

TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

THAPATHALI CAMPUS

A Final Report

On

IOE Entrance Preparation and Data Management Kit

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

COURSE OF CT 401 COMPUTER PROGRAMMING

BACHELOR OF ELECTRONICS, COMMUNICATION AND INFORMATION ENGINEERING

Submitted By:

Anish Timsina [THA077BEI007]

Bishal Khadka [THA077BEI015]

Nixon Raj Dhakal [THA077BEI028]

Sabin Acharya [THA077BEI035]

Submitted To:

Department of Electronics and Computer Engineering

Thapathali Campus

Kathmandu, Nepal

October 2021



TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

THAPATHALI CAMPUS

A Final Report

On

IOE Entrance Preparation and Data Management Kit

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

COURSE OF CT 401 COMPUTER PROGRAMMING

BACHELOR OF ELECTRONICS, COMMUNICATION AND INFORMATION ENGINEERING

Submitted By:

Anish Timsina [THA077BEI007]

Bishal Khadka [THA077BEI015]

Nixon Raj Dhakal [THA077BEI028]

Sabin Acharya [THA077BEI035]

Submitted To:

Department of Electronics and Computer Engineering

Thapathali Campus

Kathmandu, Nepal

October, 2021

# DECLARATION

We hereby declare that the project work report entitled “IOE Entrance Preparation and Data Management Kit” submitted for the partial fulfillment of the requirements for the course of CT 401 Computer Programming is our original work and the Project Work Report has not formed the basis for the award of any degree, diploma, or other similar titles.

Signature:

Anish Timsina [THA077BEI007]

Signature:

Bishal Khadka [THA077BEI015]

Signature:

Nixon Raj Dhakal [THA077BEI028]

Signature:

Sabin Acharya [THA077BEI035]

# ACKNOWLEDGEMENT

We would like to express our sincere gratitude towards the Institute of Engineering, Tribhuvan University for the inclusion of the project in the course of computer programming in Bachelors in Electronics, Communication and Information Engineering. We are also thankful to our subject teacher Er. Saroj Shakya sir and the Department of Electronics and Computer Engineering, Thapathali Campus for providing us support which is needed for this project.

Special support is taken from ‘learning C by examples’, ‘secret of C’, and the website of engineering dote. At last but not least we will like to thank our parents (The living god) who toil hard and their prayer invisibly saves us from any problem.

Anish Timsina [THA077BEI007]

Bishal Khadka [THA077BEI015]

Nixon Raj Dhakal [THA077BEI028]

Sabin Acharya [THA077BEI035]

# ABSTRACT

Every year, thousands of students from all over Nepal contemplate to Crack IOE entrance examination to get qualified to study Engineering in Nepal’s one of the oldest and largest universities i.e., Tribhuvan University (T.U.), Institute Of Engineering (IOE). As the number of IOE aspirants is increasing annually, the level of competence has also increased drastically in the last decades. So, students need to be highly competitive and need a lot of practice to crack the entrance. Few institutions are providing classes regarding entrance preparation where a lot of students enroll each year. As the number of students in a single institution is very high, there is a lack of proper management in the administration and examination. They still make use of file systems to take a record of the enrolled students, which is a quite traditional and inefficient method for this purpose which takes a lot of time to find the data of the students in case if required. They still employ the system of paper-based exams for the MCQ test which is very time-consuming to check and publish the results.

This project aspires to fabricate a software that will be a complete IOE entrance preparation kit and a data management software that will assist any institutions providing the IOE entrance preparation classes in student data and examination management and also to help the students to boost up their preparation speed and quality providing them access to a lot of practice exams with instant results. In addition to this, this software would be able to assist all the other institutions providing the MCQs based entrance preparation classes with some minor modifications in the software. This software can also be linked with the internet to provide online exam question papers to the students so that they can practice and test themselves instantly at home which is highly effective in the current situation of Lockdown.

# Table of Contents

[DECLARATION iii](#_Toc85955014)

[ACKNOWLEDGEMENT iv](#_Toc85955015)

[ABSTRACT v](#_Toc85955016)

[Table of Contents vi](#_Toc85955017)

[List of Figures vii](#_Toc85955018)

[List of Abbreviations viii](#_Toc85955019)

[1. INTRODUCTION 1](#_Toc85955020)

[1.1 Background Introduction 1](#_Toc85955021)

[1.2 Motivation 1](#_Toc85955022)

[1.3 Problem Definition 2](#_Toc85955023)

[1.4 Objectives 2](#_Toc85955024)

[1.5 Applications 3](#_Toc85955025)

[1.6 Project Features 3](#_Toc85955026)

[1.7 Feasibility Analysis 4](#_Toc85955027)

[1.7.1 Economic Feasibility 4](#_Toc85955028)

[1.7.2 Technical Feasibility 4](#_Toc85955029)

[1.7.3 Operational Feasibility 5](#_Toc85955030)

[1.7.4 Practical Uses 5](#_Toc85955031)

[1.8 System Requirement 6](#_Toc85955032)

[1.8.1 Software Requirement 6](#_Toc85955033)

[1.8.2 Hardware Requirement 6](#_Toc85955034)

[2. LITERATURE REVIEW 7](#_Toc85955035)

[2.1 “Enroll Yourself” Block 7](#_Toc85955036)

[2.2 “Take Practice Exam” Block 7](#_Toc85955037)

[2.3 “Take Full Mock Test” Block 7](#_Toc85955038)

[2.4 “View Overall Score” Block 7](#_Toc85955039)

[2.5 “View Leaderboard” Block 7](#_Toc85955040)

[2.6 “Amend Question Set” Block 8](#_Toc85955041)

[2.7 “View All Enrolled Students” Block 8](#_Toc85955042)

[2.8 “Exit” block 8](#_Toc85955043)

[3. DESIGN AND METHODOLOGY 9](#_Toc85955044)

[3.1 Introductory Block 9](#_Toc85955045)

[3.2 Option Selector Block 9](#_Toc85955046)

[3.3 Enroll Yourself Block 10](#_Toc85955047)

[3.4 Take Practice Exam Block 11](#_Toc85955048)

[3.5 Take Full Mock Test Block 11](#_Toc85955049)

[3.6 View Overall Score Block 11](#_Toc85955050)

[3.7 View Leaderboard Block 11](#_Toc85955051)

[3.8 Amend Question Set Block 11](#_Toc85955052)

[3.9 View All Enrolled Students Block 12](#_Toc85955053)

[3.10 Exit Block 12](#_Toc85955054)

[4 IMPLEMENTATION AND RESULT 13](#_Toc85955055)

[4.1 Introduction: 13](#_Toc85955056)

[4.1.1 Some screenshots of results: 13](#_Toc85955057)

[4.2 Result Analysis 14](#_Toc85955058)

[5 CONCLUSION & FURTHER WORK 15](#_Toc85955059)

[5.1 Conclusion 15](#_Toc85955060)

[5.2 Further Works 15](#_Toc85955061)

[6 REFERENCES 15](#_Toc85955062)

[7 APPENDICES 15](#_Toc85955063)

# List of Figures

[Figure 3- 1: Introductory Block 9](#_Toc85955007)

[Figure 3- 2: Option Selector Block 10](#_Toc85955008)

[Figure 3- 3: Enroll Yourself Block 10](#_Toc85955009)

# List of Abbreviations

IOE = Institute Of Engineering

# INTRODUCTION

C was initially used for system development work, in particular the programs that make up the operating system. C was adopted as a system development language because it produces code that runs nearly as fast as code written in assembly language. Thus, we used this versatile and fast language to program software for assisting students and teaching institutions regarding the IOE Entrance preparation.

## Background Introduction

The sole purpose of the IOE Entrance preparation kit is to automate the existing manual system of practicing for IOE entrance with the help of computerized equipment’s and full-fledged computer software, fulfilling their requirements listing lots of sets of questions for practice and mock test with the marks scored in their roll number. This project gives real experience for the student as if they are sitting in the IOE entrance examination. The required software is easily available and easy to work with. The entrance practice System, as described above, can lead to error-free, secure, reliable, and fast practice sessions. It can assist the user to concentrate on their other prior things for the entrance exam rather than feeling nervous by thinking about the way of entrance exam conduction. Thus, it will help the user in better utilization of resources and time.

This software revamps the system of data management and the current approach of examination (i.e., paper-based). It facilitates the institute for quick data record and access of a large number of students. This software can take the whole burden of systemized, error-free, and eased examination system for the institution taking account of the fastest result production, leadership board, and many more.

## Motivation

The idea of this IOE Entrance Preparation kit emerged from our recent experience that we acquired while we were preparing for the IOE Entrance. To get shortlisted in IOE, we need to practice as many questions as we can. The more we get familiar with the questions i.e., by practicing them, the more we gain an idea on solving the questions that we get queried in the Entrance. But on knowing these facts too, we couldn’t practice enough due to the smaller extent of question sources and the very rare number of examinations that we could attend. Also, we had a sort of fear regarding the computer-based examination, which is the format of IOE entrance as we were used to only paper-based examination systems which were obstructing our path to gain confidence.

Hence, to overcome these problems we got motivated to develop a software that could take a test based on MCQs and provide instant results of the tests. This aids a lot for the student to evaluate him/herself and build confidence in giving the actual computer-based entrance exam.

Furthermore, we thought to make this software a little bit commercialized. We thought of designing the software in such a way that it also aids the institutions that provide classes on IOE Entrance Preparation. This software would help them to keep a record of the students, examinations, students’ performance, and many more.

## Problem Definition

This project aims to flourish a software that will assist any institution providing classes for IOE entrance preparation regarding enrolling students to the institution, student data management, computer-based and error-free examination system, generating the result of the examinations, analysis of the student progress, access to all student data and add practice question by the administration.

## Objectives

The main objectives of our project are listed below:

* To help IOE entrance aspirants practice effectively for the entrance examination.
* To develop a commercial type of software for institutions conducting entrance preparation classes.
* To promote paperless examination to promote Environmental Conservation.

## Applications

This entrance preparation kit and Database management can be applied in the different educational and professional fields to prepare for various competitive exams by applying or changing certain features which are listed below:

* + - Engineering Entrance Examination Preparation
    - MBBS Entrance Examination Preparation
    - Ag and forestry Entrance Examination Preparation
    - BBS and BBA Entrance Examination Preparation
    - Army Entrance Examination Preparation
    - PCL Examination Preparation
    - Lok Sewa Examination Preparation

## Project Features

* + - It publishes the form which collects the information of candidate and their validity.
    - It stores the data of candidates for future reference.
    - One can view his/her overall performance and evaluate him/herself.
    - Students can take mock tests and practice tests.
    - It stores the date and time students took the practice exam and mock exam referencing the frequency of giving the exam by the student.
    - It calculates and stores the average marks of the students.
    - A special feature is that once the question’s answer is submitted it cannot be recorrected.
    - Special feature for admin to add new question set.
    - It displays the marks scored just after the submission of all the answers.
    - It displays the rank-wise result of the candidate among the students using “view leaderboard” features.
    - It gives the owner the right to delete all details of the students for new admission.
    - It stores question in a file and students' details in separate files.
    - It generates unique identification codes for the students automatically.
    - One cannot give the test and view his information until the identification code and citizenship number match as stored in the file which assures the personal detail security.
    - It calculates age itself and doesn’t take the invalid date if entered by the user accidentally or intentionally.

## Feasibility Analysis

### 1.7.1 Economic Feasibility

Economic feasibility analysis is the most commonly used method for determining the efficiency of a new project. It is also known as cost analysis. It helps in identifying profit against investment expected from a project. This program is highly economical and feasible. Since it is written in the C programming language, compiled by the C compiler using the IDEs like code block and VS code, it has not much of the economical investment but a lot of time has been invested on idea generation, research, and coding. So economically it is highly satisfying. Use of this software by any institute can instantly reduce the human effort and manpower regarding data management and examination which is added benefit for the institute. The student performance can also be drastically improved so a maximum number of students can crack the entrance examination which will add goodwill to the institute alluring more students to the institute in coming years. The software is also very easy to use and can be learned very easily.

### 1.7.2 Technical Feasibility

The technical feasibility study is the complete study of the project in terms of input, processes, output, fields, programs, and procedures. It is a very effective tool for long-term planning and troubleshooting. The technical feasibility study should most essentially support the financial information of an organization. It is a technically well-managed program whose source code can be easily modified for better improvement and for debugging any error.

### 1.7.3 Operational Feasibility

Operational feasibility refers to the measure of solving problems with the help of a new proposed system. It helps in taking advantage of the opportunities and fulfills the requirements as identified during the development of the project. It takes care that the management and the users support the project. It is very easy to use. Simply, a person who understands the English language can use this program. The software is very user-friendly in design. There are instructions in each part of the software that allows the user to easily operate the software. The options in the software are also self-descriptive making the software operation as simple as child-play.

### 1.7.4 Practical Uses

This software can be used by any of the institutes providing classes regarding IOE Entrance Preparation. They will be able to make a record of all the students that enroll in the institute in a very organized way such that they can access the data anytime in the future. They can take examinations and produce results instantly. They can publish the leaderboard to show which of the student is performing well and who is lagging in performance. They can amend the question sets at any time they want using the admin passcode. They can instantly access all the student data at once using their admin passcode. They can also reset the software and use it for the other academic sessions.

Not only this, but a student can also use this software at home to take the exam and check their status regarding preparation. They can access all the sets of question sets that are added by their institution and take short practice tests or a full mock test exam. In this context, the student will have access to all the question sources provided by the institute without needing any question banks and can give exams any time they want which will boost up their confidence to give the actual computer-based exam of IOE entrance.

To conclude, the software is very feasible in the context of efficient IOE entrance preparation, both for the institute and the students.

## System Requirement

### Software Requirement

This program does not require a lot of software. This program can run just in any operating system as the C-language is machine-independent. After the compilation process is performed, a .exe file is generated which runs in any OS without the need of any compiler. It also doesn’t consume much ram as the data in the file are accessed at the time of their need and all the data are stored in the txt file. It is a very lightweight and efficient software.

### Hardware Requirement

This program is an economically feasible program so it doesn’t require a lot of hardware. Any computer either be laptop or desktop that has an operating system installed in it with the hardware requirements of the OS can run this program. It can run on the lowest-performing PC available in the market. Thus this software is a boon for all those students having low system requirements due to their poor economical condition. Also, PCs having higher specifications can support the program with ease.

# LITERATURE REVIEW

## “Enroll Yourself” Block

This block takes the personal details from the students and enrolls the students in the institutions by providing them a unique automatically generated identification number. The personal details entered will be recorded for future reference. It also checks if the person has already enrolled or not by comparing their unique identity details like citizenship number.

## “Take Practice Exam” Block

This block takes a small test of the students where the students can choose the number of questions for the exam. Here, 1.5 minutes will be given to the students to answer one question. The score of the students will be recorded for future reference.

## “Take Full Mock Test” Block

This block takes the full 140 marks exam of the student where the student will not be able to select the number of questions but can select among the question sets. The students will be given a maximum time of 2 hours for the exam after which the examination will end automatically. The score of the students will be recorded for future reference.

## “View Overall Score” Block

This block shows scores of all the examinations the student has given based on the data recorded from the “Take Practice Exam” and “Take Full Mock Test” sections. The data will be displayed according to the identification number entered by the user in a tabular format.

## “View Leaderboard” Block

This block shows the data of all the students and organizes them in descending order based on the overall average percentage of the students.

## “Amend Question Set” Block

This block allows the admin to add the question set or add questions to the existing question set. The admin will be recognized through a certain passcode.

## “View All Enrolled Students” Block

This block allows the admin to see all the students that are enrolled in the institution and delete all the data of the students for a new session. The admin will be recognized through a certain passcode.

## “Exit” block

This block on selection exits the program.

# DESIGN AND METHODOLOGY

## Introductory Block

This block prints the name of the software, student details of the developers of the software i.e. team members. The student details include the Name and Roll Number of the team member. It also displays the faculty, college name, semester, and Enrolled year of the project team member. At last, it asks the user to press “Enter” to continue and directs to the option selector block as the user presses Enter.

ENTRANCE PREPARATION AND DATABASE MANAGEMENT

(Project details)

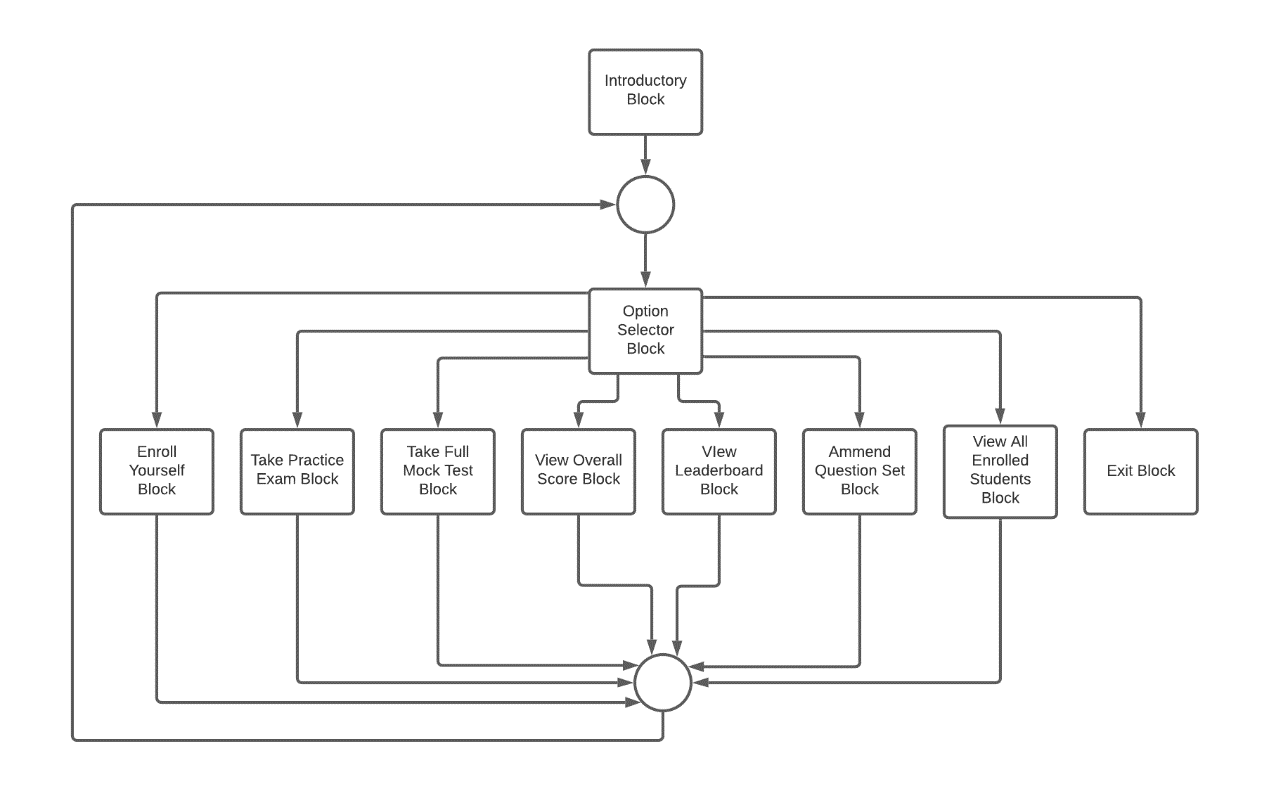
Enter to continue

###### 

Figure 3- 1: Introductory Block

## Option Selector Block

This Block consists of an Option\_Selector() function that lists options from 1 to 8 that links to the 8 blocks of the program. First of all the list of the options is displayed and then the user is asked to select among the options using the labeled number of the option. Then an integer input is taken from the user, according to which the program navigates to that part calling the respective function to display its content.

Figure 3- 2: Option Selector Block

## Enroll Yourself Block

This block asks the students who want to enroll in the institute to ask for their details, record them and provide a unique identification number to the student specifying that he/she has enrolled in the institute. The details of the students include their name, gender, date of birth, and Citizenship Number. There are functions to calculate the age from the date of birth and store it with other details in a file. There is also a function that checks if the citizenship number entered by the user is pre-registered or not and asks the user to input another citizenship number in case of registration.

Enroll Yourself

Enter name

Enter Date of Birth

Enter Gender

Enter Citizenship

Generates Unique Identification code and displays it

Figure 3- 3: Enroll Yourself Block

## 3.4 Take Practice Exam Block

This block provides a short test for the user. The user can select the question set and the number of questions from a minimum of 5 to a maximum of 20 questions. Each question will be given 1.5 minutes to answer.

