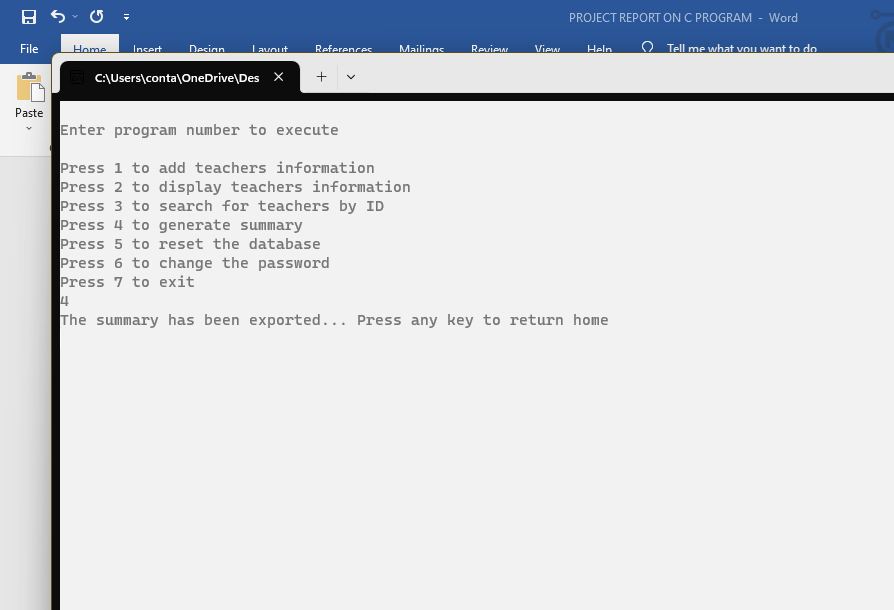
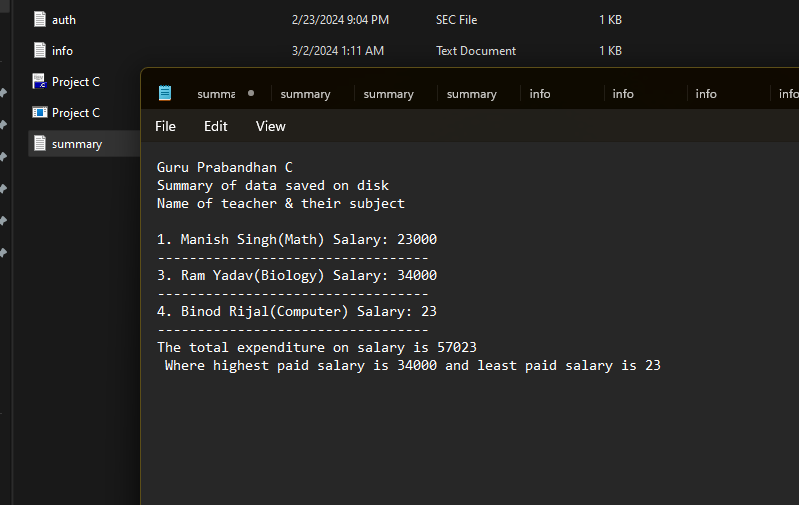
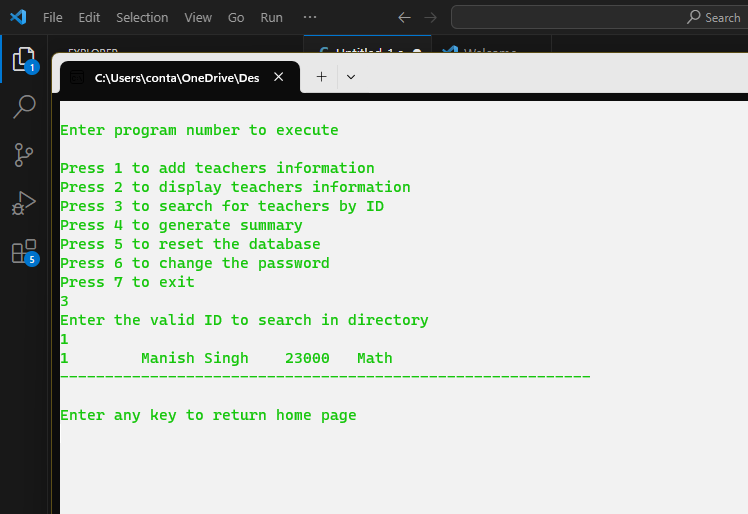
