

Operating Systems - Lab 02 In-Lab & Post-Lab Tasks

Ibrahim Johar Farooqi

23K-0074

BAI-4A

29 January 2025

In-Lab Questions:

Task 1:

Create the following classes in separate files (using .h and .cpp files) Student, Teacher, Course.

- A student has a list of courses that he is enrolled in. A teacher has a list of courses that he is teaching.
- A course has a list of students that are studying it, and a list of teachers that are teaching the course. Create some objects of all classes in main function and populate them with data.

Now compile all classes using makefile.

Course.cpp

A screenshot of a Linux desktop environment. On the left, a terminal window titled 'Course.cpp' is open, showing C++ code for a 'Course' class. On the right, a code editor window titled 'Operating Systems - Lab 02 Post-Lab Tasks' displays the same code. The desktop background shows a dark theme with various icons and a taskbar at the bottom.

```
GNU nano 7.2
#include <iostream>
#include "Course.h"
using namespace std;

Course::Course(const string& name) : name(name) {}

void Course::addStudent(Student* student)
{
    students.push_back(student);
}

void Course::addTeacher(Teacher* teacher)
{
    teachers.push_back(teacher);
}

const string& Course::getName() const { return name; }

const vector<Student*>& Course::getStudents() const { return students; }

const vector<Teacher*>& Course::getTeachers() const { return teachers; }
```

Student.cpp

A screenshot of a Linux desktop environment. On the left, a terminal window titled 'Student.cpp' is open, showing C++ code for a 'Student' class. On the right, a code editor window displays the same code. The desktop background shows a dark theme with various icons and a taskbar at the bottom.

```
GNU nano 7.2
#include <iostream>
#include "Student.h"
using namespace std;

Student::Student(const string& name) : name(name) {}

void Student::enrollCourse(Course* course)
{
    enrolledCourses.push_back(course);
}

const string& Student::getName() const { return name; }

const vector<Course*>& Student::getEnrolledCourses() const { return enrolledCourses; }
```

Teacher.cpp

```
GNU nano 7.2 Teacher.cpp
#include <iostream>
#include "Teacher.h"
using namespace std;

Teacher::Teacher(const string& name): name(name) {}

void Teacher::teachCourse(Course* course)
{
    teachingCourses.push_back(course);
}

const string& Teacher::getName() const
{
    return name;
}

const vector<Course*>& Teacher::getTeachingCourses() const
{
    return teachingCourses;
}
```

main.cpp

```
GNU nano 7.2 main.cpp
#include <iostream>
#include "Course.h"
#include "Student.h"
#include "Teacher.h"

int main()
{
    Student s1("ibrahim");
    Student s2("johar");
    Teacher t1("professor johnson");
    Course c1("mathematics");
    Course c2("physics");

    c1.addStudent(&s1); c1.addStudent(&s2);
    c1.addTeacher(&t1); c1.addTeacher(&t2);

    cout << "students enrolled in " << c1.getName() << ":"<\n";
    for(const auto& student : c1.getStudents())
    {
        cout << student->getName() << endl;
    }

    cout << "teachers teaching " << c1.getName() << ":"<\n";
    for(const auto& teacher : c1.getTeachers())
    {
        cout << teacher->getName() << endl;
    }
}
```

```
GNU nano 7.2 main.cpp
cout << "students enrolled in " << c1.getName() << ":"<\n";
for(const auto& student : c1.getStudents())
{
    cout << student->getName() << endl;
}

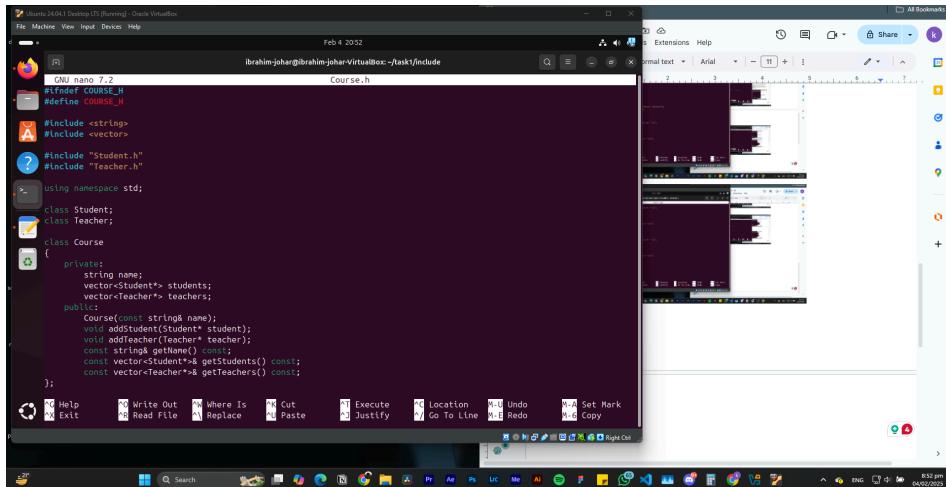
cout << "teachers teaching " << c1.getName() << ":"<\n";
for(const auto& teacher : c1.getTeachers())
{
    cout << teacher->getName() << endl;
}

cout << "students enrolled in " << c2.getName() << ":"<\n";
for(const auto& student : c2.getStudents())
{
    cout << student->getName() << endl;
}

cout << "teachers teaching " << c2.getName() << ":"<\n";
for(const auto& teacher : c2.getTeachers())
{
    cout << teacher->getName() << endl;
}

return 0;
}
```