## Take Full Mock Test Block

This block takes the full mock test of 100 questions where the time of 2 hours will be provided to the student to give the exam. The score of the student goes on increasing if the answer is correct and is displayed after each question. After all the questions are answered, the total score of the student is displayed and it is recorded with time in a file for future reference. The student giving the examination is identified with the help of their identification number which he/she has to enter before starting the exam. After 2 hours, the exam automatically ends.

## View Overall Score Block

This block displays the overall score of the exams given by the user. This block identifies the person with the help of the identification number and prints the overall score of the person in the practice exam and the mock test, taking data from the file where it was stored previously.

## View Leaderboard Block

This block displays the rank of students according to the average marks of the students which are stored in a file by calculating. It displays identification numbers including total marks and rank.

## Amend Question Set Block

This block allows the admin to make a new question set and input the questions and answers and also to add questions to the existing question set. To access this feature, the passcode is required which is already defined in the program and is provided to the administration of the institution only. When the admin enters this block, first of all, it asks passcode to access the block. If the passcode is wrong, it asks to enter the correct passcode, and if the passcode matches it shows two options, either to make a new question set or to amend the existing question set. Based on the selection made by the user, the program works.

## View All Enrolled Students Block

This block lists all the details of the enrolled students to the administration. To access this block also, we need the admin passcode. In addition to this, this block also contains a function to reset the software which clears all the data stored in all the files and makes the software ready to be used for a new session.

## Exit Block

The only function of this block is to exit the program.

# IMPLEMENTATION AND RESULT

## 4.1 Introduction:

Project implementation is the phase where theoretical design, visions, and plans turn into a working system. Its success depends on many internal and external factors. It is the most critical part of the project as it is the step of achieving the new system that will work and be effective.

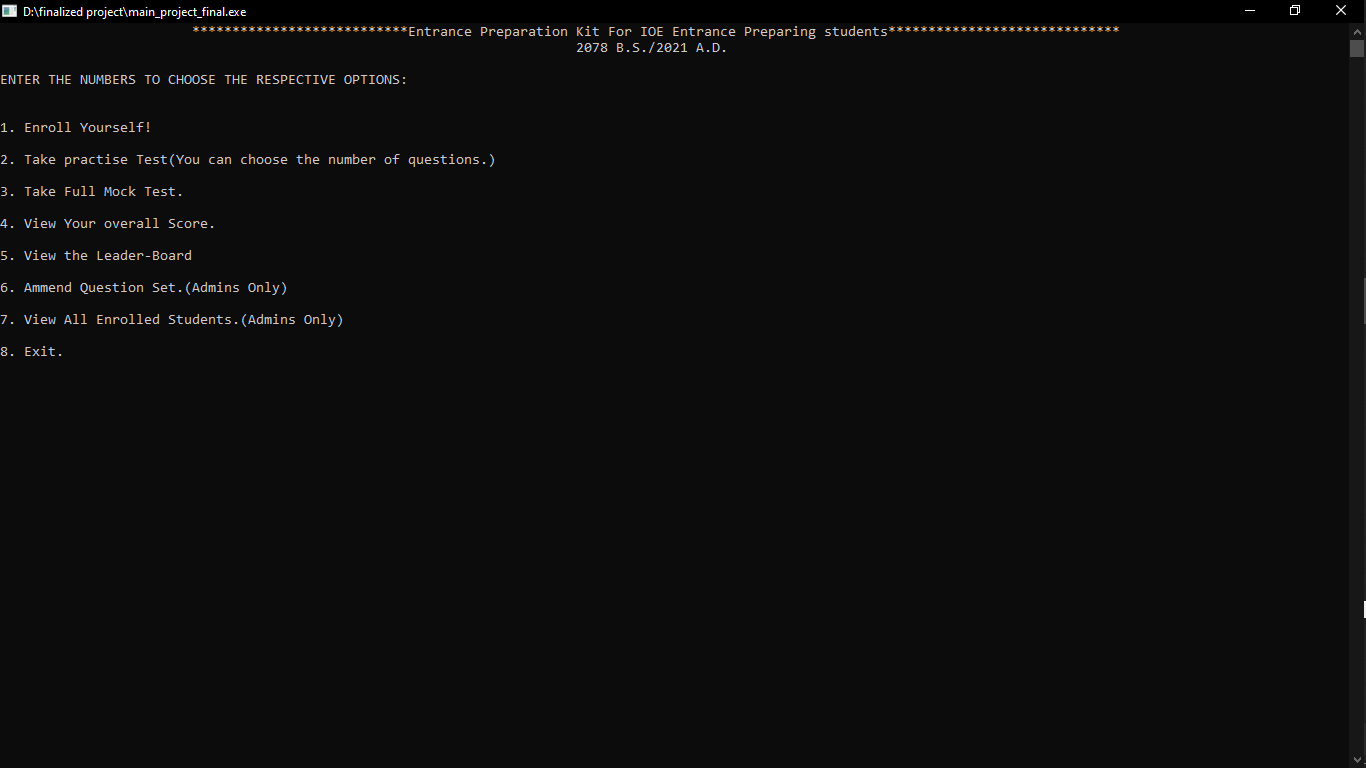
Careful planning, investigation of the existing system, designing of methods to change the system, and analysis of the advantages of change come under the stages of the implementation cycle.