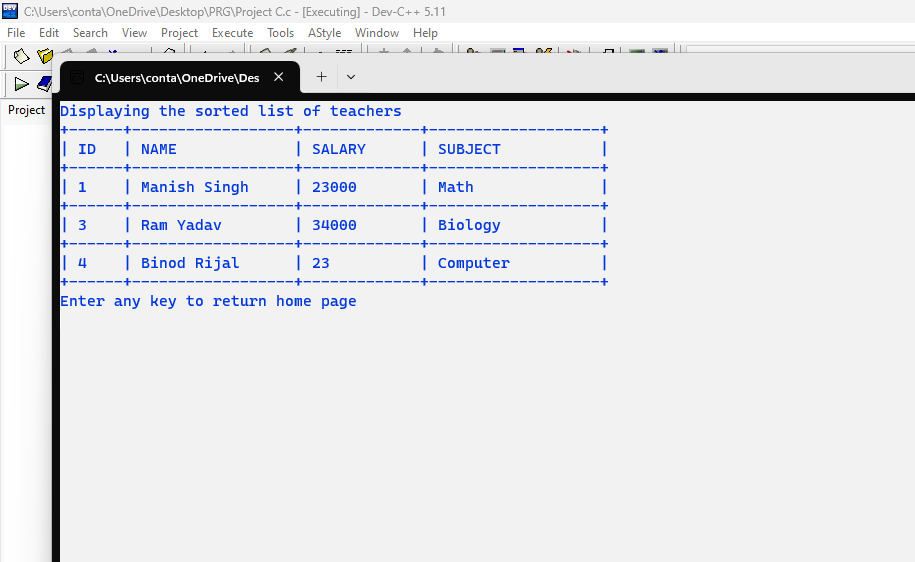
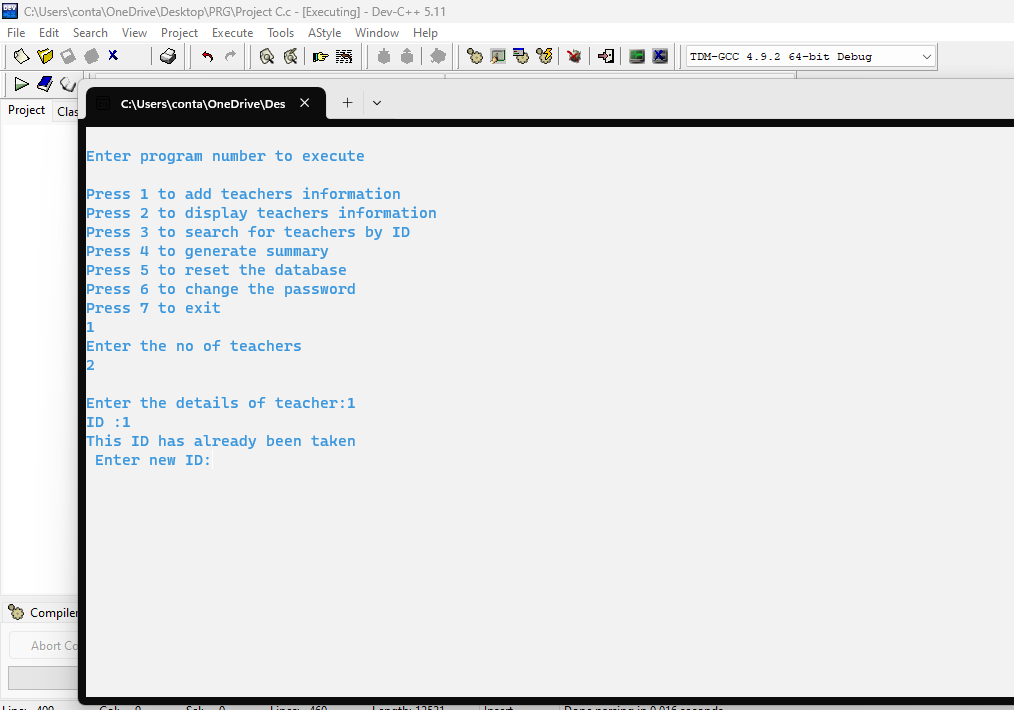
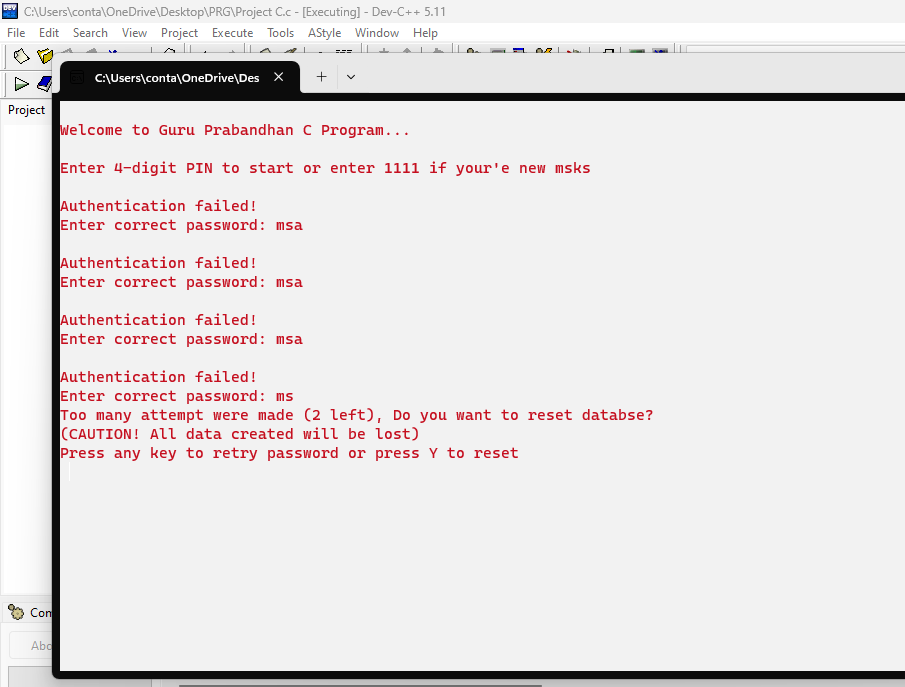