Course.h



```
ibrahim-johar@ibrahim-johar-VirtualBox:~/task1/include
```

```
GNU nano 7.2 Course.h
#ifndef COURSE_H
#define COURSE_H

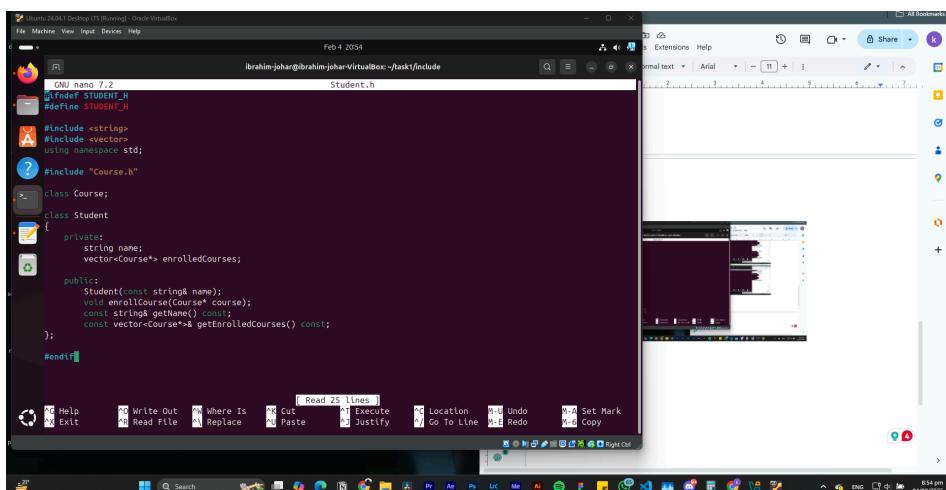
#include <string>
#include <vector>
#include "Student.h"
#include "Teacher.h"

using namespace std;

class Student;
class Teacher;

class Course {
private:
    string name;
    vector<Student*> students;
    vector<Teacher*> teachers;
public:
    Course(const string& name);
    void addStudent(Student* student);
    void addTeacher(Teacher* teacher);
    const string& getName() const;
    const vector<Student*>& getStudents() const;
    const vector<Teacher*>& getTeachers() const;
};
```

Student.h



```
ibrahim-johar@ibrahim-johar-VirtualBox:~/task1/include
```

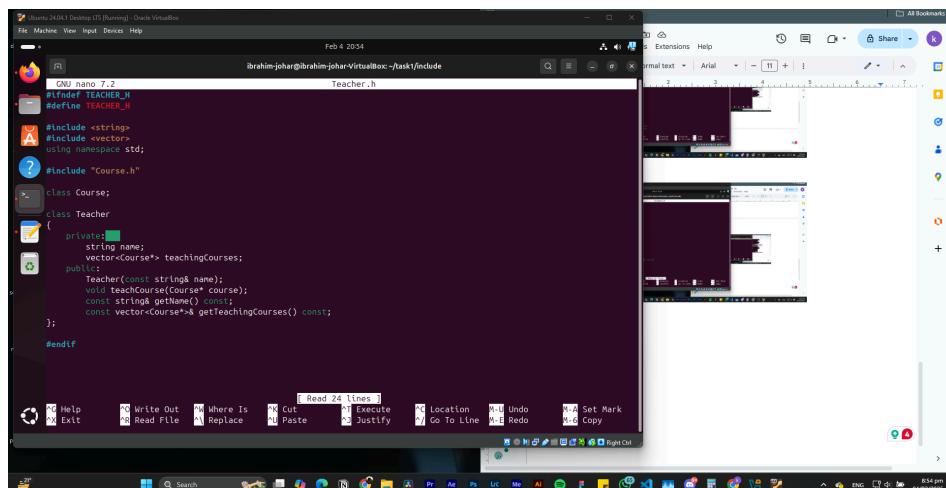
```
GNU nano 7.2 Student.h
#ifndef STUDENT_H
#define STUDENT_H

#include <string>
#include <vector>
using namespace std;
#include "Course.h"

class Student {
private:
    string name;
    vector<Course*> enrolledCourses;
public:
    Student(const string& name);
    void enrollCourse(Course* course);
    const string& getName() const;
    const vector<Course*>& getEnrolledCourses() const;
};

#endif
```