### 4.1.1 Some screenshots of results:

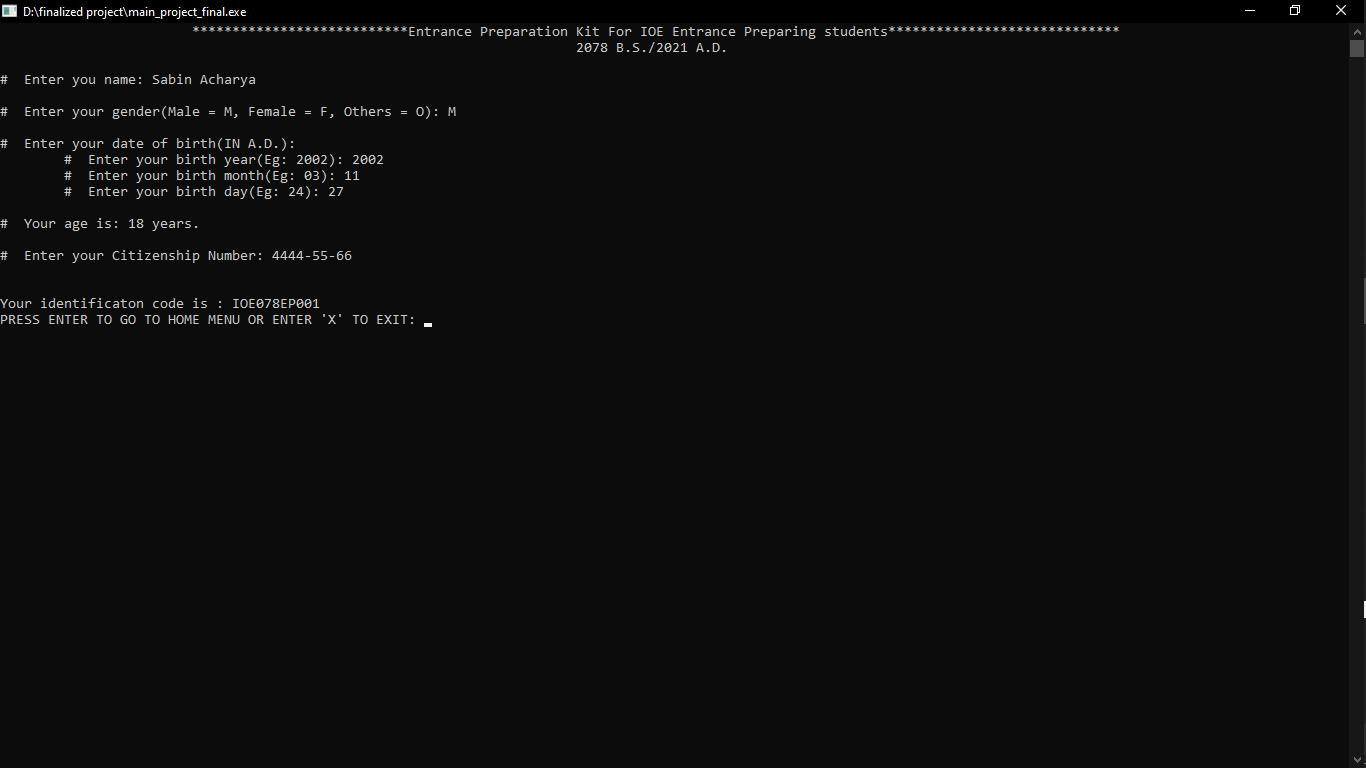
Introductory Block



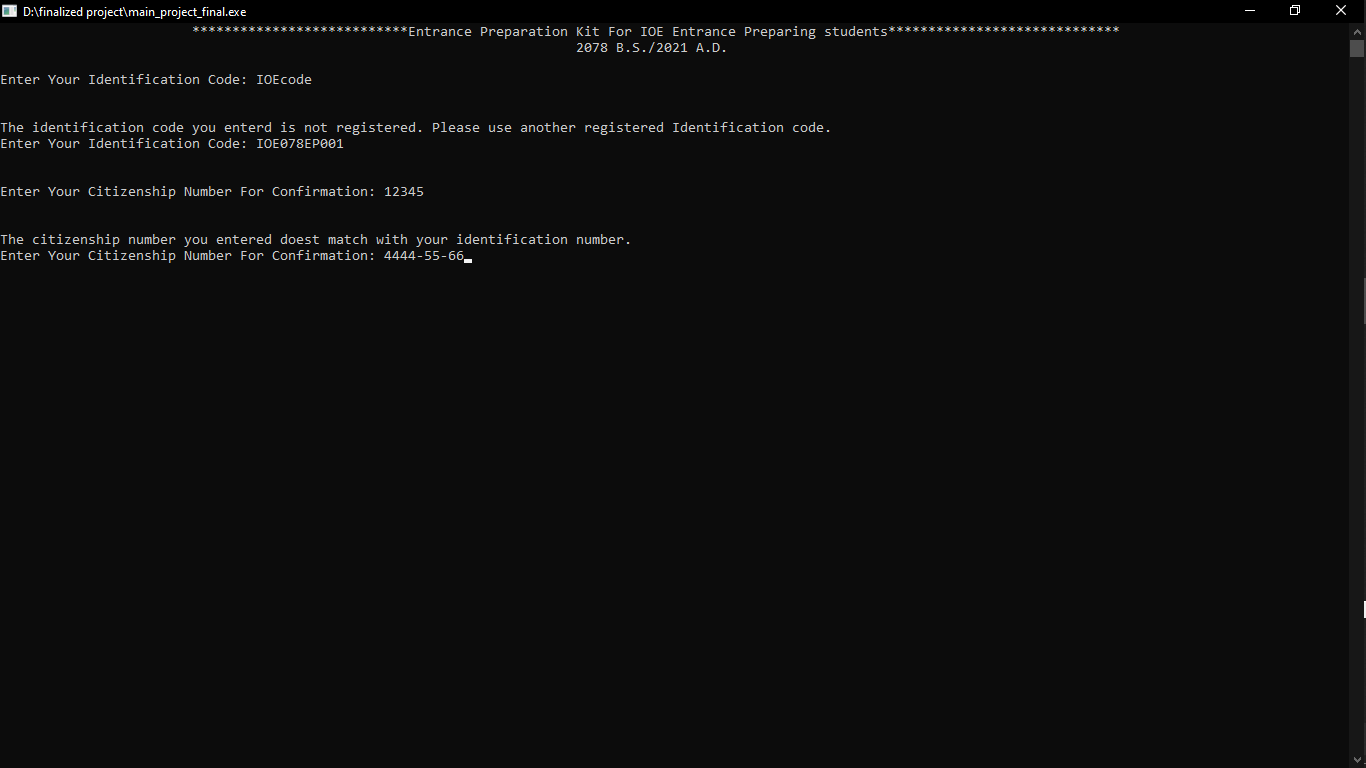
Option Selector Block

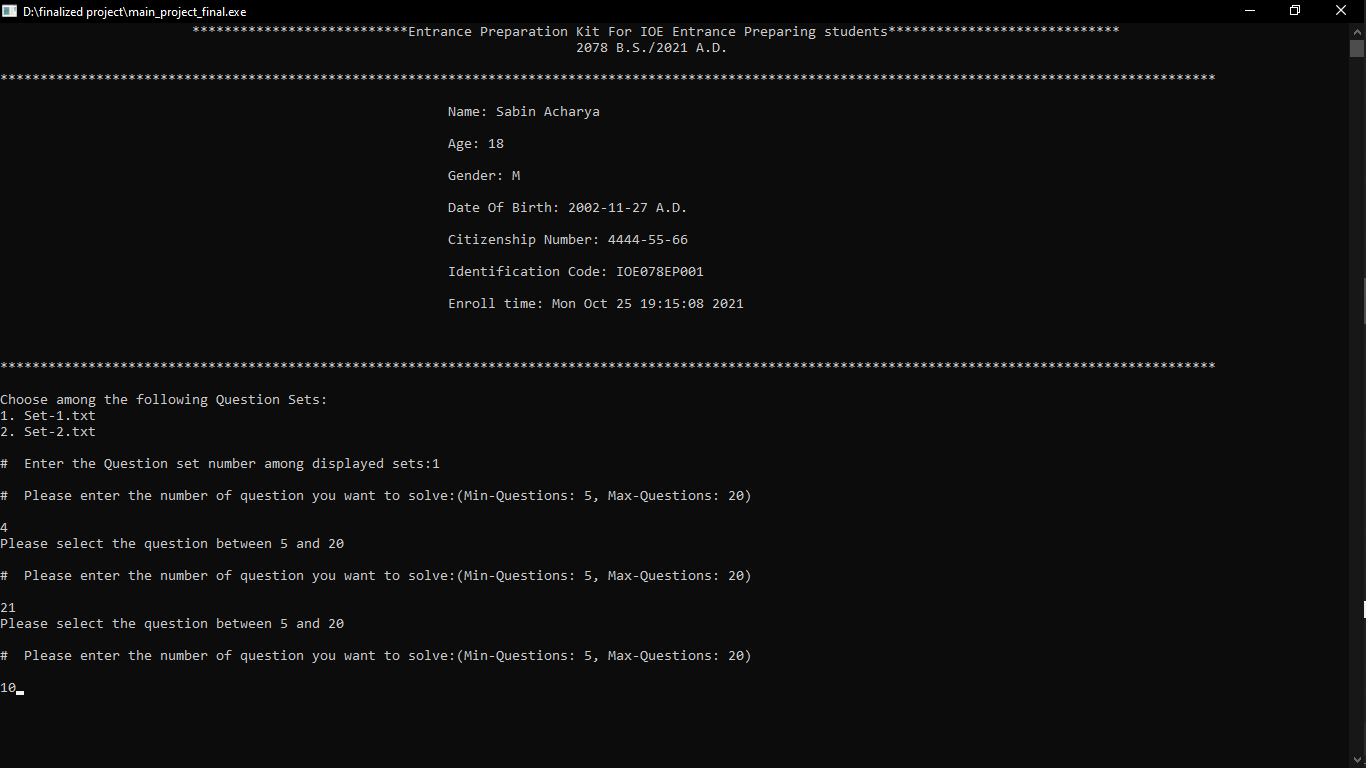


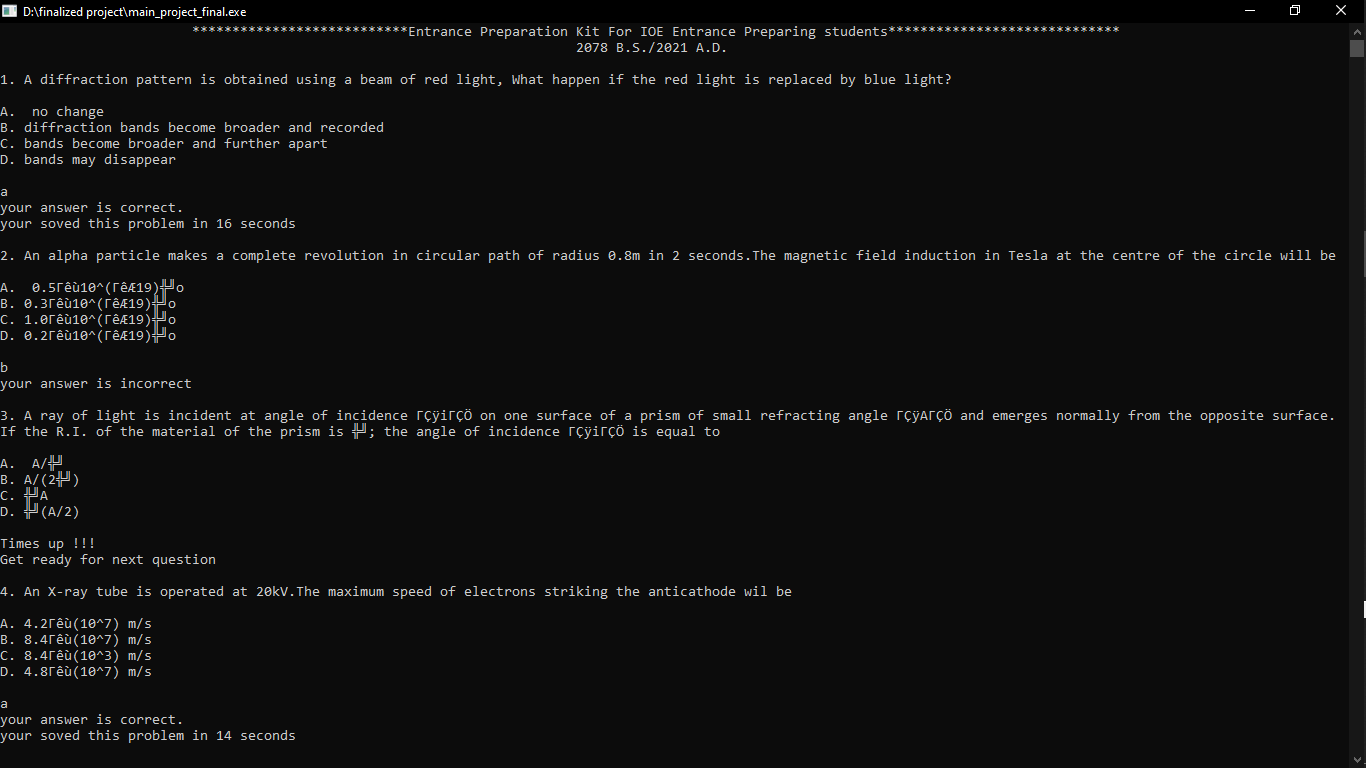
Enroll Yourself Block

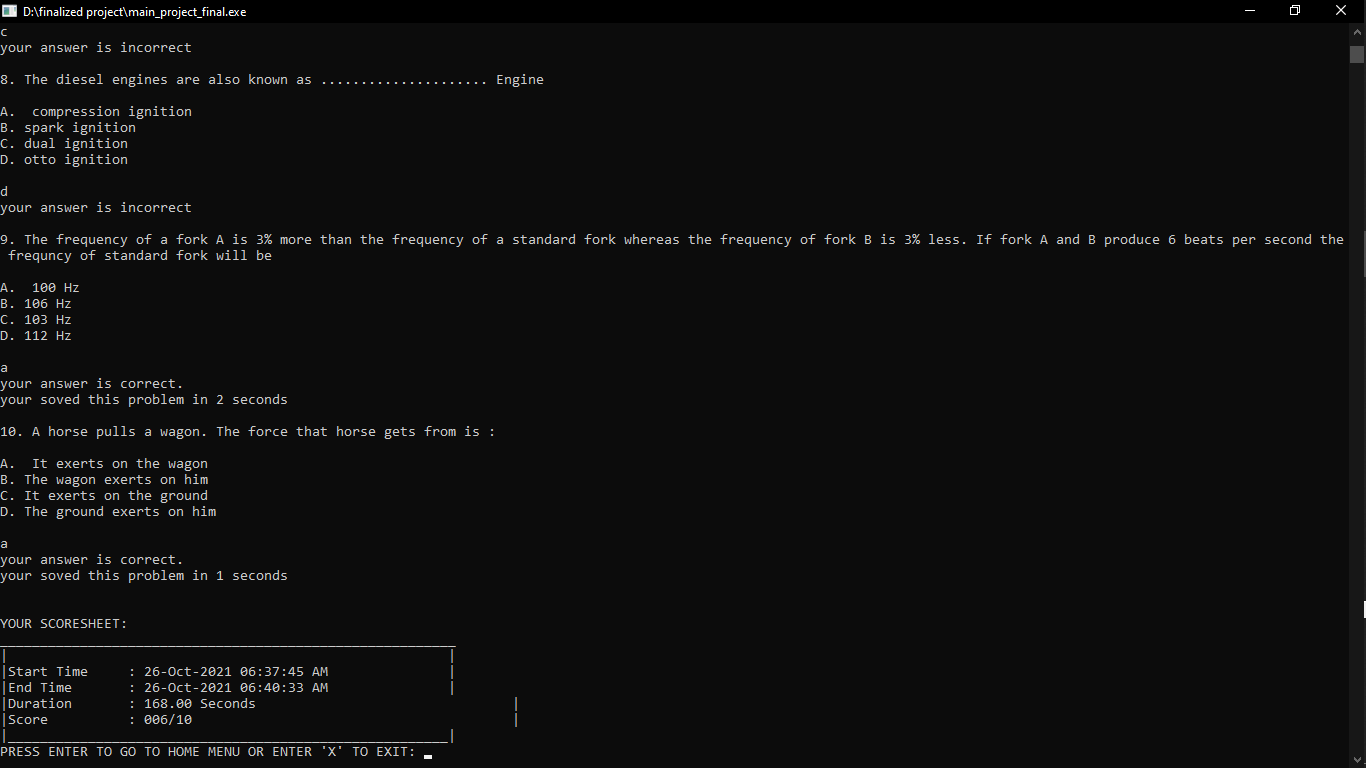


Take Practice Exam Block

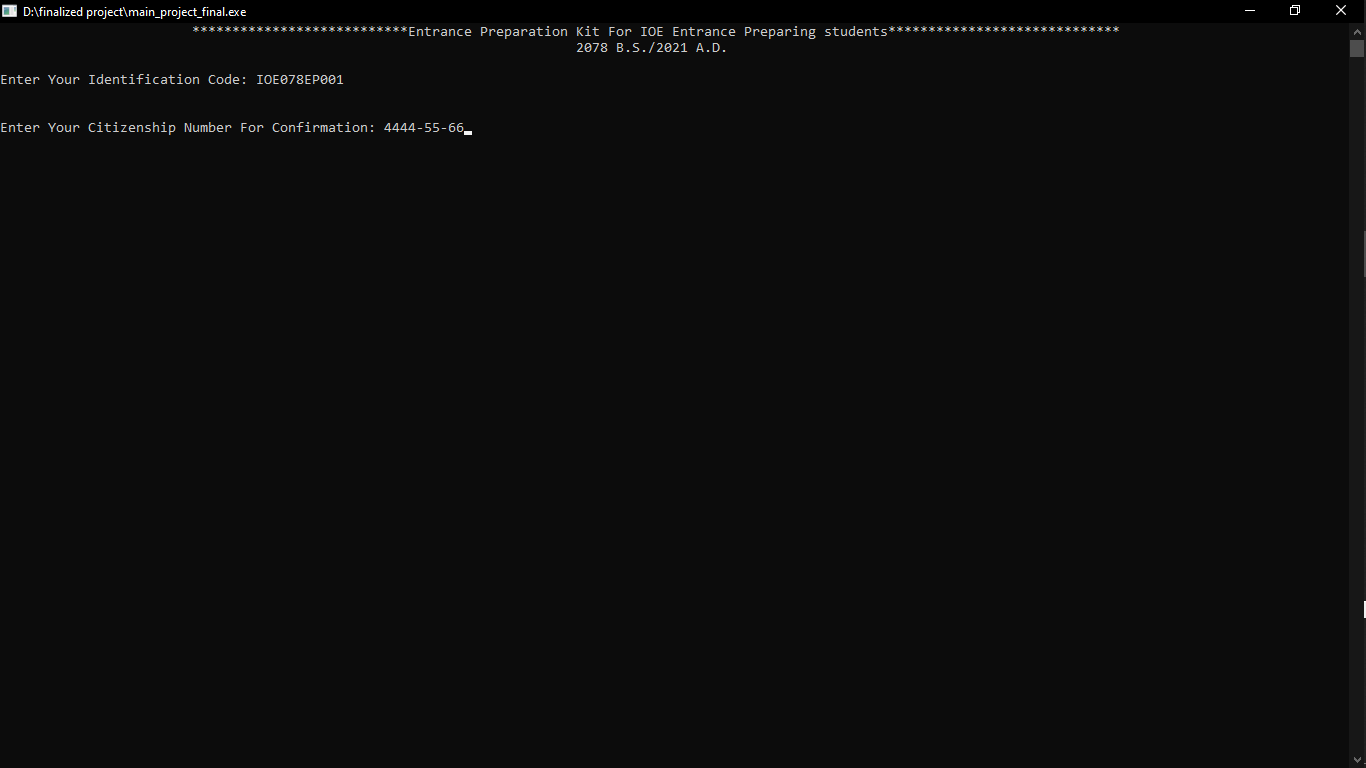


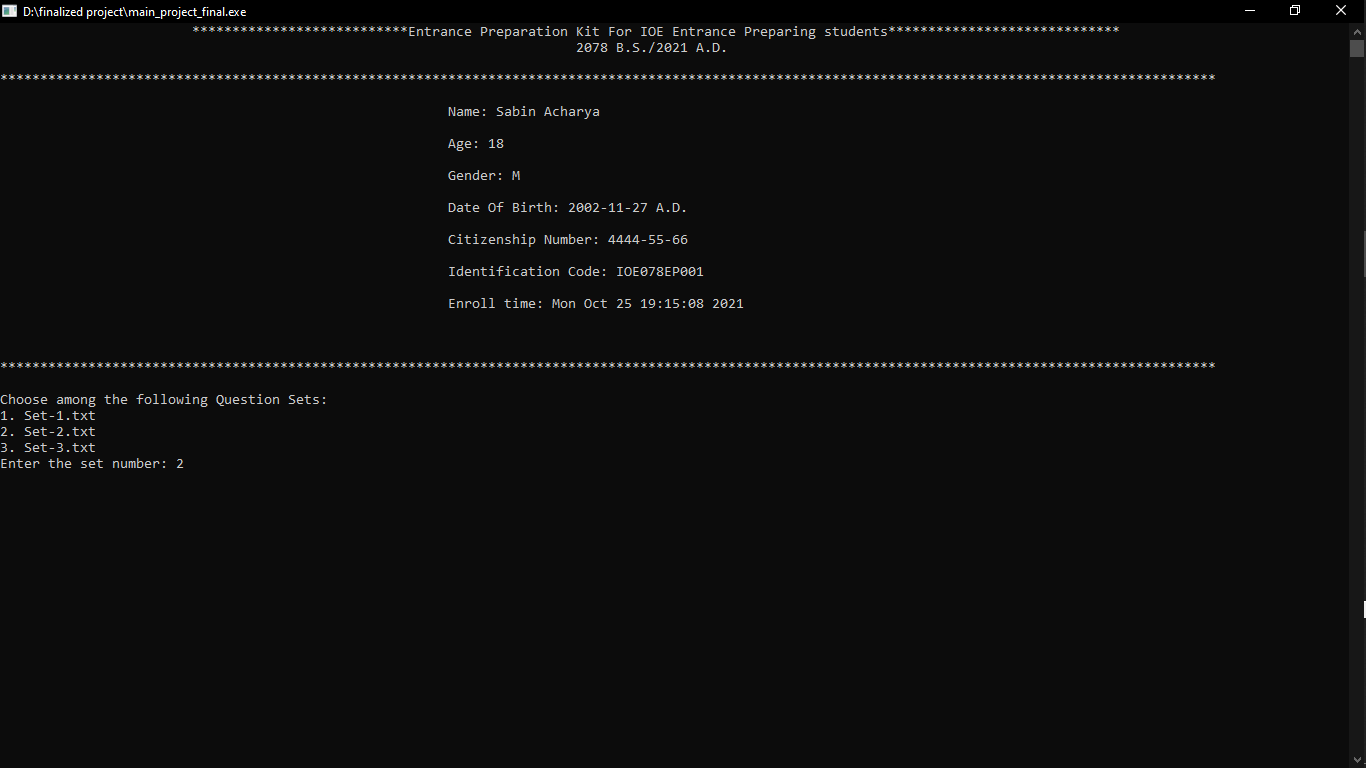






Take Full Mock Test Block