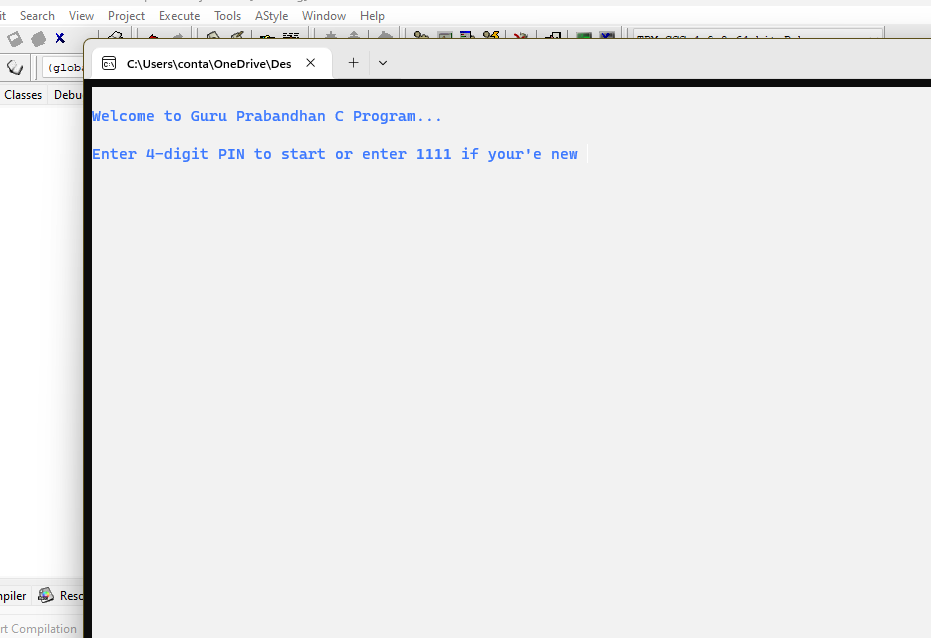
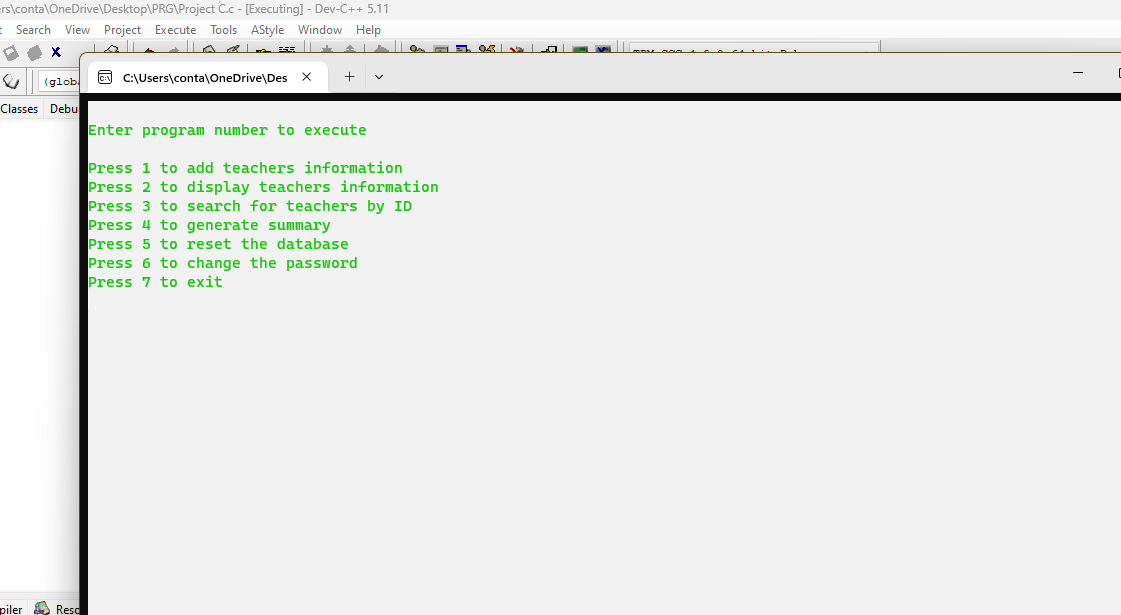
return 0;