Teacher.h



```
ibrahim-johar@ibrahim-johar-VirtualBox:~/task1/include
```

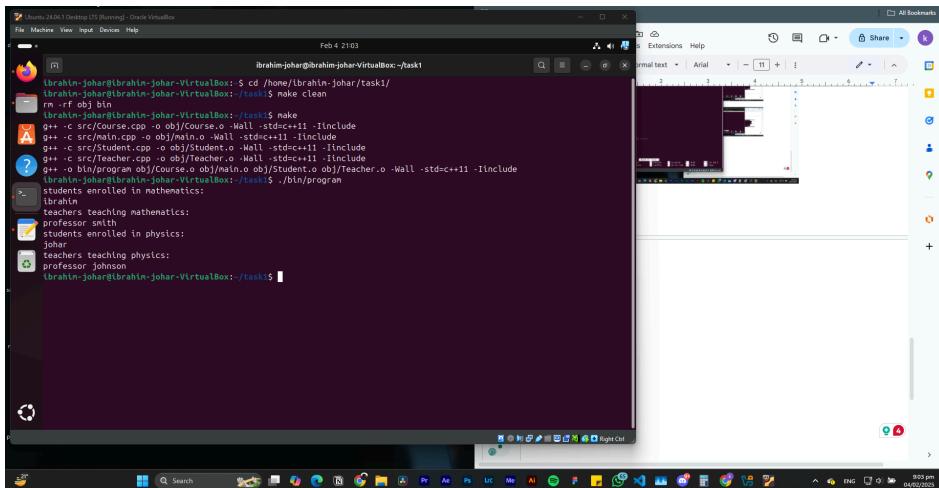
```
GNU nano 7.2 Teacher.h
#ifndef TEACHER_H
#define TEACHER_H

#include <string>
#include <vector>
using namespace std;
#include "Course.h"

class Teacher {
private:
    string name;
    vector<Course*> teachingCourses;
public:
    Teacher(const string& name);
    void teachCourse(Course* course);
    const string& getName() const;
    const vector<Course*>& getTeachingCourses() const;
};

#endif
```

Output using makefile:



The screenshot shows a dual-monitor setup. The left monitor displays a terminal window on an Ubuntu 24.04 LTS desktop. The terminal window title is "Ubuntu 24.04 LTS Desktop (Running - Oracle VirtualBox)". The command history shows:

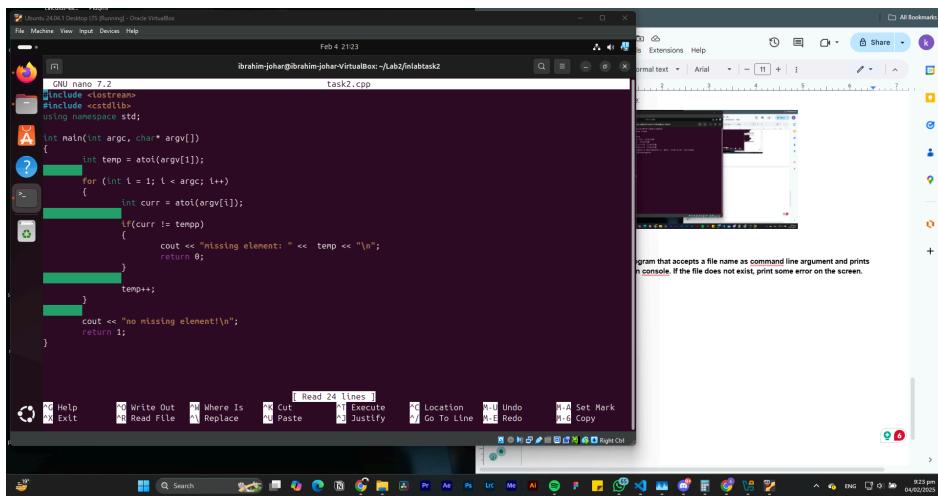
```
ibrahim-johar@ibrahim-johar-VirtualBox:~/task1$ cd /home/ibrahim-johar/task1/
ibrahim-johar@ibrahim-johar-VirtualBox:~/task1$ rm -rf obj bin
ibrahim-johar@ibrahim-johar-VirtualBox:~/task1$ make clean
g++ -c src/Course.cpp -o obj/Course.o -Wall -std=c++11 -Iinclude
g++ -c src/Student.cpp -o obj/Student.o -Wall -std=c++11 -Iinclude
g++ -c src/Teacher.cpp -o obj/Teacher.o -Wall -std=c++11 -Iinclude
g++ -o bin/program obj/Course.o obj/main.o obj/Student.o obj/Teacher.o -Wall -std=c++11 -Iinclude
ibrahim-johar@ibrahim-johar-VirtualBox:~/task1$ ./bin/program
students enrolled in mathematics:
teachers teaching mathematics:
professor smith
students enrolled in physics:
johar
teachers teaching physics:
professor johnson
ibrahim-johar@ibrahim-johar-VirtualBox:~/task1$
```

The right monitor displays a Microsoft Word document titled "Untitled - Word". The document contains the same text as the terminal output, listing students, teachers, and professors by name.