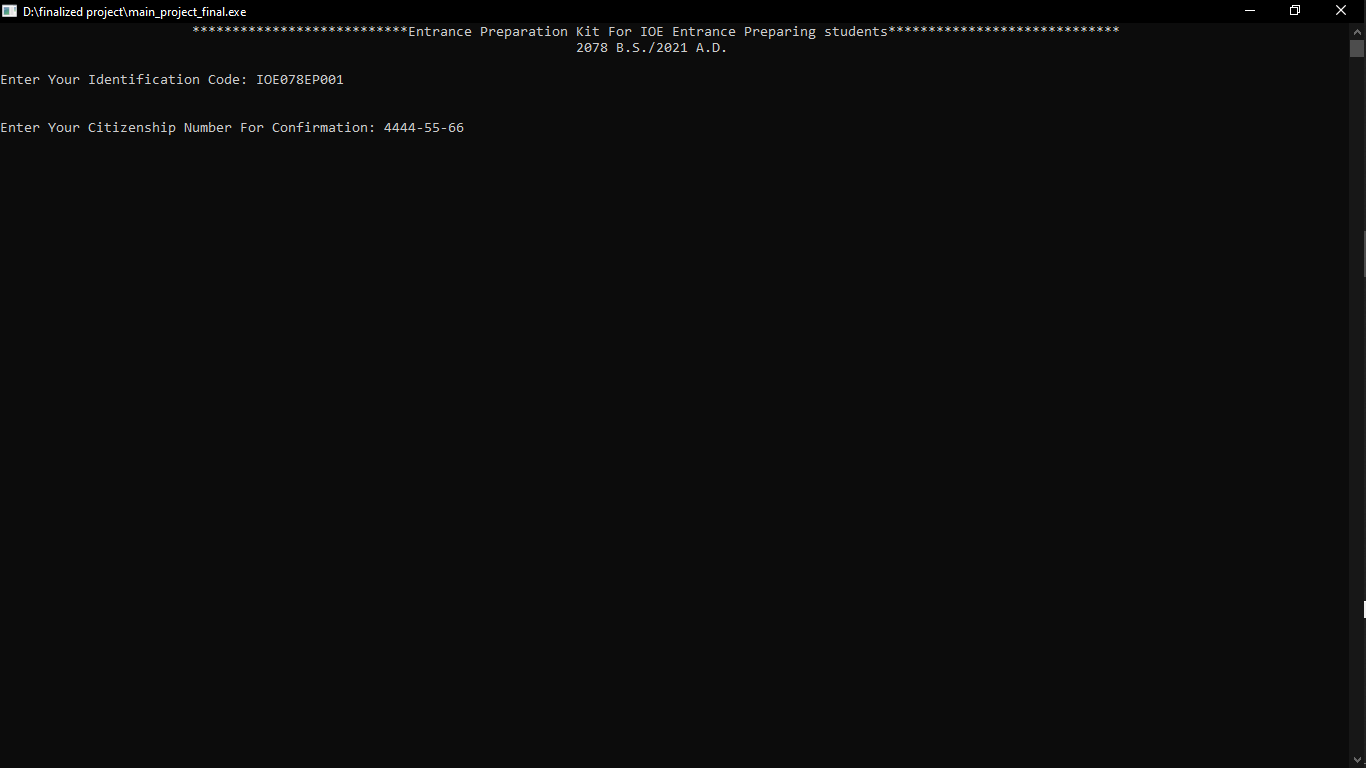


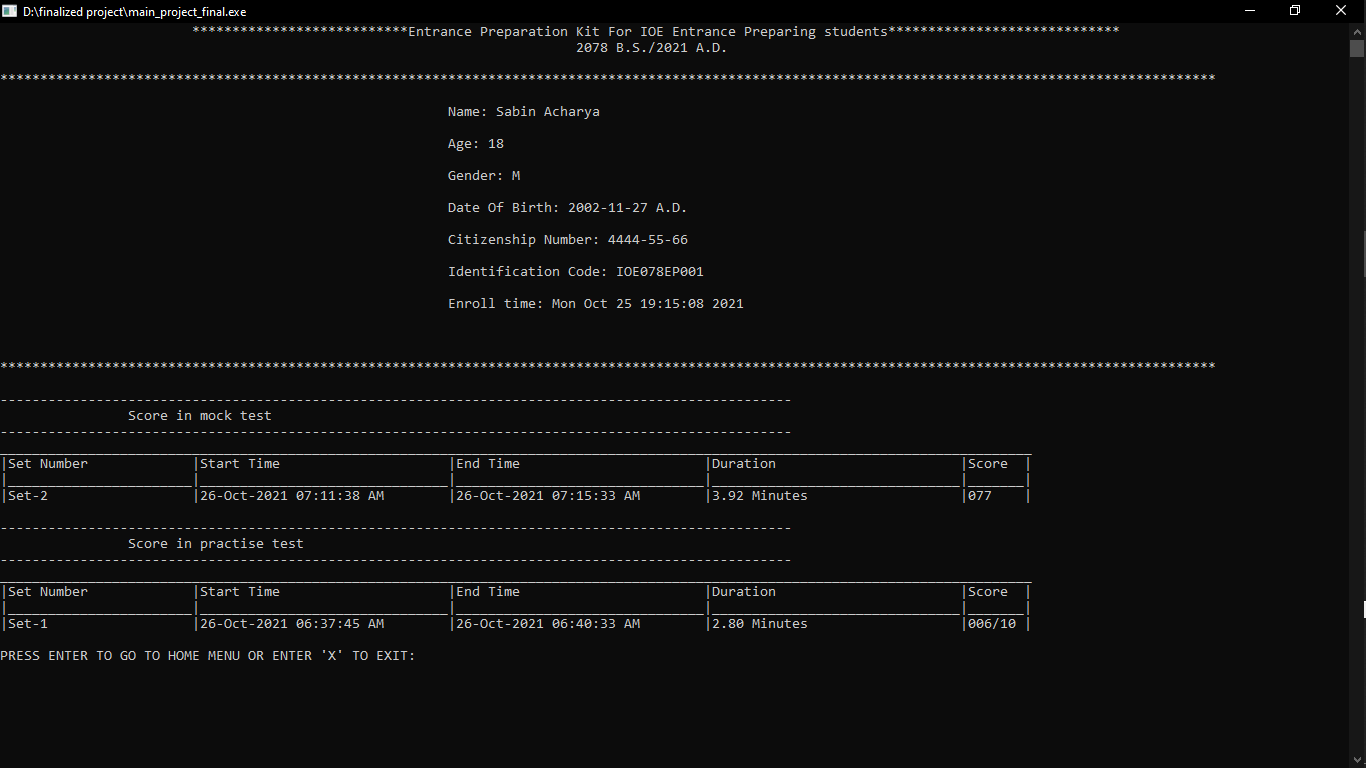




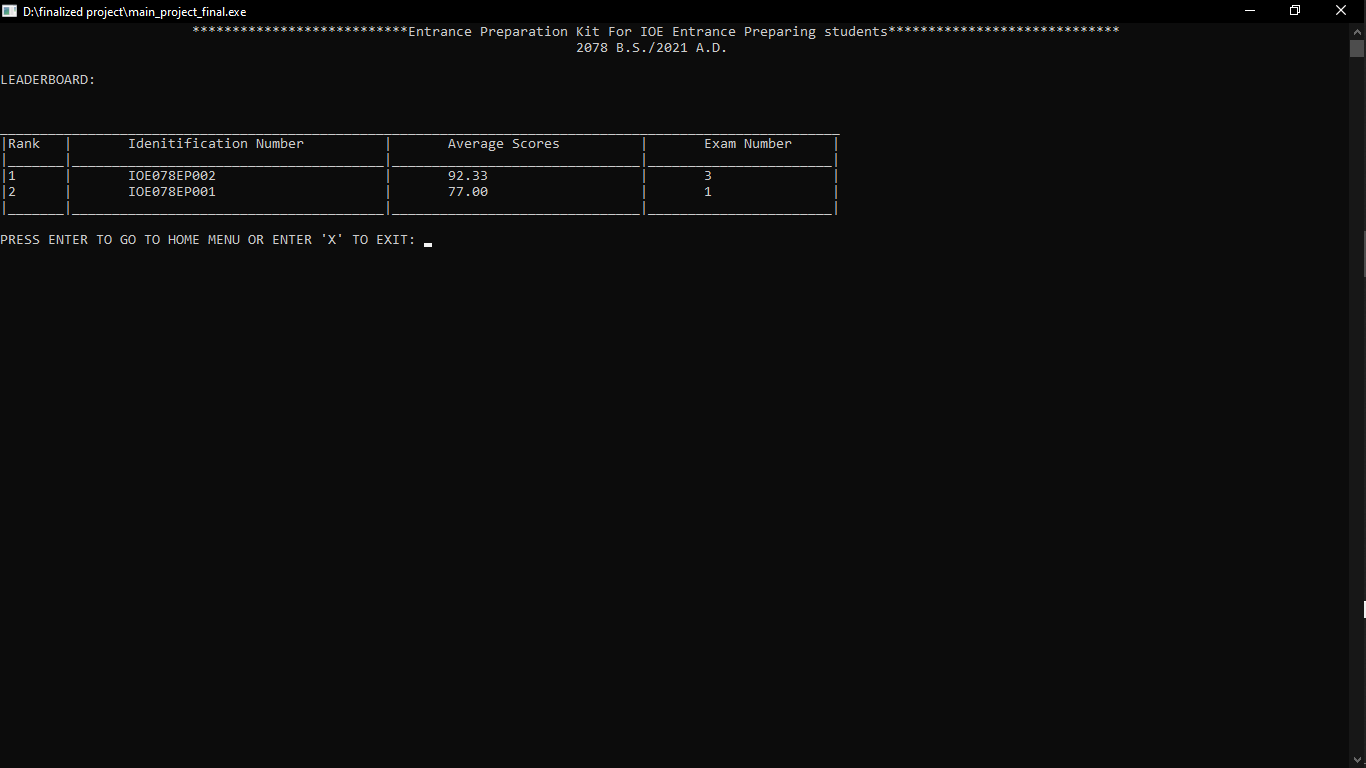


View Overall Score Block

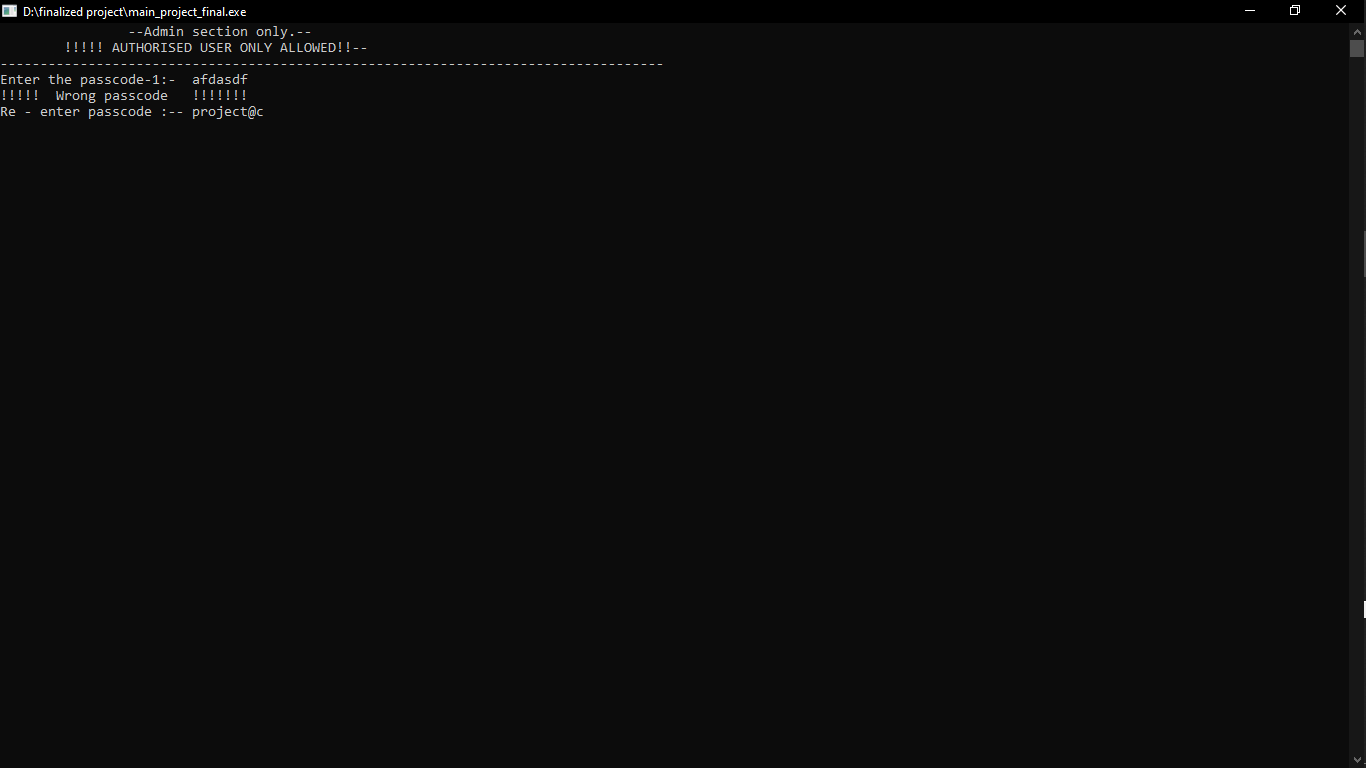


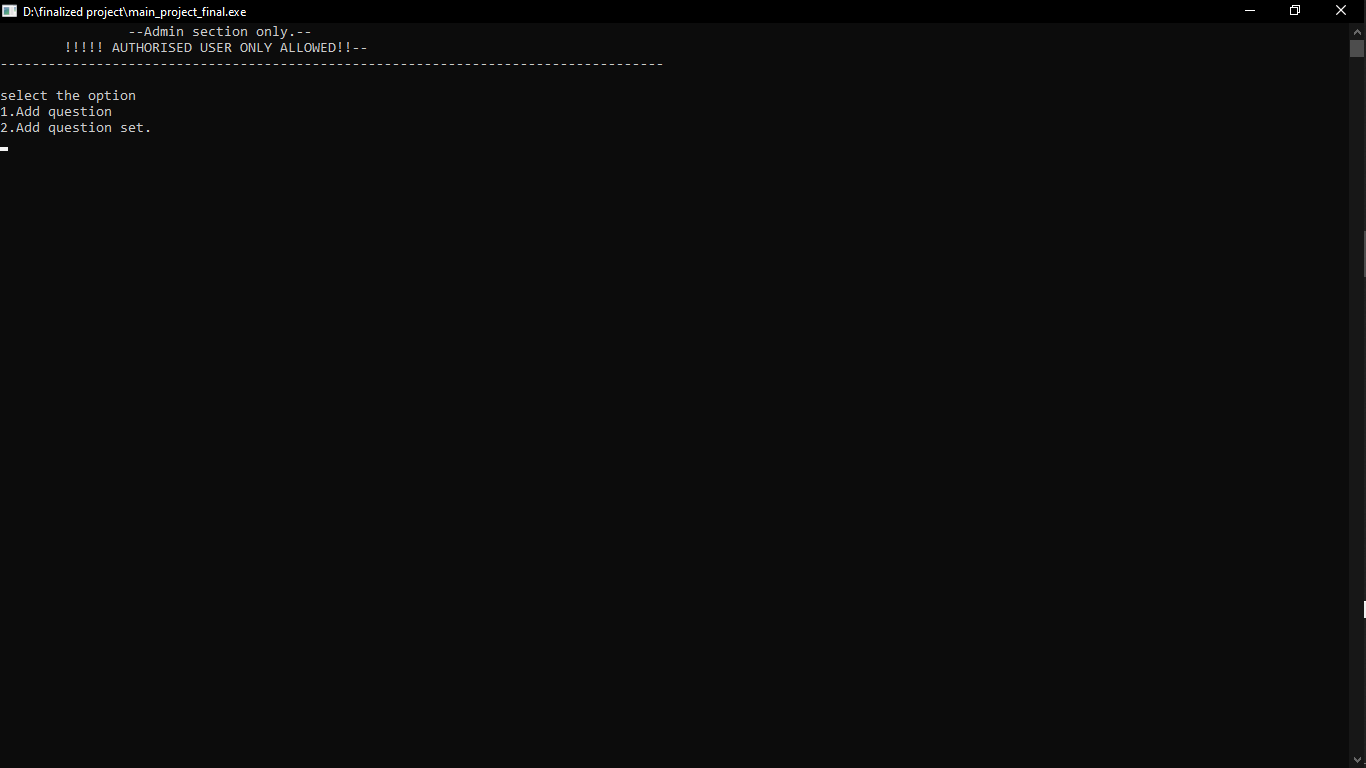


View Leaderboard Block

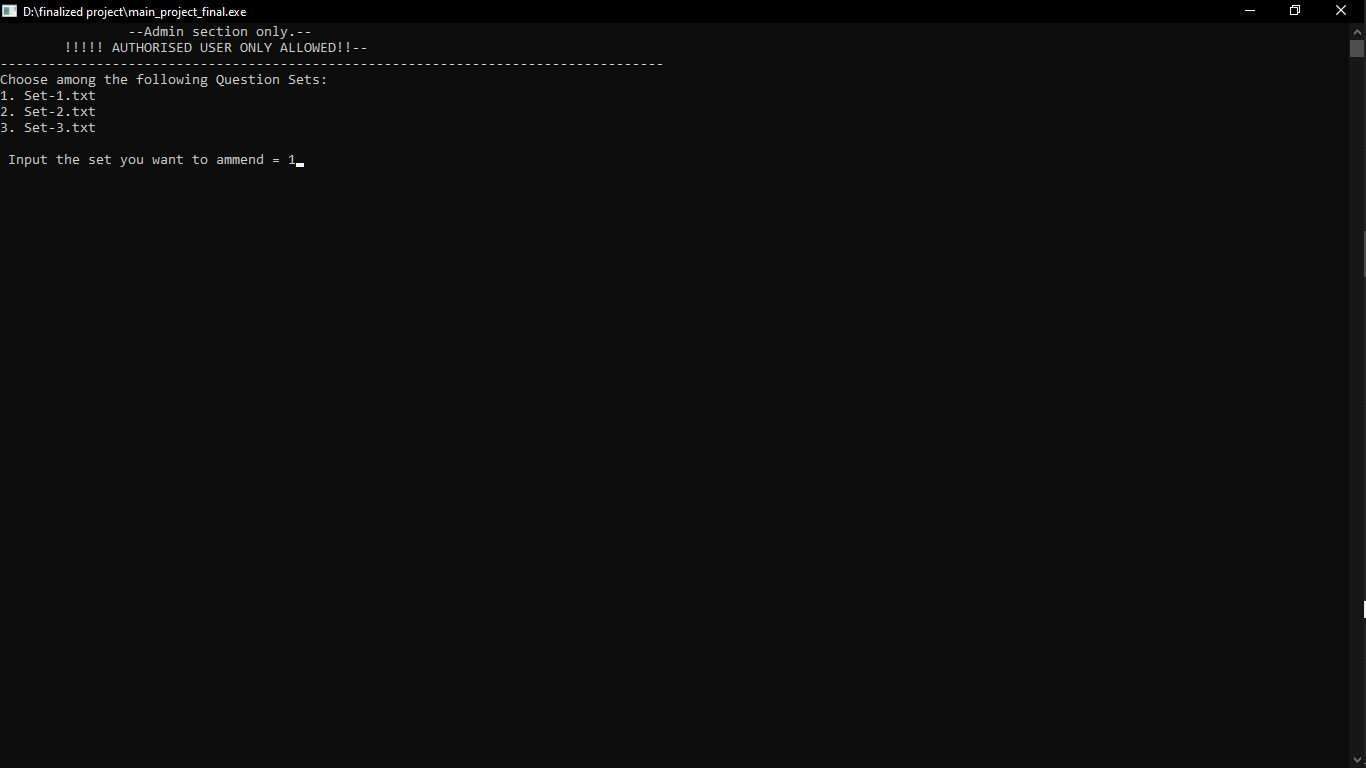


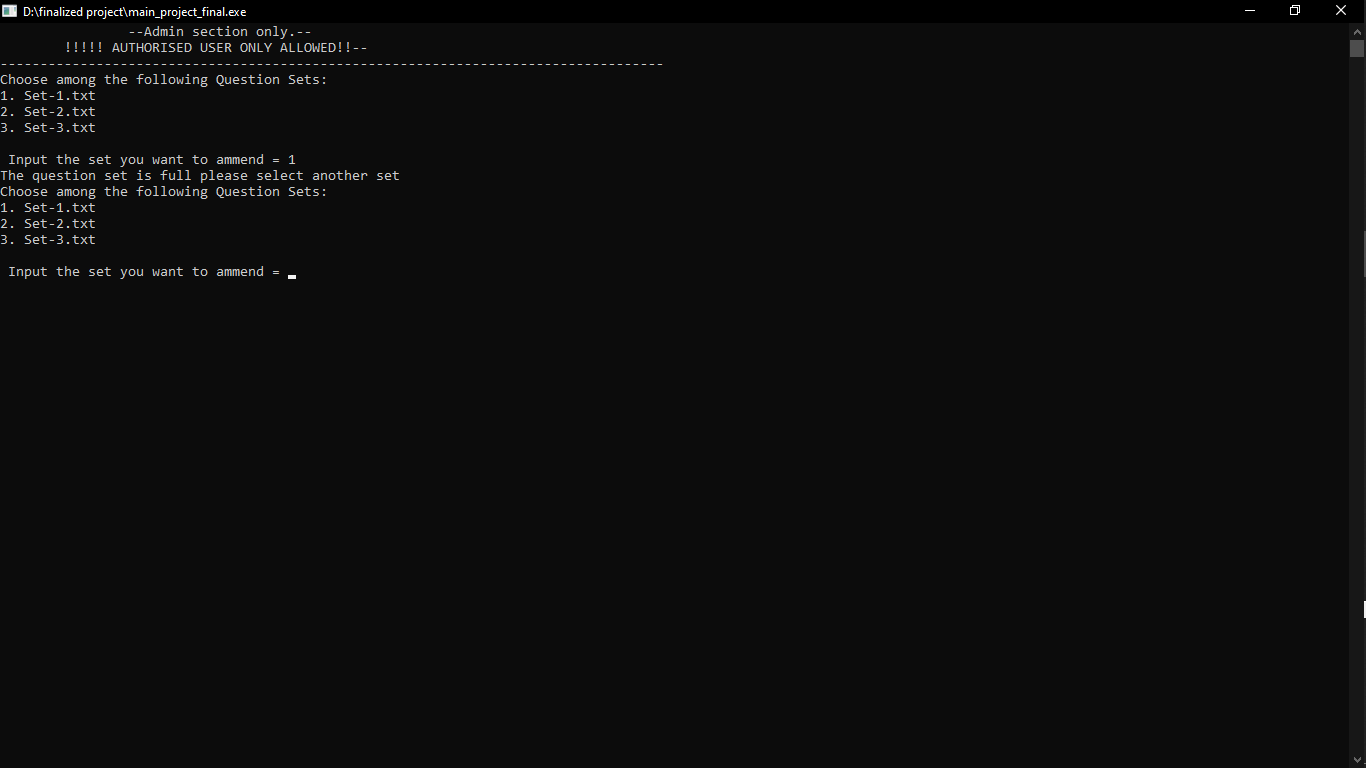
Amend Question Set Block

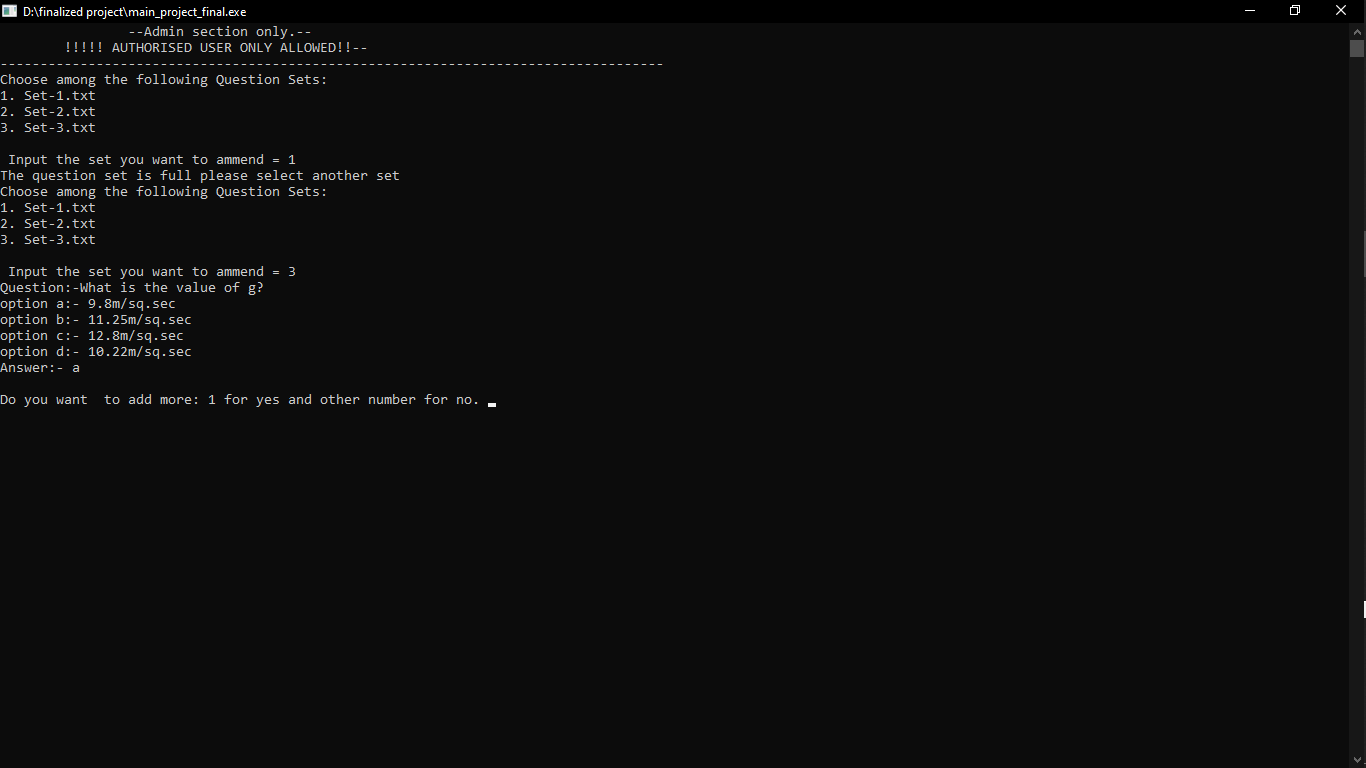


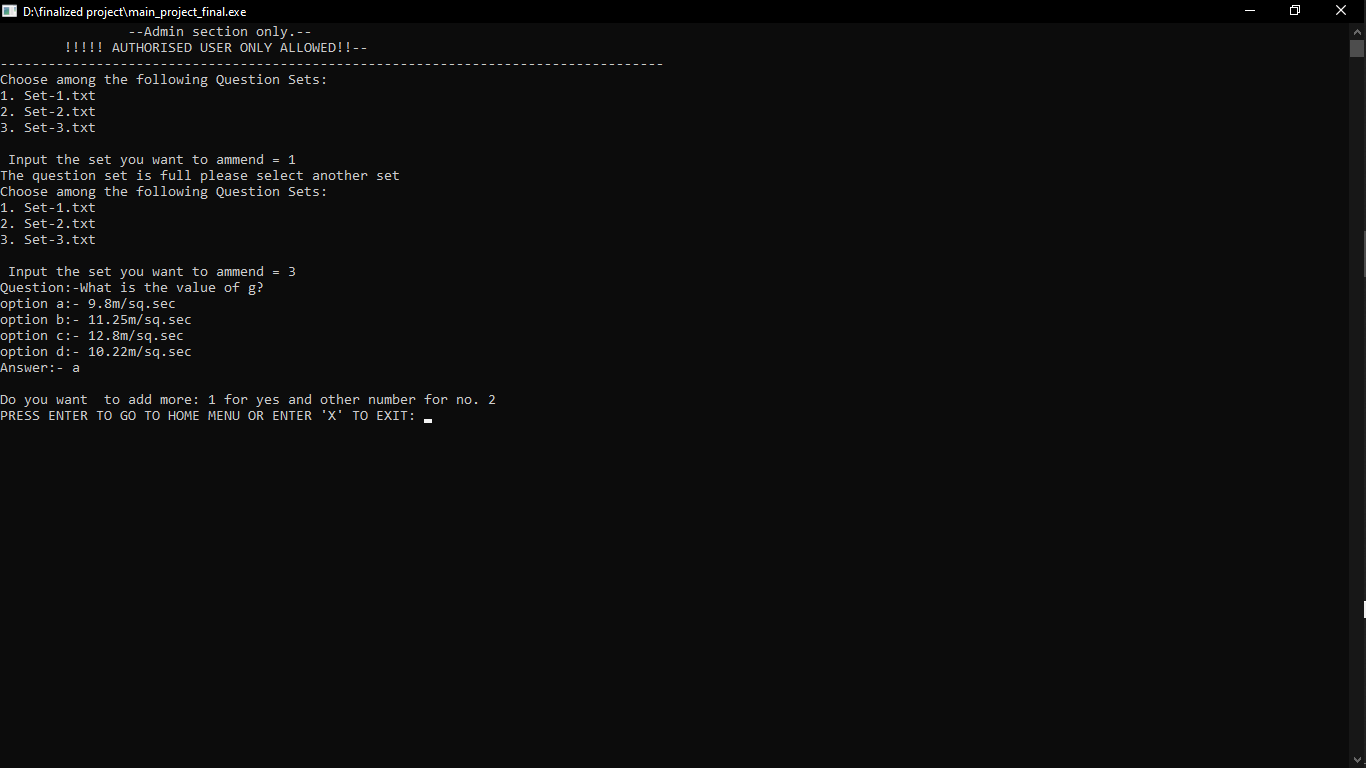


Option 1: Ammend Question Set.