}

**ANALYSIS**



**OUTPUT AND FINDINGS**



**CONCLUSION**

"Guru Prabandhan" is a comprehensive and user-friendly program designed to efficiently manage teacher information. Its intuitive interface allows users to seamlessly perform various tasks such as adding new teacher details, displaying existing information, searching for specific records, generating summaries of stored data, resetting the database, and modifying passwords.

One of the key strengths of "Guru Prabandhan" lies in its robust features for ensuring data integrity and security. Through mechanisms such as data validation, the program ensures that only valid information is accepted, preventing potential errors or inconsistencies in the database. Moreover, the implementation of password protection adds an extra layer of security, safeguarding sensitive information from unauthorized access.

The program is structured around the concept of file handling, enabling the persistent storage of teacher data. By utilizing file operations, it facilitates seamless data retrieval and manipulation, enhancing the overall efficiency of the system.

Furthermore, "Guru Prabandhan" incorporates effective error handling mechanisms to deal with unforeseen situations. For instance, it limits the number of failed authentication attempts to prevent unauthorized access and provides clear instructions for password modification, ensuring a smooth user experience.

Overall, "Guru Prabandhan" serves as an excellent example of applying fundamental principles of C programming. Its implementation of file operations, user input handling, conditional branching, and modular function usage not only makes it a practical solution for managing teacher information but also a valuable learning resource for individuals seeking to enhance their programming skills in C.

**REFERENCES**

1. *Fundamentals of programmin in C: Free course by GeeksforGeeks* (2019) *YouTube*. Available at: https://youtu.be/HMAwoMt3kek (Accessed: 02 March 2024).
2. GfG (2024) *C programming language*, *GeeksforGeeks*. Available at: https://www.geeksforgeeks.org/c-programming-language/ (Accessed: 02 March 2024).
3. *Introduction to C programming* (2018) *YouTube*. Available at: https://youtu.be/xjSpYFaBPYw (Accessed: 02 March 2024).
4. *Learn C programming* (2018) *Programiz*. Available at: https://www.programiz.com/c-programming (Accessed: 03 March 2024).
5. *Learn C programming* (2018) *Programiz*. Available at: https://www.programiz.com/c-programming (Accessed: 03 March 2024).
6. Manish Singh (2020b) *C Programming by Mold Skill*, *C & C++ for class XI & XII*. Available at: https://msmahato.com.np/4.COMPUTER/c%20programming/C++.html (Accessed: 03 March 2024).