Task 2:

Write a C or C++ program that accepts a file name as command line argument and prints the file's contents on console. If the file does not exist, print some error on the screen.

task2.cpp

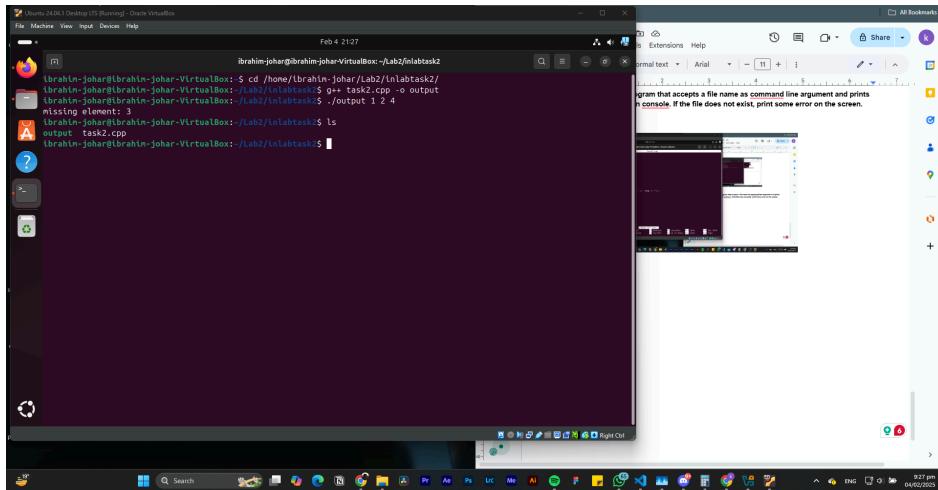


```
ibrahim-johar@ibrahim-johar-VirtualBox ~/Lab2/unlabtask2
```

```
#include <iostream>
#include <cstdlib>
using namespace std;

int main(int argc, char* argv[])
{
    int temp = atoi(argv[1]);
    for (int i = 1; i < argc; i++)
    {
        int curr = atoi(argv[i]);
        if(curr != temp)
        {
            cout << "missing element: " << temp << "\n";
            return 0;
        }
        temp++;
    }
    cout << "no missing element!\n";
    return 1;
}
```

task2 console output:



```
ibrahim-johar@ibrahim-johar-VirtualBox: ~ cd /home/ibrahim-johar/Lab2/unlabtask2/
ibrahim-johar@ibrahim-johar-VirtualBox: /Lab2/unlabtask2$ g++ task2.cpp -o output
ibrahim-johar@ibrahim-johar-VirtualBox: /Lab2/unlabtask2$ ./output 1 2 4
missing element: 3
ibrahim-johar@ibrahim-johar-VirtualBox: /Lab2/unlabtask2$ ls
task2.cpp
ibrahim-johar@ibrahim-johar-VirtualBox: /Lab2/unlabtask2$
```

Task 3:

Write a C or C++ program that accepts a list of integers as command line arguments sorts the integers and print the sorted integers on the screen.

task3.cpp

```
GNU nano 7.2          task3.cpp
ibrahim-johar@ibrahim-johar-VirtualBox:~/Lab2/inlabtask3
//23k-0074
//hol-4a
//os lab - intab - task 3

#include <iostream>
#include <vector>
#include <cstdlib>
using namespace std;

int main(int argc, char* argv[])
{
    if(argc < 2)
    {
        cout << "usage: " << argv[0] << " <int> <int> ... <int>\n";
        return 1; //exit w/ error
    }
    vector<int> numbers;
    for(int i = 1; i < argc; i++)
    {
        numbers.push_back(atol(argv[i])); //string to int
    }
    int temp;
    for(int i = 0; i < numbers.size(); i++)
    {
        for(int j = 0; j < numbers.size() - 1 - i; j++)
        {
            if(numbers[j] > numbers[j+1])
            {
                temp = numbers[j];
                numbers[j] = numbers[j+1];
                numbers[j+1] = temp;
            }
        }
    }
    cout << "sorted numbers: ";
    for(int num: numbers)
    {
        cout << num << " ";
    }
    cout << "\n";
    return 0;
}

ibrahim-johar@ibrahim-johar-VirtualBox:~/Lab2/inlabtask3
```

task3 output:

```
ibrahim-johar@ibrahim-johar-VirtualBox: $ cd /home/ibrahim-johar/Lab2/inlabtask3/
ibrahim-johar@ibrahim-johar-VirtualBox:~/Lab2/inlabtask3$ nano task3.cpp
ibrahim-johar@ibrahim-johar-VirtualBox:~/Lab2/inlabtask3$ nano task3.cpp
ibrahim-johar@ibrahim-johar-VirtualBox:~/Lab2/inlabtask3$ g++ task3.cpp -o output
ibrahim-johar@ibrahim-johar-VirtualBox:~/Lab2/inlabtask3$ ./output 12 5 9 1 7
sorted numbers: 1 5 7 9 12
ibrahim-johar@ibrahim-johar-VirtualBox:~/Lab2/inlabtask3$ nano task3.cpp
ibrahim-johar@ibrahim-johar-VirtualBox:~/Lab2/inlabtask3$
```