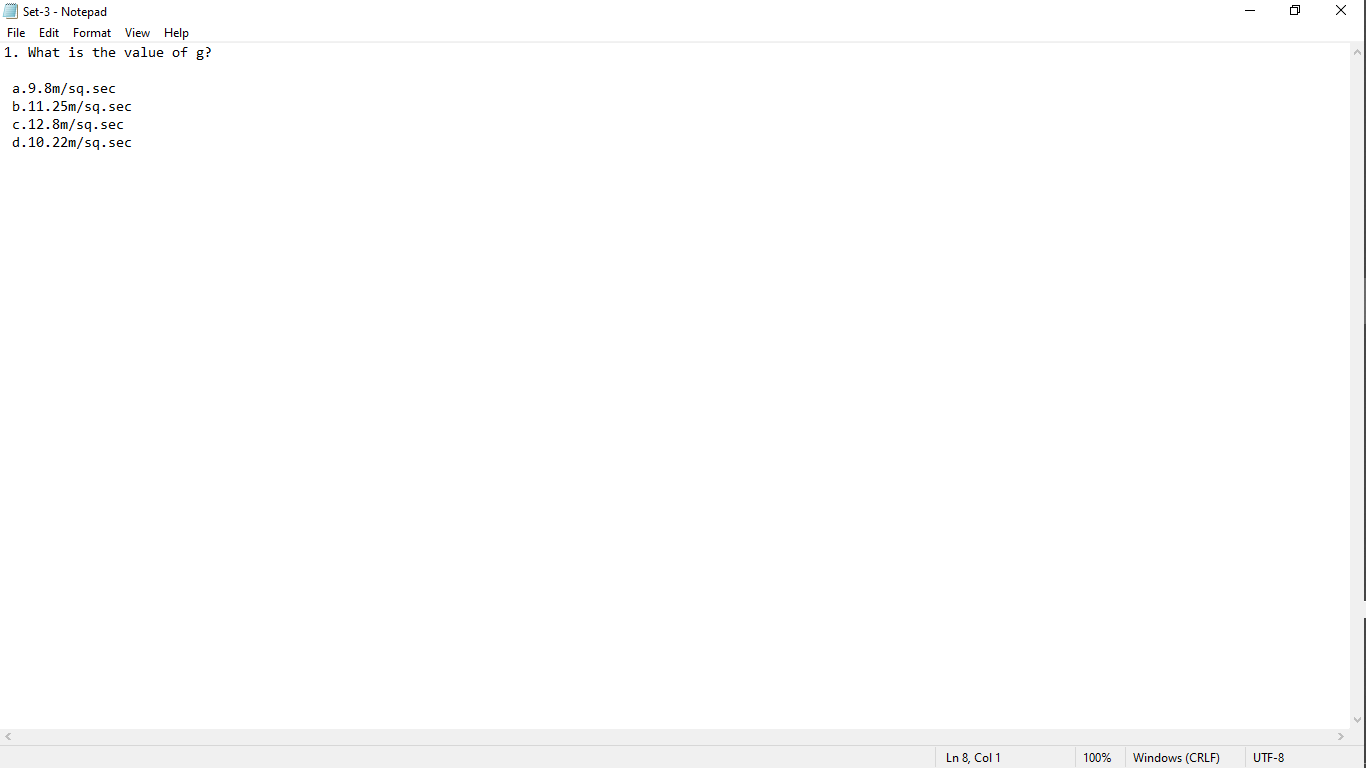




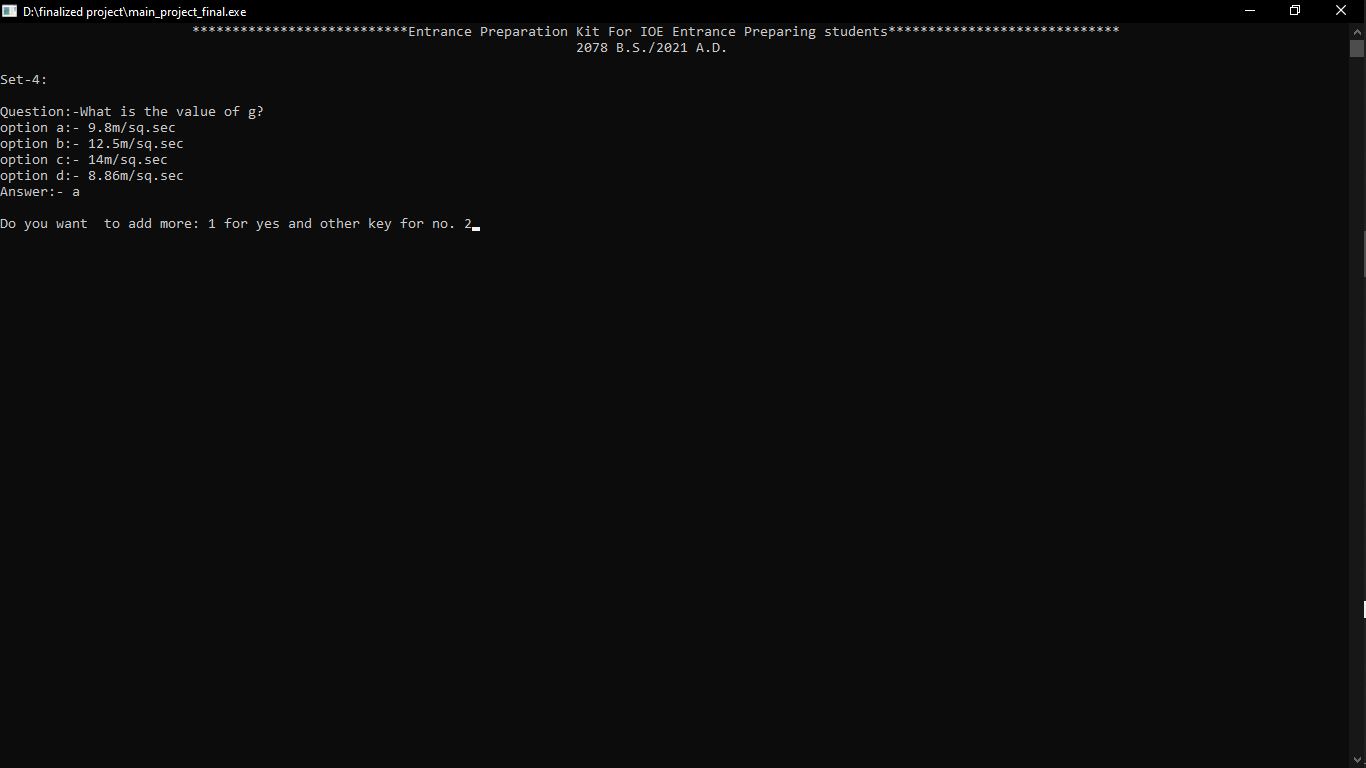




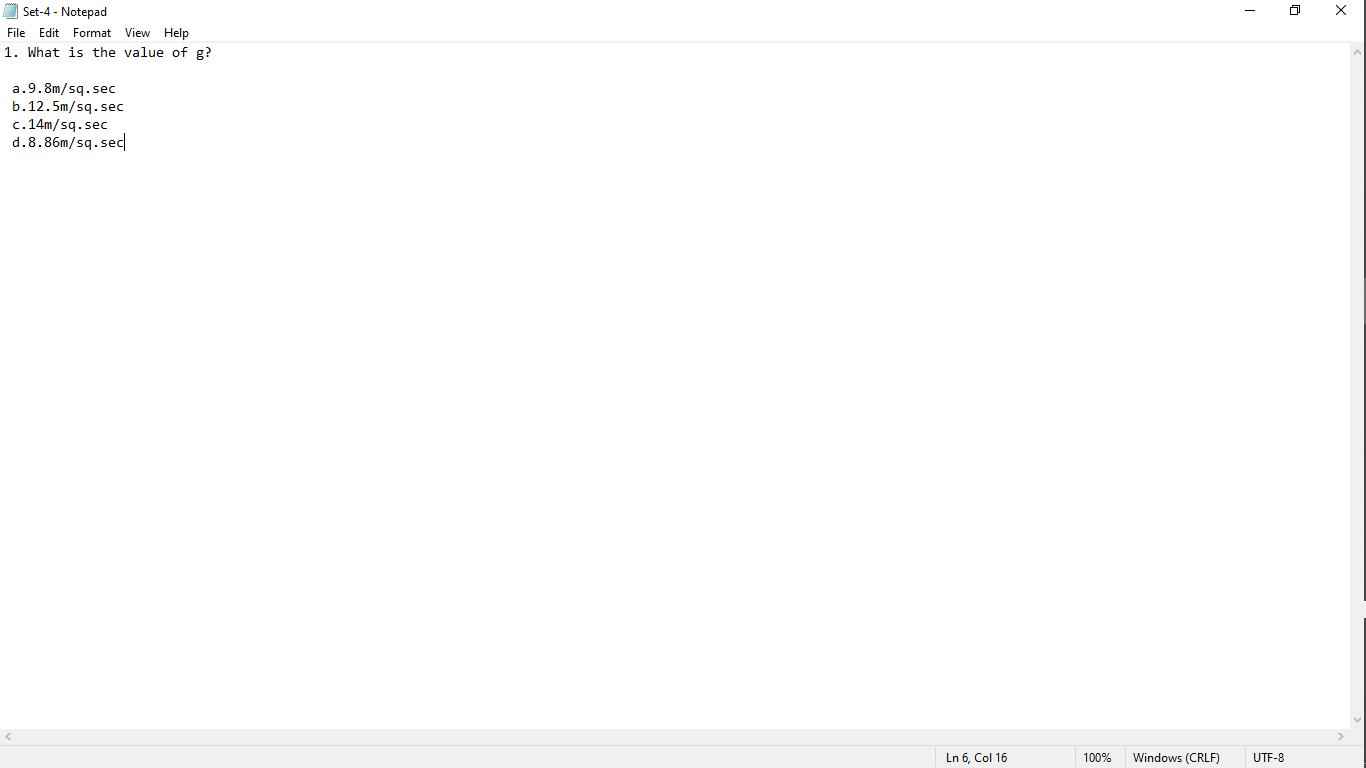
The entered question is saved in existing set which is later displayed while taking mock test:



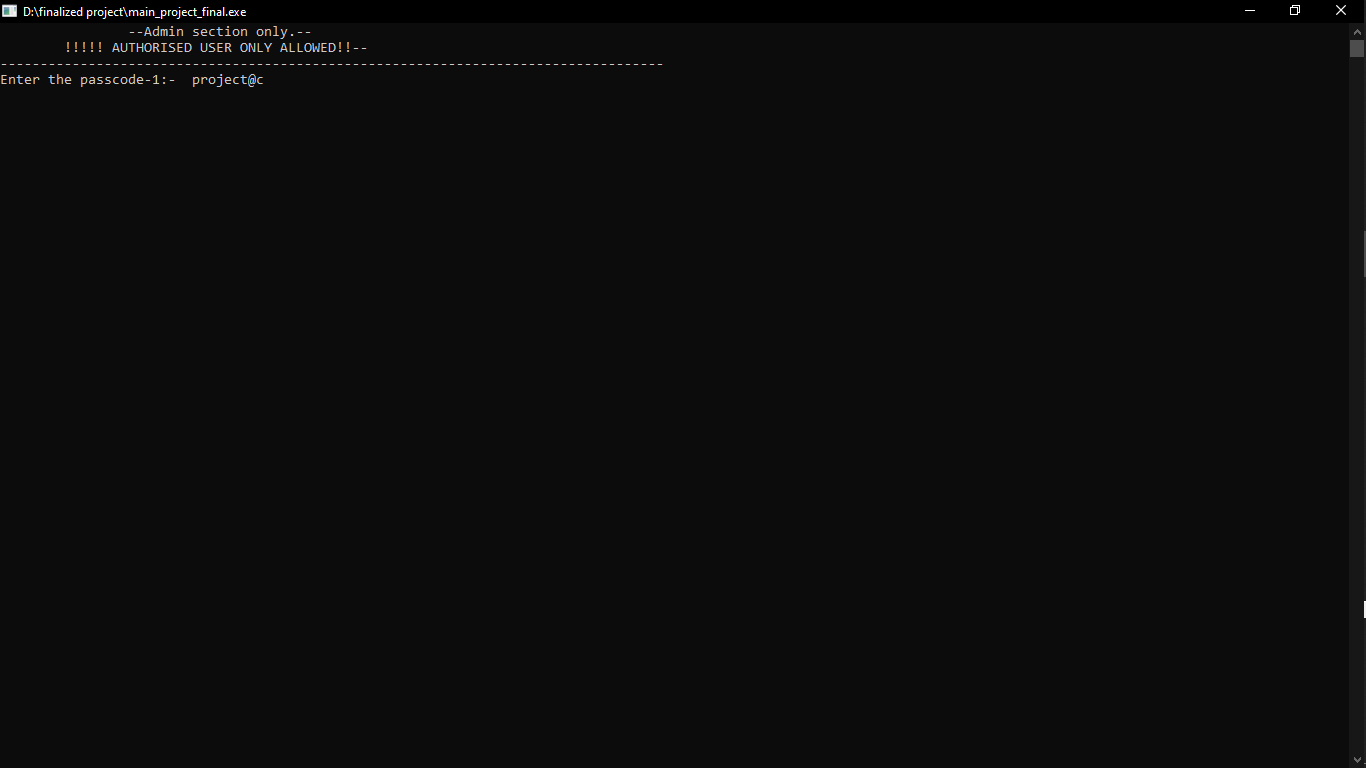
Option 2: Add Question Set.

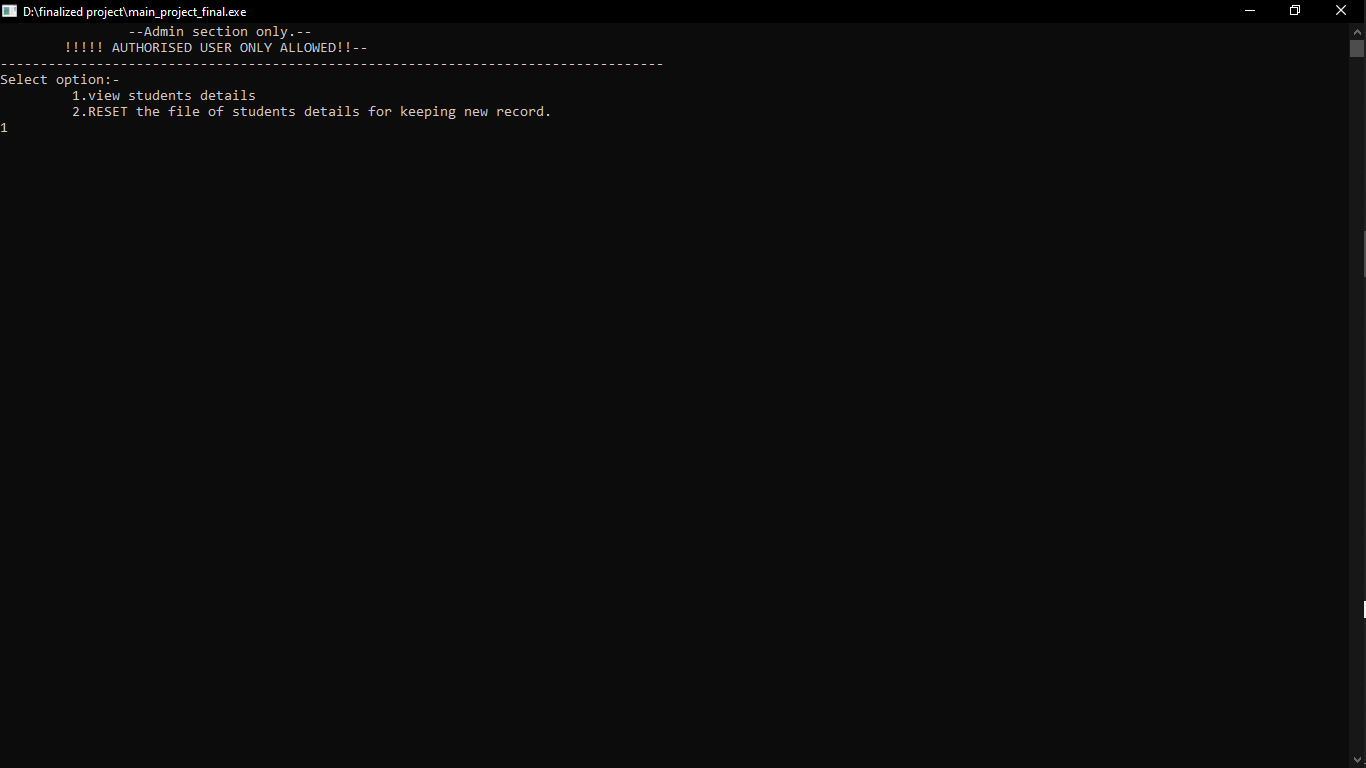


Now, a new set is formed and questions get on adding to that set. Here new set-4 is formed.