Post-Lab Questions:

Task 1: Write a C/C++ program that takes some integers as command line parameters, store them in an array and prints the sum and average of that array. Also note that you have to run the program for all possible error checks.

task1.cpp

```
//ibrahim johar farooqi
//23k-0074
//bai-4a
//os lab - post lab - task 1

#include <iostream>
#include <vector>
#include <cstdlib>
using namespace std;

int main(int argc, char* argv[])
{
    if(argc < 2)
    {
        cout << "error: no numbers provided\n";
        cout << "usage: " << argv[0] << "<int1> <int2> ... <intN>\n";
        return 1;
    }

    vector<int> numbers;
    int sum = 0;

    for(int i = 1; i < argc; i++)
    {
        bool isnum = true;
        for(int j = 0; argv[i][j] != '\0'; j++)
        {
            if(!(argv[i][j] >= '0' && argv[i][j] <= '9') && !(argv[i][j]
== '-' && j == 0))
            {
                isnum = false;
                break;
            }
        }
    }
}
```

```

if (!isnum)
{
    cout << "error: '" << argv[i] << "' is not valid integer\n";
    return 1;
}

int num = atoi(argv[i]); //string to int
numbers.push_back(num);
sum += num;
}

double average = static_cast<double>(sum) / numbers.size();

cout << "numbers: ";
for (int num : numbers) { cout << num << " "; }
cout << "\nsum: " << sum;
cout << "\nAverage: " << average << "\n";

return 0;
}

```

task1 output:

```

Ubuntu 20.04 LTS (Bionic) - Oracle VM VirtualBox
File Machine View Input Devices Help
Feb 5 01:45
ibrahim-johar@ibrahim-johar-VirtualBox:~$ cd /home/ibrahim-johar/Lab2/postlabtask1
ibrahim-johar@ibrahim-johar-VirtualBox:~/Lab2/postlabtask1$ nano task1.cpp
ibrahim-johar@ibrahim-johar-VirtualBox:~/Lab2/postlabtask1$ g++ task1.cpp -o output
ibrahim-johar@ibrahim-johar-VirtualBox:~/Lab2/postlabtask1$ ./output 10 20 30 40 50
numbers: 10 20 30 40 50
sum: 150
Average: 30
ibrahim-johar@ibrahim-johar-VirtualBox:~/Lab2/postlabtask1$ ./output 0 12 32 21 9
sum: 74
Average: 14.8
ibrahim-johar@ibrahim-johar-VirtualBox:~/Lab2/postlabtask1$ ./output -2 3 1 9
numbers: -2 3 1 9
sum: 11
Average: 2.75
ibrahim-johar@ibrahim-johar-VirtualBox:~/Lab2/postlabtask1$ ./output a b 21 13
error: 'a' is not valid integer
ibrahim-johar@ibrahim-johar-VirtualBox:~/Lab2/postlabtask1$ 

```

Task 2:

Write a C/C++ program that takes some integers in the form of series as command line parameters; store them in array than compute the missing element from that series and output that missing element to file.

```
//ibrahim johar farooqi
//23k-0074
//bai-4a
//os lab - post lab - task 2

#include <iostream>
#include <vector>
#include <fstream>
#include <algorithm>
#include <cstdlib>
using namespace std;

int missingnum(vector<int> &numbers)
{
    sort(numbers.begin(), numbers.end()); //sort

    for (int i = 0; i < numbers.size() - 1; i++)
    {
        if (numbers[i] + 1 != numbers[i + 1])
        {
            return numbers[i] + 1; //missing num found
        }
    }
    return -1; //no missing num found
}

int main(int argc, char *argv[])
{
    if (argc < 3)
    {
        cout << "error: please provide a series that has atleast 2
numbers.\n";
        cout << "usage: " << argv[0] << " <int1> <int2> ... <intN>\n";
        return 1;
    }
}
```

```
}

vector<int> numbers;

for (int i = 1; i < argc; i++)
{
    numbers.push_back(atoi(argv[i])); //string to integer
}

int missing = missingnum(numbers);

//write missing num to file
ofstream outputfile("output.txt");
if (!outputfile)
{
    cout << "couldnt open file for writing.\n";
    return 1;
}

if (missing == -1)
{
    outputfile << "no missing number found in sequence.\n";
    cout << "no missing number found.\n";
}
else
{
    outputfile << "missing number: " << missing << "\n";
    cout << "missing number found & saved in output.txt: " << missing
<< "\n";
}

outputfile.close();
return 0;
}
```

task2 output:

The screenshot shows a Linux desktop environment with a terminal window and a file viewer window. The terminal window displays the command-line session for task2, including the compilation of task2.cpp and its execution with different input files (output1.txt and output2.txt). The file viewer window shows the contents of output1.txt and output2.txt, which both contain the message "no missing number found in sequence.\nmissing number found;".

```
ibrahim-johar@ibrahim-johar-VirtualBox:~/Lab2/postlabtask2$ cd /home/ibrahim-johar/Lab2/postlabtask2/
ibrahim-johar@ibrahim-johar-VirtualBox:~/Lab2/postlabtask2$ nano task2.cpp
ibrahim-johar@ibrahim-johar-VirtualBox:~/Lab2/postlabtask2$ g++ task2.cpp -o output
ibrahim-johar@ibrahim-johar-VirtualBox:~/Lab2/postlabtask2$ ./output 1 2 3 5 6
missing number found & saved in output.txt: 4
ibrahim-johar@ibrahim-johar-VirtualBox:~/Lab2/postlabtask2$ cat output.txt
missing number: 4
ibrahim-johar@ibrahim-johar-VirtualBox:~/Lab2/postlabtask2$ ./output 56 58 59 60
missing number found & saved in output.txt: 57
ibrahim-johar@ibrahim-johar-VirtualBox:~/Lab2/postlabtask2$ cat output.txt
missing number: 57
ibrahim-johar@ibrahim-johar-VirtualBox:~/Lab2/postlabtask2$
```

Task 3:

Write a C/C++ program that reads file in which there are integers related to series and store them in array than compute the missing element from that series and output that missing element to file.

```
//ibrahim johar farooqi
//23k-0074
//bai-4a
//os lab - post lab - task 3

#include <iostream>
#include <vector>
#include <fstream>
#include <algorithm>
using namespace std;

int missingnum(vector<int> &numbers)
{
    sort(numbers.begin(), numbers.end()); //sort

    for (int i = 0; i < numbers.size() - 1; i++)
    {
        if (numbers[i] + 1 != numbers[i + 1])
        {
            return numbers[i] + 1; //missing num found
        }
    }
}
```

```
    }

    return -1; //no missing num found
}

int main()
{
    string filename;
    cout << "enter filename: ";
    cin >> filename;

    ifstream file1(filename);
    if (!file1)
    {
        cout << "file not found or cant be opened.\n";
        return 1;
    }

    vector<int> numbers;
    int num;

    //read file
    while (file1 >> num)
    {
        numbers.push_back(num);
    }
    file1.close();

    if (numbers.size() < 2)
    {
        cout << "insufficient numbers in the file.\n";
        return 1;
    }

    int missing = missingnum(numbers);

    ofstream file2("output.txt");
    if (!file2)
    {
        cout << "couldnt open file for writing.\n";
        return 1;
    }
```

```

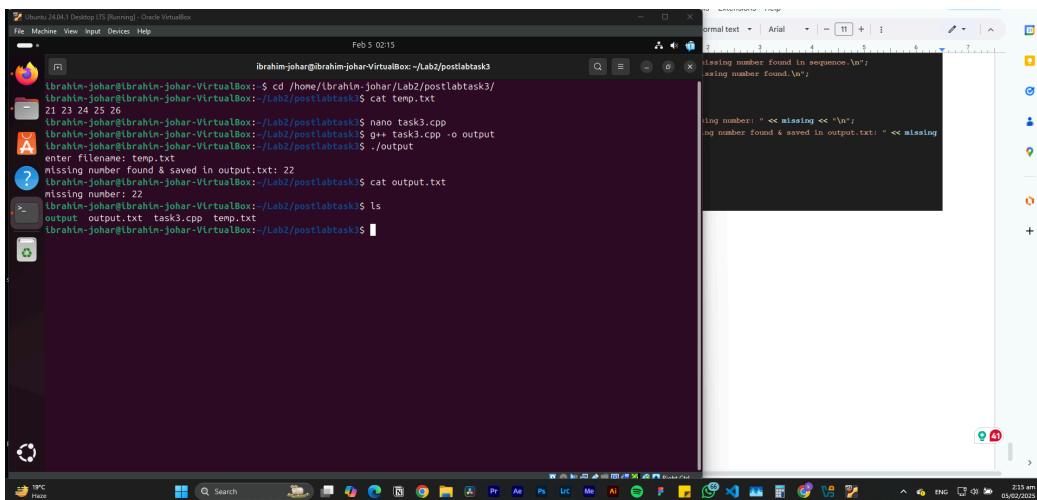
}

if (missing == -1)
{
    file2 << "no missing number found in sequence.\n";
    cout << "no missing number found.\n";
}
else
{
    file2 << "missing number: " << missing << "\n";
    cout << "missing number found & saved in output.txt: " << missing
<< "\n";
}

file2.close();
return 0;
}