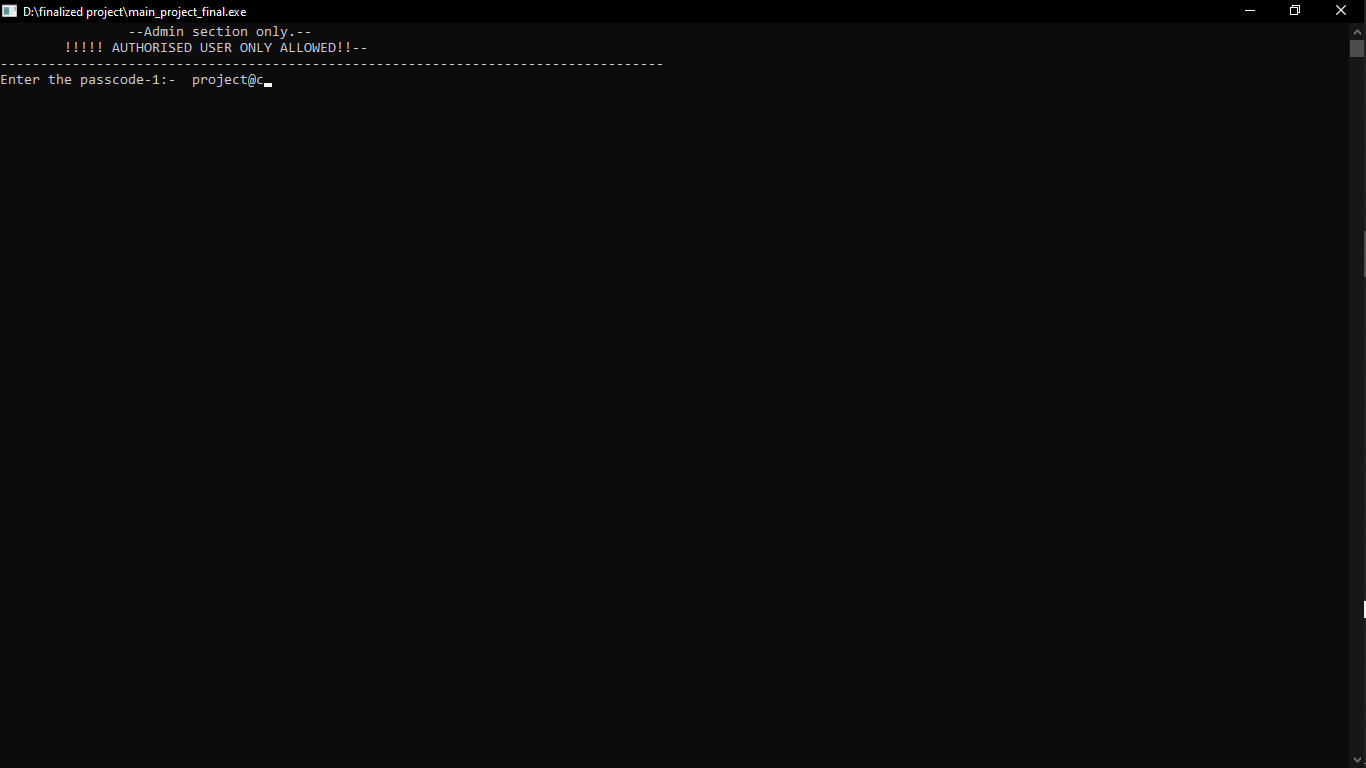


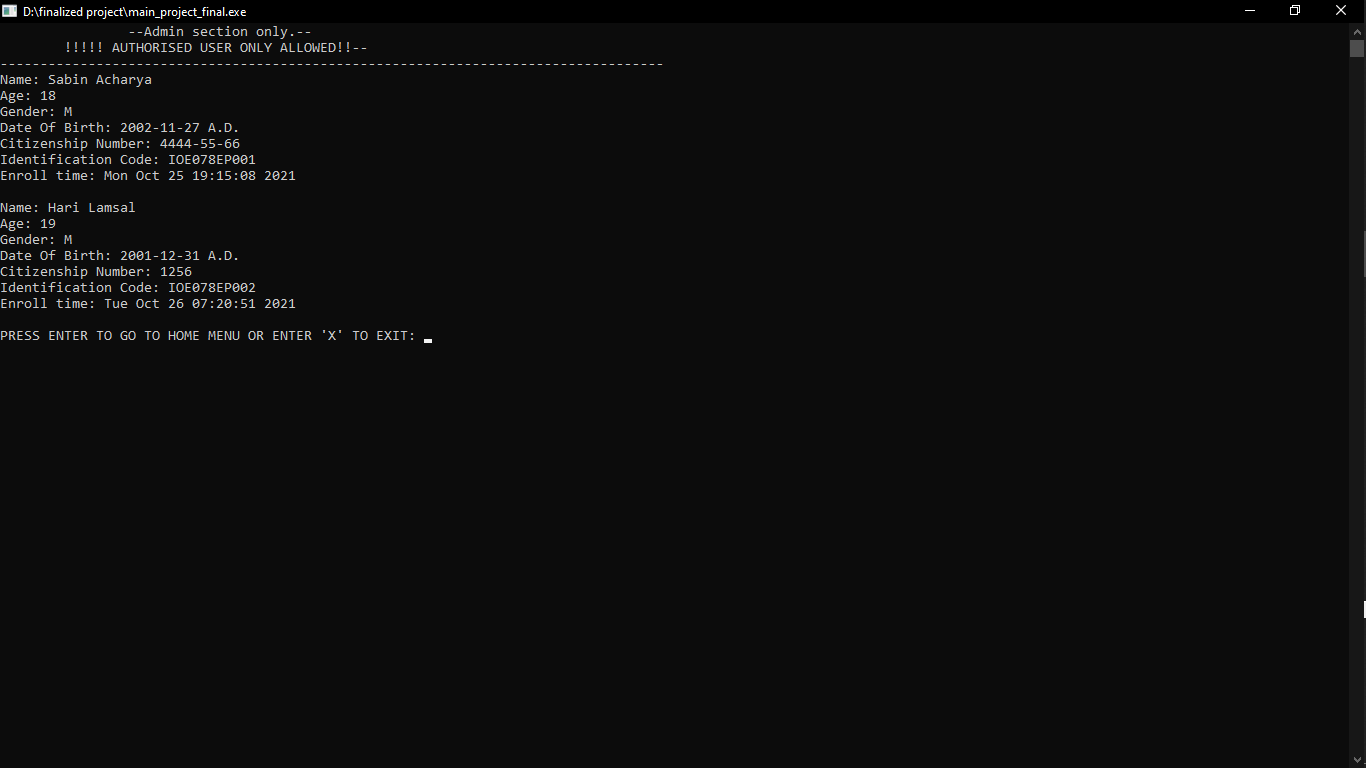
View All Enrolled Students Block



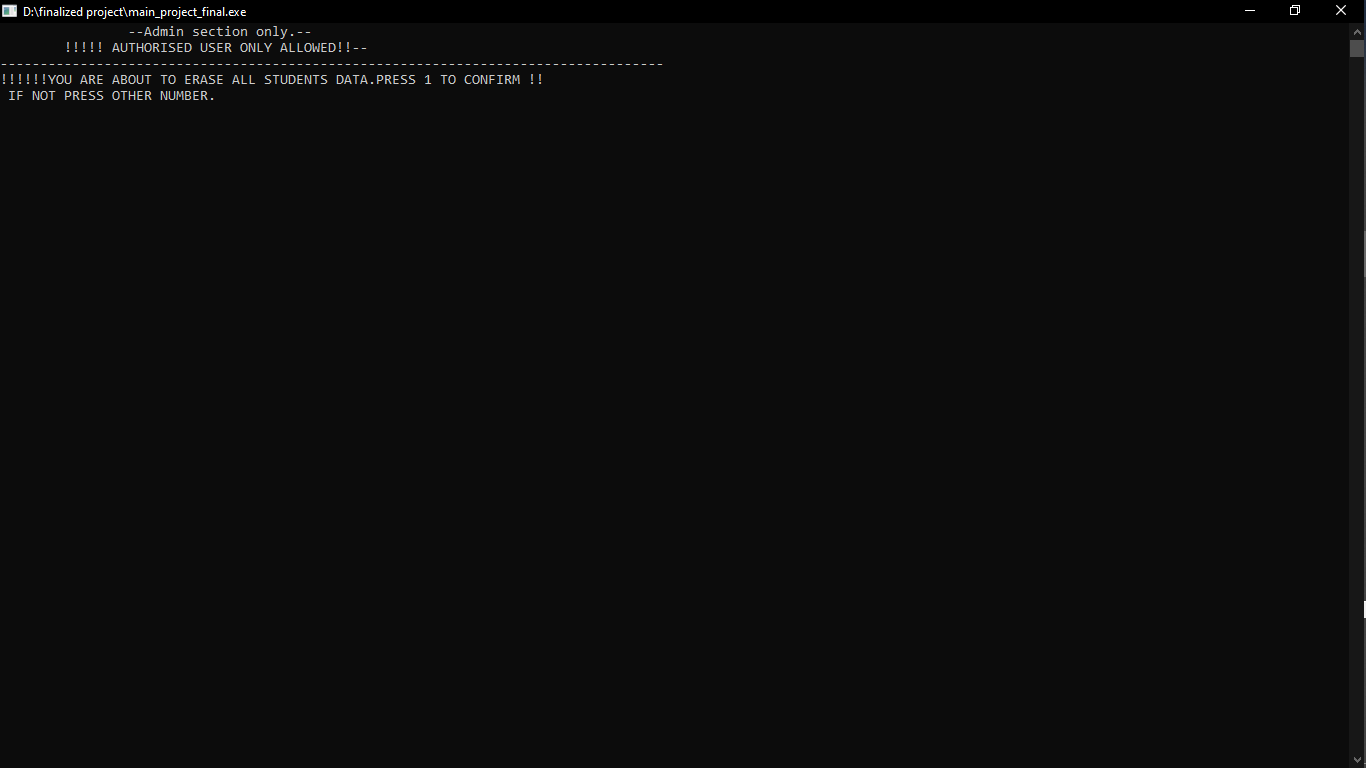


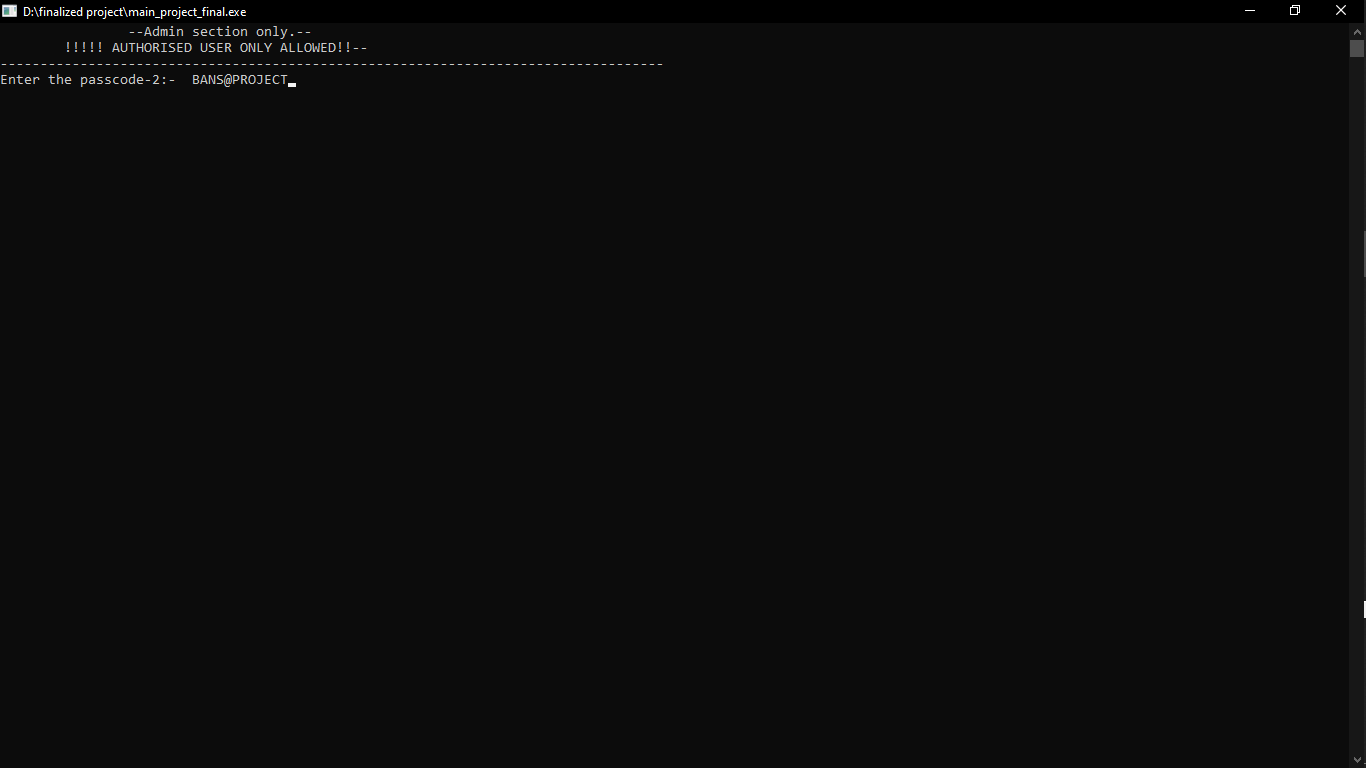
Option-1: View Enrolled Students.

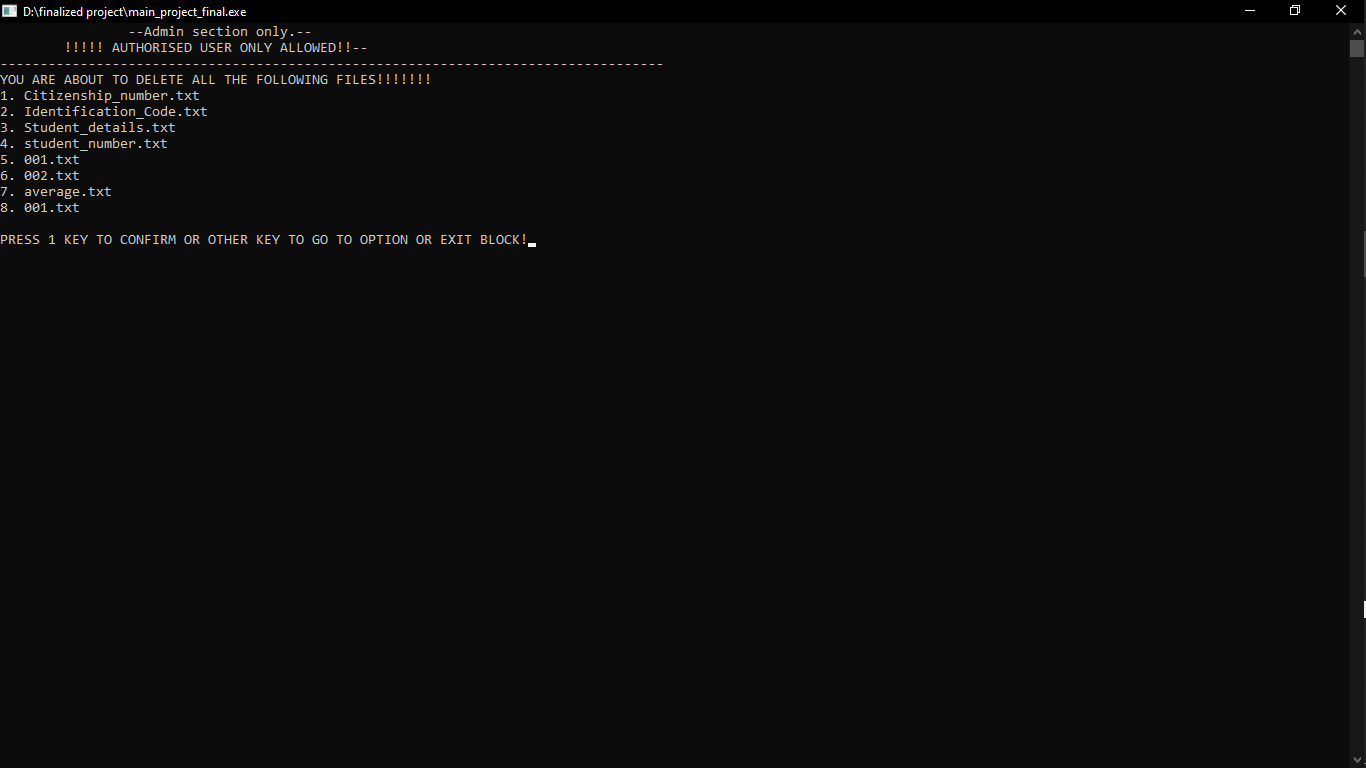


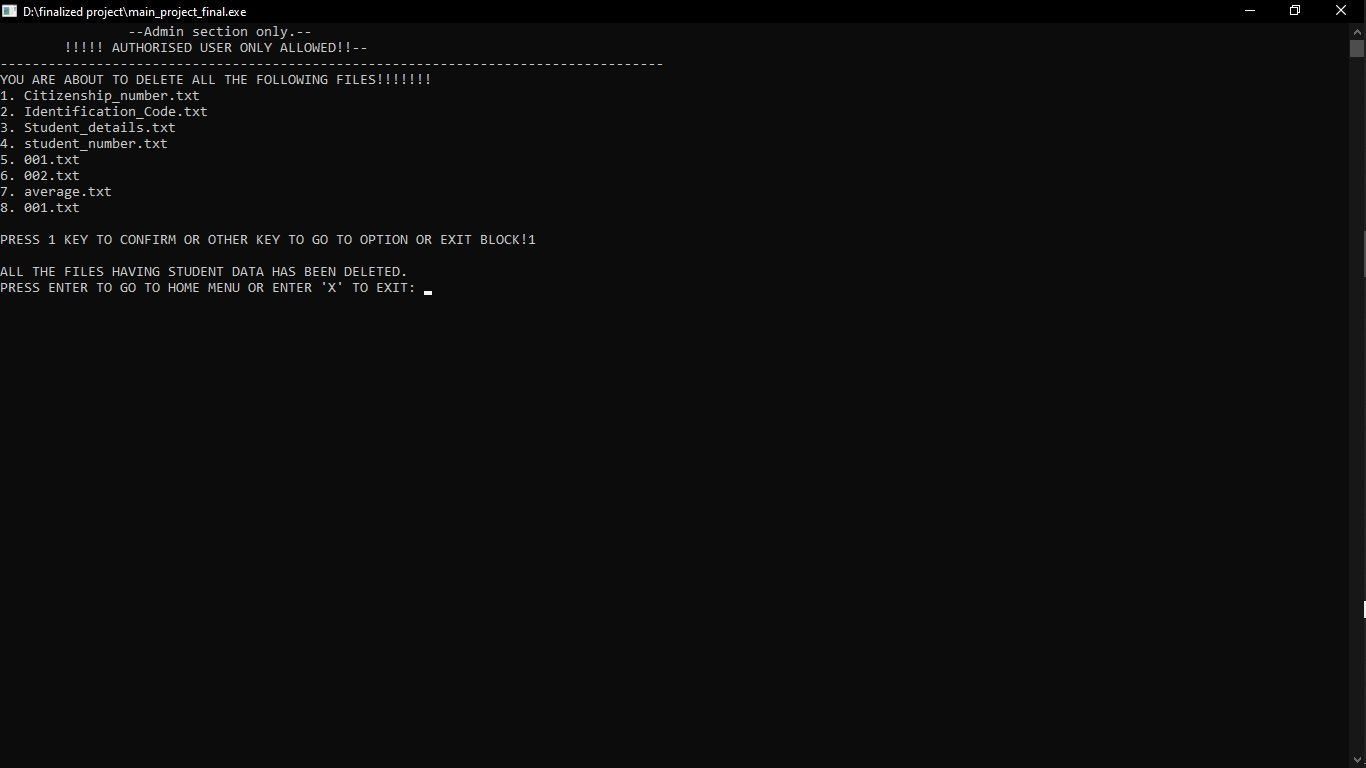


Option-2: View Enrolled Students.









Exit Block

The program exits here.

## 4.2 Result Analysis

The last and most important part of the system is to store and display the overall marks of the user with the number of the exam he has taken in “view overall score block” and rank of user with the total number of users using the system in “view leaderboard block” which a user can access through his identification number and citizenship number and can view all of his data including the number of exams given and the time he\she started and ended the exam.

# CONCLUSION & FURTHER WORK

## Conclusion

Since one can use it to practice for the various engineering entrance, one can go through the various sets of questions to prepare for the entrance along with he\she can take practice set by choosing how many questions he\she wants to do. One can see their progress and also helps them to increase their time management skill, discover their weak areas and work on them. Thus it helps to provide an extra edge to crack the entrance exam.

## 5.2 Further Works

Many experiments, tests have been left for the future because of lack of time(for example adding the multiple numbers of questions sets with the different questions are usually time-consuming). Due to lack of time, we only add a few question and answer sets. We had also intended to make different subject-wise sets so students can work on their weak subjects.

# REFERENCES

1. Tutorialspoint.com (https://www.tutorialspoint.com/index.htm?gclid=CjwKCAjw3riIBhAwEiwAzD3TiR0J0PaMxJWMj6FsQp2wMtml8uvPnT7SWV01C8y7UaxejMtMPCX0uBoCqmoQAvD\_BwE)
2. A Text Book Of C Programming (Author: Ram Dutta Bhatta, Babu Ram Dawadi)
3. “CodeWithHarry” Youtube Channel

(<https://www.youtube.com/channel/UCeVMnSShP_Iviwkknt83cww>)

# APPENDICES

//below code was written using code blocks and vs code.

#include<stdio.h> // To include the standard input output functions present in this library.

#include<conio.h> // To include other input output functions which are not present in stdio.h library.

#include<string.h> // To include functions related to string.

#include<stdlib.h> // To use the functions which take or provide data to a file.

#include<stdbool.h> // To use the boolean datatype in c which is an additional datatype in c.

#include<time.h>

#include<dirent.h>

typedef struct details

{

char name[20];

char gender;

char identification\_code[15];

char ctznshp\_no[20];

} std;

void Initializing\_Display(); // Declaration of Initializing\_Display Function.

void Heading\_Printer(); // Declaration of the Heading Printer Function.

int Home\_Menu(); // Declaration of the Home\_Menu Function.

void Option\_Selector(); // Declaration of Option\_Selector Functions.

void Enroll\_Yourself(); // Declaration of Enroll\_Yourself Function.

void Take\_Practise\_Exam(); // Declaration of Take\_Practise\_Exam Function.

void Take\_Full\_Mock\_Test(); // Declaration of Take\_Full\_Mock\_Test Function.

void View\_Overall\_Score(); // Declaration of View\_Overall\_Score Function.

void View\_Leaderboard(); // Declaration of View\_Leaderboard Function.

void Amend\_Question\_Set(); // Declaration of Amend\_Question\_Set Function.

void View\_Enrolled\_Students(); // Declaration of View\_Enrolled\_Students Function.

void Exit(); // Declaration of Exit Function.

void Option\_Or\_Exit();

void Identification\_Code\_Generator(char id\_string[15]); // Declaration of Identification\_Code\_Generator Function.

bool Citizenship\_Checker(char number[20]); // Declaration of Citizenship\_Checker Function.

bool Identification\_Code\_Checker(char code[15]); // Declaration of Identification\_Code\_Checker Function.

void Student\_Detail\_Printer(int n);

int id\_citizenship\_matcher();

int Mocktest\_QFile\_Lister();

int Practisetest\_QFile\_Lister();

void Mocktest\_AFile\_Lister();

void Newline\_Remover(char string[]);

void Average\_Marks\_Updator(int new\_marks, int student\_number);

int Age\_Finder(int day , int month, int year);

bool Leap\_Year\_Checker(int year);

void diplay\_question(int c, char Qpath[50]);

int answer\_check(char a, int c, char Apath[50]);

void password(char passcode[20]);

void admin();

int students\_scores();

int students\_details();

void Directory\_Maker();

bool folder\_checker( char path[100], char name[20] );

int question\_counter(int set\_no);

int main() // From here, the execution of program begins.

{

Initializing\_Display(); // Function call upon the initialing\_display Function to show the details of program and developers.

Option\_Selector(); // Function call to Option\_Selector Function to provide the options to select the input.

return 0; // The program Execution Ends here.

}

void Initializing\_Display() // This function contains the statements that shows the information about the developers of the project.

{

char continue\_input;

Heading\_Printer();

printf("\nComputer Programming Project:\n\n\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n\t\t|\t\t\t\t\t\t\t\t\t|\n\t\t|\t\t\t\t Project By:\t\t\t\t|\n\t\t|\t\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\t\t\t\t|\n\t\t|\t\t\t\t\t\t\t\t\t|\n\t\t|\t\t\t\t\t\t\t\t\t|\n");

printf("\t\t|\t\t\t1. Anish Timsina(THA077BEI007)\t\t\t|\n\t\t|\t\t\t\t\t\t\t\t\t|\n");

printf("\t\t|\t\t\t2. Bishal Khadka(THA077BEI015)\t\t\t|\n\t\t|\t\t\t\t\t\t\t\t\t|\n");

printf("\t\t|\t\t\t3. Nixon Raj Dhakal(THA077BEI028)\t\t|\n\t\t|\t\t\t\t\t\t\t\t\t|\n");

printf("\t\t|\t\t\t4. Sabin Acharya(THA077BEI035)\t\t\t|\n\t\t|\t\t\t\t\t\t\t\t\t|\n\t\t|\t\t\t\t\t\t\t\t\t|\n");

printf("\t\t| Faculty: Electonics, Communication and Information Engineering.\t|\n\t\t|\t\t\t\t\t\t\t\t\t|\n");

printf("\t\t| Semester: 1st\t\t\t\t\t\t\t\t|\n\t\t|\t\t\t\t\t\t\t\t\t|\n");

printf("\t\t| College: Thapathali Campus.\t\t\t\t\t\t|\n\t\t|\t\t\t\t\t\t\t\t\t|\n");

printf("\t\t| Enrolled Year : 2077 B.S.\t\t\t\t\t\t|\n\t\t|\t\t\t\t\t\t\t\t\t|\n");

printf("\t\t|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\n\n");

Directory\_Maker();

do // This loop is to keep on asking the user to enter "Enter" to continue until the user presses Enter.