```

task3 output:



The screenshot shows a terminal window in Oracle VM VirtualBox. The terminal session starts with the user navigating to the directory /Lab2/postlabtask3. They then compile the C++ program task3.cpp using g++, link it with the command 'g++ task3.cpp -o output', and run it with ./output. The program prompts for a filename, which is specified as temp.txt. The output of the program is displayed in the terminal, indicating that no missing number was found in the sequence and that the missing number (22) has been saved to output.txt. The terminal also shows the contents of output.txt, which is 22. Finally, the user runs ls to list the files in the directory, showing task3.cpp, temp.txt, and output.txt.

```

Ubuntu 24.04.1 Desktop (TS (Bluering)) - Oracle VM VirtualBox
File Machine View Input Devices Help
Feb 5 02:15
ibrahim-johar@ibrahim-johar-VirtualBox:/Lab2/postlabtask3
ibrahim-johar@ibrahim-johar-VirtualBox:~$ cd /home/ibrahim-johar/Lab2/postlabtask3
ibrahim-johar@ibrahim-johar-VirtualBox:/Lab2/postlabtask3$ cat temp.txt
21 23 24 25 26
ibrahim-johar@ibrahim-johar-VirtualBox:/Lab2/postlabtask3$ nano task3.cpp
ibrahim-johar@ibrahim-johar-VirtualBox:/Lab2/postlabtask3$ g++ task3.cpp -o output
ibrahim-johar@ibrahim-johar-VirtualBox:/Lab2/postlabtask3$ ./output
enter filename: temp.txt
missing number found & saved in output.txt: 22
ibrahim-johar@ibrahim-johar-VirtualBox:/Lab2/postlabtask3$ cat output.txt
missing number: 22
ibrahim-johar@ibrahim-johar-VirtualBox:/Lab2/postlabtask3$ ls
output.txt task3.cpp temp.txt
ibrahim-johar@ibrahim-johar-VirtualBox:/Lab2/postlabtask3$ 

```

Task 4:

Create the following classes in separate files (using .h and .cpp files) LetterCount, WordCount, LineCount.

- a. LetterCount counts number of letters in a text file.
- b. WordCount counts number of words in a text file.
- c. LineCount counts number of lines in a text file.

Create some objects of all classes in main function and populate them with data. Now compile all classes using makefile.

task4 .h files:

```
#ifndef LETTER_COUNT_H
#define LETTER_COUNT_H
#include <string>
using namespace std;

class LetterCount
{
public:
    int count_letters(const string &filename);
};

#endif
```

```
#ifndef WORD_COUNT_H
#define WORD_COUNT_H
#include <string>
using namespace std;

class WordCount
{
public:
    int count_words(const string &filename);
};

#endif
```

```
#ifndef LINE_COUNT_H
#define LINE_COUNT_H
#include <string>
```

```
using namespace std;

class LineCount
{
public:
    int count_lines(const string &filename);
};

#endif
```

task4 .cpp files:

```
#include "LetterCount.h"
#include <iostream>
#include <fstream>
#include <cctype>
using namespace std;

int LetterCount::count_letters(const string &filename)
{
    ifstream file1(filename);
    if (!file1)
    {
        cout << "cant open file.\n";
        return -1;
    }

    char ch;
    int letterCount = 0;
    while (file1.get(ch))
    {
        if (isalpha(ch)) letterCount++; //only count letters
    }

    file1.close();

    return letterCount;
}
```

```
#include "WordCount.h"
#include <iostream>
#include <fstream>
using namespace std;

int WordCount::count_words(const string &filename)
{
    ifstream file1(filename);

    if (!file1)
    {
        cout << "cant open file.\n";
        return -1;
    }

    string word;
    int wordCount = 0;
    while (file1 >> word)
    {
        wordCount++; //count words only (identified by whitespaces)
    }

    file1.close();

    return wordCount;
}
```

```
#include "LineCount.h"
#include <iostream>
#include <fstream>
using namespace std;

int LineCount::count_lines(const string &filename)
{
    ifstream file1(filename);

    if (!file1)
    {
        cout << "cant open file.\n";
```

```
        return -1;
    }

    string line;
    int lineCount = 0;

    while (getline(file1, line))
    {
        lineCount++; //counting every line
    }

    file1.close();

    return lineCount;
}
```

```
//ibrahim johar farooqi
//23k-0074
//bai-4a
//os lab - post lab - task 4

#include <iostream>
#include "LetterCount.h"
#include "WordCount.h"
#include "LineCount.h"
using namespace std;

int main()
{
    string filename = "task4output.txt";

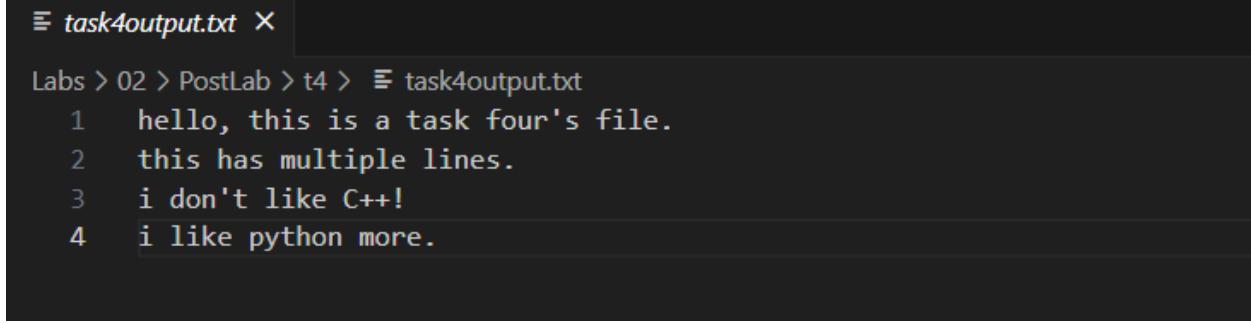
    LetterCount letter_counter;
    WordCount word_counter;
    LineCount line_counter;

    int letters = letter_counter.count_letters(filename);
    int words = word_counter.count_words(filename);
    int lines = line_counter.count_lines(filename);
```

```
    if (letters != -1) cout << "letters: " << letters << "\n";
    if (words != -1) cout << "words: " << words << "\n";
    if (lines != -1) cout << "lines: " << lines << "\n";

    return 0;
}
```