{

printf("Press Enter to Continue...\n");

fflush(stdin);

continue\_input = getchar();

} while(continue\_input != '\n');

}

void Heading\_Printer()

{

printf("\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Entrance Preparation Kit For IOE Entrance Preparing students\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\t\t\t\t\t\t\t\t\t2078 B.S./2021 A.D.");

printf("\n\n");

//printing the top heading in the project

}

int Home\_Menu()

{

int loop\_condition\_input;

do

{

system("cls");

Heading\_Printer();

printf("ENTER THE NUMBERS TO CHOOSE THE RESPECTIVE OPTIONS:\n\n\n");

printf("1. Enroll Yourself!\n\n");

printf("2. Take practise Test(You can choose the number of questions.)\n\n");

printf("3. Take Full Mock Test.\n\n");

printf("4. View Your overall Score.\n\n");

printf("5. View the Leader-Board\n\n");

printf("6. Ammend Question Set.(Admins Only)\n\n");

printf("7. View All Enrolled Students.(Admins Only)\n\n");

printf("8. Exit.\n");

fflush(stdin);

scanf("%d", &loop\_condition\_input);

}while(loop\_condition\_input<1 || loop\_condition\_input>8);

if(loop\_condition\_input == 8)

{

exit(1);

}

else

{

return loop\_condition\_input;

}

}

void Option\_Selector()

{

start:

system("cls");

int opt\_select\_input = Home\_Menu();

switch (opt\_select\_input)

{

case 1:

Enroll\_Yourself();

break;

case 2:

Take\_Practise\_Exam();

break;

case 3:

Take\_Full\_Mock\_Test();

break;

case 4:

View\_Overall\_Score();

break;

case 5:

View\_Leaderboard();

break;

case 6:

Amend\_Question\_Set();

break;

case 7:

View\_Enrolled\_Students();

break;

case 8:

Exit();

break;

default:

goto start;

break;

}

}

void Enroll\_Yourself()

{

int day , month, year;

time\_t enroll\_time;

std details; //Making structure to take student details.

FILE \*detail, \*citizenship, \*Identification\_Code, \*Average;

short int age;

bool ct\_checker;

//Below part is taking the personal details from the user.

system("cls");

Heading\_Printer();

fseek(stdin, 0, SEEK\_END);

printf("# Enter you name: ");

scanf("%[^\n]s", details.name);

printf("\n");

fseek(stdin, 0, SEEK\_END);

printf("# Enter your gender(Male = M, Female = F, Others = O): ");

details.gender = getchar();

printf("\n");

DOB\_TAKER:

fseek(stdin, 0, SEEK\_END);

printf("# Enter your date of birth(IN A.D.):\n");

printf("\t# Enter your birth year(Eg: 2002): ");

scanf("%d", &year);

printf("\t# Enter your birth month(Eg: 03): ");

scanf("%d", &month);

printf("\t# Enter your birth day(Eg: 24): ");

scanf("%d", &day);

age = Age\_Finder(day, month, year);

if(age == 0)

{

printf("\nYour DOB Doesn't Seem To Be Valid. Please Enter A Valid DOB.\n");

goto DOB\_TAKER;

}

else if(age<16)

{

printf("\n# Your age is: %d years.\n", age);

printf("YOU'RE UNDER-AGE TO GIVE THE ENTRANCE EXAM. SORRY!\n");

printf("PRESS ANY KEY TO EXIT");

fseek(stdin, 0, SEEK\_END);

char quit = getch();

exit(1);

}

printf("\n# Your age is: %d years.\n", age);

printf("\n");

fseek(stdin, 0, SEEK\_END);

Repeat: //This label is used to retake the citizenship number in case of pre registerd citizenship number.

printf("# Enter your Citizenship Number: ");

memset(details.ctznshp\_no, 0, sizeof(details.ctznshp\_no));

scanf("%[^\n]s", details.ctznshp\_no);

ct\_checker = Citizenship\_Checker(details.ctznshp\_no);

if(ct\_checker == true)

{

printf("\n\nThe citizenship number you enterd is already registered. Please use another unregistered citizenship number.\n\n\n");

fseek(stdin,0, SEEK\_END);

goto Repeat;

}

Identification\_Code\_Generator(details.identification\_code);

printf("\n\nYour identificaton code is : %s \n", details.identification\_code);

enroll\_time = time(NULL);

localtime(&enroll\_time);

detail = fopen("C:\\IOEEPADMK\_DATABASE\\studentinfo\\Student\_details.txt", "a");

fprintf(detail, "Name: %s \n", details.name);

fprintf(detail, "Age: %d\n", age);

fprintf(detail, "Gender: %c\n", details.gender);

fprintf(detail, "Date Of Birth: %d-%d-%d A.D.\n", year,month,day );

fprintf(detail, "Citizenship Number: %s\n", details.ctznshp\_no);

fprintf(detail, "Identification Code: %s\n", details.identification\_code);

fprintf(detail, "Enroll time: %s\n", ctime(&enroll\_time));

fclose(detail);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*This below part is to print the identification code of the enrolled students so that it can be used in other functions.\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

if((Identification\_Code = fopen("C:\\IOEEPADMK\_DATABASE\\studentinfo\\Identification\_Code.txt","r")) == NULL) //This condition is to check for new run of the program.

{

Identification\_Code = fopen("C:\\IOEEPADMK\_DATABASE\\studentinfo\\Identification\_Code.txt","w");

fprintf(Identification\_Code, "%s", details.identification\_code);

fclose(Identification\_Code);

}

else //This part is for the older run of the program.

{

fclose(Identification\_Code);

Identification\_Code = fopen("C:\\IOEEPADMK\_DATABASE\\studentinfo\\Identification\_Code.txt","a");

fprintf(Identification\_Code, "\n%s", details.identification\_code);

fclose(Identification\_Code);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*This below part is to print the citizenship\_number of the enrolled students so that it can be used in other functions.\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

if((citizenship = fopen("C:\\IOEEPADMK\_DATABASE\\studentinfo\\Citizenship\_number.txt","r")) == NULL) //This condition is to check for new run of the program.

{

citizenship = fopen("C:\\IOEEPADMK\_DATABASE\\studentinfo\\Citizenship\_number.txt","w");

fprintf(citizenship, "%s", details.ctznshp\_no);

fclose(citizenship);

}

else //This part is for the older run of the program.

{

fclose(citizenship);

citizenship = fopen("C:\\IOEEPADMK\_DATABASE\\studentinfo\\Citizenship\_number.txt","a");

fprintf(citizenship, "\n%s", details.ctznshp\_no);

fclose(citizenship);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

if((Average = fopen("C:\\IOEEPADMK\_DATABASE\\mocktest\\Scoresheets\\average.txt","r")) == NULL) //This condition is to check for new run of the program.

{

Average = fopen("C:\\IOEEPADMK\_DATABASE\\mocktest\\Scoresheets\\average.txt","w");

fprintf(Average,"00.00 0");

fclose(Average);

}

else //This part is for the older run of the program.

{

fclose(Average);

Average = fopen("C:\\IOEEPADMK\_DATABASE\\mocktest\\Scoresheets\\average.txt","a");

fprintf(Average,"\n00.00 0");

fclose(Average);

}

fflush(stdin);

Option\_Or\_Exit();

}

void Take\_Practise\_Exam()

{

system("cls");

Heading\_Printer();

time\_t start\_time, stop\_time;

int student\_id\_number = id\_citizenship\_matcher();

printf("\n");

int number\_of\_question, a = 0;

int score = 0, time\_start = 0;

char b, st\_time[30], en\_time[30];

char Qpath[100], Apath[100], Scoresheetname[50];

int set\_number\_input;

FILE \*Scoresheet;

int setNumber;

reEnter:

printf("# Enter the Question set number among displayed sets:");

setNumber = Practisetest\_QFile\_Lister();

scanf("%d",&set\_number\_input);

printf("\n");

if(set\_number\_input >= setNumber)

{

printf("\nInvalid Input!!\n");

goto reEnter;

}

sprintf(Qpath, "practiseset\\QuestionSets\\Set-%d.txt", set\_number\_input); //This function is used to make a formatted string.

sprintf(Apath, "practiseset\\AnswerSets\\Set-%d.txt", set\_number\_input);

start:

printf("# Please enter the number of question you want to solve:(Min-Questions: 5, Max-Questions: 20) \n");

printf("\n");

scanf("%d", &number\_of\_question);

if (number\_of\_question >= 5 && number\_of\_question <= 20)

{

system("cls");

Heading\_Printer();

int Qstn\_repeat\_checker[100];

for(int arr\_iterator = 0; arr\_iterator < 100; arr\_iterator++)

{

Qstn\_repeat\_checker[arr\_iterator] = 0;

}

start\_time = time(NULL);

struct tm \*time1 = localtime(&start\_time);

strftime(st\_time, sizeof(st\_time), "%d-%b-%Y %I:%M:%S %p", time1);

for (int i = 0; i < number\_of\_question; i++)

{

loop:

if (a < number\_of\_question)

{

int c;

another:

srand(time(0));

c = rand() % 100 + 1;

time\_t start, end;

bool YesNo = false;

for(int arr\_iterator = 0; arr\_iterator < 100; arr\_iterator++)

{

if(Qstn\_repeat\_checker[arr\_iterator] == c)

{

YesNo = true;

break;

}

}

if(YesNo == true)

{

goto another;

}

else

{

Qstn\_repeat\_checker[i] = c;

}

start = time(NULL);

printf("%d. ", i+1);

diplay\_question(c, Qpath);

while (!kbhit())

{

end = time(NULL);

if (difftime(end, start) >= 90)

{

printf("Times up !!! \nGet ready for next question\n\n");

a++;

i++;

goto loop;

}

}

b = getche();

printf("\n");

answer\_check(b, c, Apath);

if (answer\_check(b, c, Apath) == 1)

{

printf("your answer is correct. \n");

int tiMe = difftime(end, start);

printf("your soved this problem in %d seconds \n\n", tiMe);

score++;

}

else

{

printf("your answer is incorrect! \n\n");

}

a++;

}

}

stop\_time = time(NULL);

struct tm \*time2 = localtime(&stop\_time);

strftime(en\_time, sizeof(en\_time), "%d-%b-%Y %I:%M:%S %p", time2);

sprintf(Scoresheetname, "C:\\IOEEPADMK\_DATABASE\\practiseset\\Scoresheets\\%03d.txt", student\_id\_number);

if((Scoresheet = fopen(Scoresheetname,"r")) == NULL) //This condition is to check for new run of the program.

{

Scoresheet = fopen(Scoresheetname,"w");

fprintf(Scoresheet,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

fprintf(Scoresheet, "|%s\t\t|%s\t\t\t|%s\t\t\t|%s\t\t\t|%s\t|","Set Number", "Start Time", "End Time", "Duration", "Score" );

fprintf(Scoresheet,"\n|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_|");

fprintf(Scoresheet, "\n|Set-%d\t\t\t|%s\t|%s\t|%0.2f Minutes\t\t\t|%03d/%d\t|",set\_number\_input, st\_time, en\_time, (difftime(stop\_time, start\_time))/60, score,number\_of\_question);

fclose(Scoresheet);

}

else //This part is for the older run of the program.

{

fclose(Scoresheet);

Scoresheet = fopen(Scoresheetname,"a");

fprintf(Scoresheet, "\n|Set-%d\t\t\t|%s\t|%s\t|%0.2f Minutes\t\t\t|%03d/%d\t|",set\_number\_input, st\_time, en\_time, (difftime(stop\_time, start\_time))/60, score,number\_of\_question);

fclose(Scoresheet);

}

printf("\nYOUR SCORESHEET:");

printf("\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n|\t\t\t\t\t\t\t|\n");

printf("|Start Time\t: %s\t\t|\n", st\_time);

printf("|End Time\t: %s\t\t|\n", en\_time);

printf("|Duration\t: %0.2f Seconds\t\t\t\t|\n", difftime(stop\_time, start\_time));

printf("|Score\t\t: %03d/%d\t\t\t\t\t|\n", score,number\_of\_question);

printf("|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\n");

}

else

{

printf("Please select the question between 5 and 20 \n\n");

goto start;

}

Option\_Or\_Exit();

}

void Take\_Full\_Mock\_Test()

{

time\_t start\_time, stop\_time, start, end;

char id\_number\_string[3], Qsetname[100], Asetname[100], Scoresheetname[100], Question[500], st\_time[30], en\_time[30];

int id\_number\_integer, set\_number\_input, score = 0;

char Answer, Answer\_input;

FILE \*Qset, \*Aset, \*Scoresheet;

system("cls");

Heading\_Printer();

id\_number\_integer = id\_citizenship\_matcher();

int set\_number;

reEnter:

set\_number = Mocktest\_QFile\_Lister();

printf("Enter the set number: ");

scanf("%d", &set\_number\_input);

if(set\_number\_input >= set\_number)

{

printf("\nInvalid Input!!\n");

goto reEnter;

}

fseek(stdin, 0, SEEK\_END);

sprintf(Qsetname, "mocktest\\QuestionSets\\Set-%d.txt", set\_number\_input); //This function is used to make a formatted string.

sprintf(Asetname, "mocktest\\AnswerSets\\Set-%d.txt", set\_number\_input);

Qset = fopen(Qsetname, "r");

Aset = fopen(Asetname, "r");

start\_time = time(NULL);

struct tm \*time1 = localtime(&start\_time);

strftime(st\_time, sizeof(st\_time), "%d-%b-%Y %I:%M:%S %p", time1);

start = time(NULL);

system("cls");

Heading\_Printer();

for (int b = 0; (b<100 && difftime(end, start)<=7200); b++)

{

end = time(NULL);

printf("\n\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

int tiMe = difftime(end, start);

printf("Time: %d min %d seconds /120 min\n", tiMe/60, (tiMe%60));

for(int a= 0; a<7; a++)

{

fgets(Question, sizeof(Question), Qset);

printf("%s", Question);

}

printf("Enter Your option: ");

while(kbhit() == 0)

{

end = time(NULL);

if((difftime(end, start)) >= 7200)

{

break;

}

}

if(kbhit() != 0)

{

Answer\_input = getche();

Answer = fgetc(Aset);

fgetc(Aset);

if(Answer\_input == Answer)

{

score ++;

}

}

printf("\n\nThe correct answer is: %c\n", Answer);

printf("\nScore: %02d\n", score);

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

}

stop\_time = time(NULL);

struct tm \*time2 = localtime(&stop\_time);

strftime(en\_time, sizeof(en\_time), "%d-%b-%Y %I:%M:%S %p", time2);

fclose(Qset);

fclose(Aset);

sprintf(Scoresheetname, "C:\\IOEEPADMK\_DATABASE\\mocktest\\Scoresheets\\%03d.txt", id\_number\_integer);

Newline\_Remover(st\_time);

Newline\_Remover(en\_time);

if((Scoresheet = fopen(Scoresheetname,"r")) == NULL) //This condition is to check for new run of the program.

{

Scoresheet = fopen(Scoresheetname,"w");

fprintf(Scoresheet,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

fprintf(Scoresheet, "|%s\t\t|%s\t\t\t|%s\t\t\t|%s\t\t\t|%s\t|","Set Number", "Start Time", "End Time", "Duration", "Score" );

fprintf(Scoresheet,"\n|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_|");

fprintf(Scoresheet, "\n|Set-%d\t\t\t|%s\t|%s\t|%0.2f Minutes\t\t\t|%03d\t|",set\_number\_input, st\_time, en\_time, (difftime(stop\_time, start\_time))/60, score);

fclose(Scoresheet);

}

else //This part is for the older run of the program.

{

fclose(Scoresheet);

Scoresheet = fopen(Scoresheetname,"a");

fprintf(Scoresheet, "\n|Set-%d\t\t\t|%s\t|%s\t|%0.2f Minutes\t\t\t|%03d\t|",set\_number\_input, st\_time, en\_time, (difftime(stop\_time, start\_time))/60, score);

fclose(Scoresheet);

}

Average\_Marks\_Updator(score, id\_number\_integer);

//Marksheet Printing\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

printf("\nYOUR SCORESHEET:");

printf("\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n|\t\t\t\t\t\t\t|\n");

printf("|Start Time\t: %s\t\t|\n", st\_time);

printf("|End Time\t: %s\t\t|\n", en\_time);

printf("|Duration\t: %0.2f Seconds\t\t\t\t|\n", difftime(stop\_time, start\_time));

printf("|Score\t\t: %03d\t\t\t\t\t|\n", score);

printf("|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\n");

fflush(stdin);

Option\_Or\_Exit();

}

void View\_Overall\_Score()

{

system("cls");

Heading\_Printer();

int file\_no;

char pathmock[100],pathpractise[100],c,s;

FILE \*score\_mocktest,\*score\_practisetest;

file\_no = id\_citizenship\_matcher();

sprintf(pathmock,"C:\\IOEEPADMK\_DATABASE\\mocktest\\Scoresheets\\%03d.txt",file\_no);

score\_mocktest=fopen(pathmock,"r");

sprintf(pathpractise,"C:\\IOEEPADMK\_DATABASE\\practiseset\\Scoresheets\\%03d.txt",file\_no);

score\_practisetest=fopen(pathpractise,"r");

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

printf("\t\tScore in mock test\n");

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

if(score\_mocktest == NULL)

{

printf("File is empty.");

}

while((c=fgetc(score\_mocktest)) != EOF) //reading data of student performance stored in file

{

printf("%c",c);

}

fflush(stdin);

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

printf("\t\tScore in practise test\n");

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

while((s=fgetc(score\_practisetest))!=EOF)

{

printf("%c",s);

}

fflush(stdin);

printf("\n\n");

fclose(score\_mocktest);

fclose(score\_practisetest);

Option\_Or\_Exit();

}

void View\_Leaderboard()

{

system("cls");

Heading\_Printer();

printf("LEADERBOARD:\n\n");

FILE \*averagemarks, \*ptr;

char num,line\_counter\_array[10];

int number\_of\_data=0;

averagemarks=fopen("C:\\IOEEPADMK\_DATABASE\\mocktest\\Scoresheets\\average.txt","r");

while((num=fgetc(averagemarks))!=EOF){

fgets(line\_counter\_array,sizeof(line\_counter\_array),averagemarks);

number\_of\_data+=1;

}

fclose(averagemarks);

float average\_scores[number\_of\_data];

int number\_of\_exams[number\_of\_data];

averagemarks=fopen("C:\\IOEEPADMK\_DATABASE\\mocktest\\Scoresheets\\average.txt","r");

for(int a = 0; a<number\_of\_data; a++)

{

fscanf(averagemarks, "%f", &average\_scores[a]);

fscanf(averagemarks, "%d", &number\_of\_exams[a]);

}

char ids[number\_of\_data][15];

ptr = fopen("C:\\IOEEPADMK\_DATABASE\\studentinfo\\Identification\_Code.txt", "r");

for(int a = 0; a<number\_of\_data; a++)

{

fgets(ids[a], sizeof(ids[a]), ptr);

Newline\_Remover(ids[a]);

}

fclose(averagemarks);

fclose(ptr);

for(int a = 0; a<number\_of\_data; a++)

{

for(int b = a+1; b<number\_of\_data; b++)

{

if(average\_scores[a]<average\_scores[b])

{

float swapper = average\_scores[b];

average\_scores[b] = average\_scores[a];

average\_scores[a] = swapper;

char id\_swapper[15];

strcpy(id\_swapper, ids[a]);

strcpy(ids[a], ids[b]);

strcpy(ids[b], id\_swapper);

int swapper1 = number\_of\_exams[b];

number\_of\_exams[b] = number\_of\_exams[a];

number\_of\_exams[a] = swapper1;

}

}

}

printf("\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("|Rank\t|\tIdenitification Number\t\t|\tAverage Scores\t\t|\tExam Number\t|\n");

printf("|\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\n");

for(int a = 0; a<number\_of\_data; a++)

{

printf("|%d\t|\t%s\t\t\t|\t%0.2f\t\t\t|\t%d\t\t|\n", a+1, ids[a], average\_scores[a], number\_of\_exams[a]);

}

printf("|\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\n\n");

Option\_Or\_Exit();

}

void Amend\_Question\_Set()

{

int input,set\_no,Qnum = 0,check;

char Qsetname[100],Question[500],Aset[100],Asetname[500],new\_question[500],option1[50],option2[50],option3[50],option4[50];

char answer,character;

fflush(stdin);

char passcode[20] = "project@c";

admin();

password(passcode);

new\_set:

printf("\nselect the option\n");

printf("1.Add question to existing set.\n");

printf("2.Add new question set.\n");

scanf("%d",&input);

while(input<1 || input>2){ //loop continues if input is wrong

printf("!!!!! Wrong Input !!!!!!!\n");

printf("Re - enter :-- ");

fflush(stdin);

scanf("%d",&input);

}

admin();

if(input==1)

{

ammend:

fflush(stdin);

Mocktest\_QFile\_Lister();

printf("\n Input the set you want to ammend = ");

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

fseek(stdin, 0,SEEK\_END);

FILE\* Qset;

Qnum=question\_counter(set\_no);

if(Qnum>=100){

printf("The question set is full please select another set\n");

Qnum=0;

fflush(stdin);

goto ammend;