task4output.txt



```
Labs > 02 > PostLab > t4 > task4output.txt
1 hello, this is a task four's file.
2 this has multiple lines.
3 i don't like C++!
4 i like python more.
```

makefile:

```
#compiler
CC = g++
CFLAGS = -Wall -std=c++11 -Iinclude

#directories
SRC_DIR = src
OBJ_DIR = obj
BIN_DIR = bin

#source & object files
SRCS = $(SRC_DIR)/main.cpp $(SRC_DIR)/LetterCount.cpp
$(SRC_DIR)/WordCount.cpp $(SRC_DIR)/LineCount.cpp
OBJS = $(OBJ_DIR)/main.o $(OBJ_DIR)/LetterCount.o $(OBJ_DIR)/WordCount.o
$(OBJ_DIR)/LineCount.o

#output executable
TARGET = $(BIN_DIR)/counter

#rule to create the executable
all: $(TARGET)
```

```

$(TARGET) : $(OBJS) | $(BIN_DIR)
    $(CC) -o $(TARGET) $(OBJS) $(CFLAGS)

#rule for compiling .cpp files into .o files
$(OBJ_DIR)/%.o: $(SRC_DIR)/%.cpp | $(OBJ_DIR)
    $(CC) -c $< -o $@ $(CFLAGS)

#ensure that required directories exist
$(OBJ_DIR):
    mkdir -p $(OBJ_DIR)

$(BIN_DIR):
    mkdir -p $(BIN_DIR)

#the clean rule
clean:
    rm -rf $(OBJ_DIR) $(BIN_DIR)

```

task4 output:

The screenshot shows a Linux desktop environment with a terminal window and a browser window.

Terminal Window Content:

```

Ubuntu 24.04.1 Desktop LTS [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Feb 5 03:03
ibrahim-johar@ibrahim-johar-VirtualBox:~/Lab2/postlabtask4
ibrahim-johar@ibrahim-johar-VirtualBox:~/Lab2/postlabtask4$ make
mkdir -p obj
g++ -c src/main.cpp -o obj/main.o -Wall -std=c++11 -Iinclude
g++ -c src/LetterCount.cpp -o obj/LetterCount.o -Wall -std=c++11 -Iinclude
g++ -c src/WordCount.cpp -o obj/WordCount.o -Wall -std=c++11 -Iinclude
g++ -c src/LineCount.cpp -o obj/LineCount.o -Wall -std=c++11 -Iinclude
mkdir -p bin
g++ -o bin/counter obj/main.o obj/LetterCount.o obj/WordCount.o obj/LineCount.o -Wall -std=c++11 -Iinclude
ibrahim-johar@ibrahim-johar-VirtualBox:~/Lab2/postlabtask4$ ./bin/counter
letters: 70
words: 19
lines: 4
ibrahim-johar@ibrahim-johar-VirtualBox:~/Lab2/postlabtask4$ cat task4output.txt
hello, this is a task four's file.
this has multiple lines.
i don't like C++!
i like python more.
ibrahim-johar@ibrahim-johar-VirtualBox:~/Lab2/postlabtask4$ []

```

Browser Window: A dark-themed browser window is visible on the right side of the screen, showing a blank page.