}

fseek(stdin, 0,SEEK\_END);

FILE\* Aset;

sprintf(Qsetname, "mocktest\\QuestionSets\\Set-%d.txt", set\_no);

sprintf(Asetname, "mocktest\\AnswerSets\\Set-%d.txt", set\_no);

Question: //calling again from here if user want to add more questions

Qset=fopen(Qsetname,"a");

Aset=fopen(Asetname,"a");

Qnum+=1;

fseek(stdin, 0,SEEK\_END);

printf("Question:-");

scanf("%[^\n]s",new\_question);

fprintf(Qset,"%d. %s",Qnum,new\_question);

fseek(stdin, 0,SEEK\_END);

printf("option a:- ");

scanf("%[^\n]s",option1);

fprintf(Qset,"\n\n a.%s",option1);

fseek(stdin, 0,SEEK\_END);

printf("option b:- ");

scanf("%[^\n]s",option2);

fprintf(Qset,"\n b.%s",option2);

fseek(stdin, 0,SEEK\_END);

printf("option c:- ");

scanf("%[^\n]s",option3);

fprintf(Qset,"\n c.%s",option3);

fseek(stdin, 0,SEEK\_END);

printf("option d:- ");

scanf("%[^\n]s",option4);

fprintf(Qset,"\n d.%s\n\n",option4);

fseek(stdin, 0,SEEK\_END);

printf("Answer:- ");

scanf("%c",&answer);

fprintf(Aset,"%c\n",answer);

fseek(stdin, 0,SEEK\_END);

fclose(Qset);

fclose(Aset);

printf("\nDo you want to add more: 1 for yes and other number for no. ");

scanf("%d",&check);

Qnum=question\_counter(set\_no);

if(Qnum>=100){

printf("The question set is full please select another set\n");

Qnum=0;

fflush(stdin);

goto ammend;

}

if (check==1)

goto Question;

else

{

Option\_Or\_Exit();

}

}

if(input==2){

int set;

FILE\* Qset;

FILE\* Aset;

set=Mocktest\_QFile\_Lister();

system("cls");

Heading\_Printer();

printf("Set-%d:\n\n", set);

sprintf(Qsetname, "mocktest\\QuestionSets\\Set-%d.txt", set);

sprintf(Asetname, "mocktest\\AnswerSets\\Set-%d.txt", set);

Question\_new:

Qset=fopen(Qsetname,"a"); //opening new set for input of question

Aset=fopen(Asetname,"a"); //opening new set for answer input

Qnum=Qnum+1;

fseek(stdin, 0,SEEK\_END);

printf("Question:-");

scanf("%[^\n]s",&new\_question);

fprintf(Qset,"%d. %s",Qnum,new\_question);

fseek(stdin, 0,SEEK\_END);

printf("option a:- ");

scanf("%[^\n]s",&option1);

fprintf(Qset,"\n\n a.%s",option1);

fseek(stdin, 0,SEEK\_END);

printf("option b:- ");

scanf("%[^\n]s",&option2);

fprintf(Qset,"\n b.%s",option2);

fseek(stdin, 0,SEEK\_END);

printf("option c:- ");

scanf("%[^\n]s",&option3);

fprintf(Qset,"\n c.%s",option3);

fseek(stdin, 0,SEEK\_END);

printf("option d:- ");

scanf("%[^\n]s",&option4);

fprintf(Qset,"\n d.%s\n\n",option4);

fseek(stdin, 0,SEEK\_END);

printf("Answer:- ");

scanf("%[^\n]s",&answer);

fprintf(Aset,"%c\n",answer);

fseek(stdin, 0,SEEK\_END);

fclose(Qset);

fclose(Aset);

printf("\nDo you want to add more: 1 for yes and other key for no. ");

scanf("%d",&check);

Qnum=question\_counter(set);

if(Qnum>=100){

printf("The question set is full please make another set\n");

Qnum=0;

fflush(stdin);

goto new\_set;

}

if (check==1)

goto Question\_new;

else

{

Option\_Or\_Exit();

}

}

}

void View\_Enrolled\_Students()

{

char c;

int input,check;

char passcode1[20]="project@c", passcode2[20] = "BANS@PROJECT";

password(passcode1);

admin();

printf("Select option:-\n ");

printf("\t 1.view students details\n");

printf("\t 2.RESET the file of students details for keeping new record.\n");

scanf("%d",&input);

while(input<1 || input>2){ //calling loop again and again if input is wrong

printf("!!!!! Wrong Input !!!!!!!\n");

printf("Re - enter :-- ");

fflush(stdin);

scanf("%d",&input);

}

admin();

if(input==1)

{

fflush(stdin);

admin();

password(passcode2);

FILE \*fptr;

fptr=fopen("C:\\IOEEPADMK\_DATABASE\\studentinfo\\student\_details.txt","r");

while((c=fgetc(fptr))!=EOF){

printf("%c",c);

}

fclose(fptr);

Option\_Or\_Exit();

}

if(input==2){

int i;

printf("!!!!!!YOU ARE ABOUT TO ERASE ALL STUDENTS DATA.PRESS 1 TO CONFIRM !!\n IF NOT PRESS OTHER NUMBER. ");

scanf("%d",&i);

if(i==1){

fflush(stdin);

password(passcode2);

admin();

printf("YOU ARE ABOUT TO DELETE ALL THE FOLLOWING FILES!!!!!!!\n");

students\_scores();

printf("\nPRESS 1 KEY TO CONFIRM OR OTHER KEY TO GO TO OPTION OR EXIT BLOCK!");

scanf("%d",&check);

if (check==1)

{

DIR \*theFolder = opendir("C:\\IOEEPADMK\_DATABASE\\studentinfo");

struct dirent \*next\_file;

char filepath[250];

while ( (next\_file = readdir(theFolder)) != NULL )

{

// build the path for each file in the folder

sprintf(filepath, "C:\\IOEEPADMK\_DATABASE\\studentinfo\\%s", next\_file->d\_name);//comes the name of files serially

remove(filepath); //delet the respective files from the folder of studentinfo

}

closedir(theFolder);

DIR \*Folder\_scorecard = opendir("C:\\IOEEPADMK\_DATABASE\\mocktest\\Scoresheets");

struct dirent \*score\_file;

char filelocation[250];

while ( (score\_file = readdir(Folder\_scorecard)) != NULL )

{

// build the path for each file in the folder

sprintf(filelocation, "C:\\IOEEPADMK\_DATABASE\\mocktest\\Scoresheets\\%s", score\_file->d\_name);

remove(filelocation);

}

closedir(Folder\_scorecard);

DIR \*Folder\_practicescore = opendir("C:\\IOEEPADMK\_DATABASE\\practiseset\\Scoresheets");

struct dirent \*practice\_score;

char fileset[250];

while ( (practice\_score = readdir(Folder\_practicescore)) != NULL )

{

// build the path for each file in the folder

sprintf(fileset, "C:\\IOEEPADMK\_DATABASE\\practiseset\\Scoresheets\\%s", practice\_score->d\_name);

remove(fileset);

}

closedir(Folder\_practicescore);

printf("\nALL THE FILES HAVING STUDENT DATA HAS BEEN DELETED.\n");

Option\_Or\_Exit();

}

else

{

Option\_Or\_Exit();

}

}

else

{

Option\_Or\_Exit();

}

}

}

void Exit()

{

exit(1);

}

void Option\_Or\_Exit()

{

char input;

Top:

printf("PRESS ENTER TO GO TO HOME MENU OR ENTER 'X' TO EXIT: ");

fflush(stdin);

scanf("%c", &input);

if( input == 'x' || input == 'X' || input == '\n')

{

if(input == '\n')

{

Option\_Selector();

}

else

{

Exit();

}

}

else

{

goto Top;

}

}

void Identification\_Code\_Generator(char id\_string[15])

{

FILE \*number;

short int numb;

char end\_checker;

//This below part is just to make the code number appear automatically for the users.

number = fopen("C:\\IOEEPADMK\_DATABASE\\studentinfo\\student\_number.txt", "r");

if(number == NULL)

{

fclose(number);

number = fopen("C:\\IOEEPADMK\_DATABASE\\studentinfo\\student\_number.txt", "w");

fprintf(number, "001");

fclose(number);

number = fopen("C:\\IOEEPADMK\_DATABASE\\studentinfo\\student\_number.txt", "r");

}

do

{

fscanf(number, "%d", &numb);

} while((end\_checker = fgetc(number)) != EOF);

fclose(number);

memset(id\_string,0, 15);

sprintf(id\_string, "IOE078EP%03d", numb);

numb++;

number = fopen("C:\\IOEEPADMK\_DATABASE\\studentinfo\\student\_number.txt", "a");

fprintf(number, "\n%03d", numb);

fclose(number);

}

bool Citizenship\_Checker(char number[20])

{

FILE \*ptr = fopen("C:\\IOEEPADMK\_DATABASE\\studentinfo\\Citizenship\_number.txt", "r"); //Opering the file Citizenship\_number.txt in read mode to read the numbers in the file.

char number\_from\_file[20]; //This string stores the citizenship\_number from file.

char checker; //This character is used to identify the End Of File.

if(ptr == NULL) //This is for the first run of the program. If there is no file named Citizenship\_number.txt, the function returns false.

{

fclose(ptr);

return false;

}

else

{

while((checker = fgetc(ptr)) != EOF)

{

fseek(ptr, -1, SEEK\_CUR);

fgets(number\_from\_file, sizeof(number\_from\_file), ptr);

for(int a = 0; number\_from\_file[a] != '\0'; a++) //This loop is to remove the '\n' that comes along with the string at its end when using fgets function.

{

if(number\_from\_file[a] == '\n') //if '\n' is identified, this block is executed.

{

number\_from\_file[a] = '\0'; //'\n'is replaced by '\0'.

}

}

if(strcmp(number\_from\_file, number) == 0) //If the citizenship number from file is equal to the citizenship number from the function, it returns true.

{

fclose(ptr);

return true;

}

}

}

fclose(ptr);

return false; //If no matching Citizenship number is found, the function returns false.

}

bool Identification\_Code\_Checker(char code[15])

{

FILE \*ptr = fopen("C:\\IOEEPADMK\_DATABASE\\studentinfo\\Identification\_Code.txt", "r"); //Opering the file Identification\_Code.txt in read mode to read the numbers in the file.

char code\_from\_file[20]; //This string stores the citizenship\_number from file.

char checker; //This character is used to identify the End Of File.

if(ptr == NULL) //This is for the first run of the program. If there is no file named Identification\_Code.txt, the function returns false.

{

fclose(ptr);

return false;

}

else

{

while((checker = fgetc(ptr)) != EOF)

{

fseek(ptr, -1, SEEK\_CUR);

fgets(code\_from\_file, sizeof(code\_from\_file), ptr);

for(int a = 0; code\_from\_file[a] != '\0'; a++) //This loop is to remove the '\n' that comes along with the string at its end when using fgets function.

{

if(code\_from\_file[a] == '\n') //if '\n' is identified, this block is executed.

{

code\_from\_file[a] = '\0'; //'\n'is replaced by '\0'.

}

}

if(strcmp(code\_from\_file, code) == 0) //If the citizenship number from file is equal to the Identification code from the function, it returns true.

{

fclose(ptr);

return true;

}

}

}

fclose(ptr); //If no matching Identification Code is found, the function returns false.

return false;

}

void Student\_Detail\_Printer(int n)

{

char det[50];

FILE \*ptr;

ptr = fopen("C:\\IOEEPADMK\_DATABASE\\studentinfo\\Student\_details.txt", "r");

for(int a = 1; a<=n; a++)

{

if(a != n)

{

for(int b = 0; b<8; b++)

{

fgets(det, sizeof(det), ptr);

}

}

else if(a == n)

{

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

for(int b = 0; b<8; b++)

{

fgets(det, sizeof(det), ptr);

printf("\t\t\t\t\t\t\t%s\n", det);

}

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

}

}

fclose(ptr);

}

int Mocktest\_QFile\_Lister()

{

int set\_number = 1;

DIR \*Directory;

printf("Choose among the following Question Sets: \n");

Directory = opendir("mocktest\\QuestionSets");

struct dirent \*lister;

lister =readdir(Directory);

lister =readdir(Directory);

while((lister = readdir(Directory)) != NULL)

{

printf("%d. %s\n",set\_number, lister->d\_name);

set\_number++;

}

closedir(Directory);

return set\_number;

}

int Practisetest\_QFile\_Lister()

{

int set\_number = 1;

DIR \*Directory;

printf("Choose among the following Question Sets: \n");

Directory = opendir("practiseset\\QuestionSets");

struct dirent \*lister;

lister =readdir(Directory);

lister =readdir(Directory);

while((lister = readdir(Directory)) != NULL)

{

printf("%d. %s\n",set\_number, lister->d\_name);

set\_number++;

}

closedir(Directory);

return set\_number;

}

void Mocktest\_AFile\_Lister()

{

int set\_number = 1;

DIR \*Directory;

printf("Choose among the following Question Sets: \n");

Directory = opendir("practiseset\\QuestionSets");

struct dirent \*lister;

lister =readdir(Directory);

lister =readdir(Directory);

while((lister = readdir(Directory)) != NULL)

{

printf("%d. %s\n",set\_number, lister->d\_name);

set\_number++;

}

closedir(Directory);

}

int id\_citizenship\_matcher()

{

char id\_code[15], id\_number\_string[3], ct\_no\_input[20], ct\_no\_file[20];

int id\_number\_integer;

bool checker;

FILE \*Citizenship\_Number;

Repeat: //This label is used to retake the identification code in case of not registerd identification code.

printf("Enter Your Identification Code: ");

memset(id\_code, 0, sizeof(id\_code));

scanf("%s", id\_code);

printf("\n\n");

checker = Identification\_Code\_Checker(id\_code);

if(checker == false)

{

printf("The identification code you enterd is not registered. Please use another registered Identification code.\n");

fseek(stdin,0, SEEK\_END);

goto Repeat;

}

for(int a = 0; a<3; a++)

{

id\_number\_string[a] = id\_code[8+a];

}

id\_number\_integer = atoi(id\_number\_string);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*This part matches the respective identification number with the person's citizenship number to avoid trespassing\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Take\_Citizenship:

printf("Enter Your Citizenship Number For Confirmation: ");

fseek(stdin, 0, SEEK\_END);

scanf("%s", ct\_no\_input);

printf("\n\n");

Citizenship\_Number = fopen("C:\\IOEEPADMK\_DATABASE\\studentinfo\\Citizenship\_number.txt", "r");

for(int a = 0; a<id\_number\_integer; a++)

{

fgets(ct\_no\_file, sizeof(ct\_no\_file), Citizenship\_Number);

}

fclose(Citizenship\_Number);

for(int a = 0; ct\_no\_file[a] != '\0'; a++) //This loop is to remove the '\n' that comes along with the string at its end when using fgets function.

{

if(ct\_no\_file[a] == '\n') //if '\n' is identified, this block is executed.

{

ct\_no\_file[a] = '\0'; //'\n'is replaced by '\0'.

}

}

if(strcmp(ct\_no\_input, ct\_no\_file) != 0)

{

printf("The citizenship number you entered doest match with your identification number.\n");

goto Take\_Citizenship;

}

else

{

system("cls");

Heading\_Printer();

Student\_Detail\_Printer(id\_number\_integer);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

return id\_number\_integer;

}

void Newline\_Remover(char string[])

{

for(int a = 0; string[a] != '\0'; a++) //This loop is to remove the '\n' that comes along with the string at its end when using fgets function.

{

if(string[a] == '\n') //if '\n' is identified, this block is executed.

{

string[a] = '\0'; //'\n'is replaced by '\0'.

}

}

}

void Average\_Marks\_Updator(int new\_marks, int student\_number)

{

float marks;

int exam\_times;

char end\_checker;

FILE \*ptr1, \*ptr2 = fopen("C:\\IOEEPADMK\_DATABASE\\mocktest\\Scoresheets\\averageupdate.txt", "w");

ptr1 = fopen("C:\\IOEEPADMK\_DATABASE\\mocktest\\Scoresheets\\average.txt", "r");

for(int a = 1; (end\_checker = fgetc(ptr1)) != EOF; a++)

{

fseek(ptr1, -1, SEEK\_CUR);

fscanf(ptr1, "%f", &marks);

printf("%0.2f\n", marks);

fscanf(ptr1, "%d", &exam\_times);

if(a == student\_number)

{

marks = ((marks\*exam\_times)+new\_marks)/(++exam\_times);

}

fprintf(ptr2, "\n%0.2f %d", marks, exam\_times );

}

fclose(ptr1);

fclose(ptr2);

ptr2 = fopen("C:\\IOEEPADMK\_DATABASE\\mocktest\\Scoresheets\\averageupdate.txt", "r");

ptr1 = fopen("C:\\IOEEPADMK\_DATABASE\\mocktest\\Scoresheets\\average.txt", "w");

end\_checker = fgetc(ptr2);

while((end\_checker = fgetc(ptr2)) != EOF)

{

fputc(end\_checker, ptr1);

}

fclose(ptr1);

fclose(ptr2);

remove("C:\\IOEEPADMK\_DATABASE\\mocktest\\Scoresheets\\averageupdate.txt");

}

int Age\_Finder(int day , int month, int year)

{

time\_t today;

time(&today);

struct tm \*today\_time = localtime(&today);

if(year > today\_time->tm\_year+1900 || (year == today\_time->tm\_year+1900 && month > (today\_time->tm\_mon+1)) || (year == today\_time->tm\_year+1900 && month == (today\_time->tm\_mon+1)) && day > today\_time->tm\_mday) //if the entered date is of later date than today's date, it is displayed to be invalid.

{

return 0;

}

else

{

if(month<=12 && month>0) //checking if the month is valid. i.e between 1 to 12. this if statement is executed only if the month number is 1-12.

{

if(Leap\_Year\_Checker(year) == true) //if the year is leap year, the validity of the february date is upto 29 feb which is checked here.

{

if(month == 2 && (day<1 || day>29)) //if date in february in leap year is less than one or greater than 29, the date is displayed to be invalid.

{

return 0;

}

}

else //if the year is not a leap year, the validity of the february date is upto 28 feb which is checked here.

{

if(month == 2 && (day<1 || day>28)) //if date in february in not leap year is less than one or greater than 28, the date is displayed to be invalid.

{

return 0;

}

}

if(month == 1 || month == 3 || month == 5 || month == 7 || month == 8 || month == 10 || month == 12) //if the months are January, March, May, July, August, October and December they have 31 days, so date more than that is considered as invalid.

{

if(day>31 || day<1)

{

return 0;

}

}

else // If the months are April, June, September and November, they have 30 days and date more than that is considered as invalid.

{

if(day<1 || day>30)

{

return 0;

}

}

}

else //if the month number is less than or equal to 0 or greater than 12, the date is shown as invalid.

{

return 0;

}

year = (today\_time->tm\_year + 1900) - year; // if the date is valid, the year difference is calculated.

if(month>(today\_time->tm\_mon+1)) //if the entered month is greater than the current month, the person has not completed that year in age. so 1 is subtracted from the calculated year.

{

year--;

return year;

}

else if(month == (today\_time->tm\_mon+1)) // if the month is equal to the current month, day value is checked.

{

if(day>(today\_time->tm\_mday)) //if the entered day is greater than the current day of month, the person has not completed that year in age. so 1 is subtracted from the calculated year.

{

year --;

return year;

}

else //if the entered day is less than or equal to the current day of month, year value is returned directly.

{

return year;

}

}

else // if the month is less than the current month, the year value is returned directly.

{

return year;

}

}

}

bool Leap\_Year\_Checker(int year)

{

if(year % 4 == 0)

{

if(year % 100 == 0)

{

if(year % 400 == 0)

{

return true;

}

else

{

return false;

}

}

else

{

return true;

}

}

else

{

return false;

}

}

void diplay\_question(int c, char Qpath[50])

{

char b[1000];

FILE \*ptr;

ptr = fopen(Qpath, "r");

for (int v = 0; v < c; v++)

{

if (v < c - 1)

{

for (int i = 0; i < 7; i++)

{

fgets(b, sizeof(b), ptr);

}

}

else

{

for (int j = 0; j < 7; j++)

{

fgets(b, sizeof(b), ptr);

printf("%s", b);

}

}

}

fclose;

}

int answer\_check(char a, int c, char Apath[50])

{

char q;

FILE \*ptr1;

ptr1 = fopen(Apath, "r");

for (int b = 1; b <= c; b++)

{

if (b != c)

{

q = fgetc(ptr1);

fgetc(ptr1);

}

if (b == c)

{

q = fgetc(ptr1);

fgetc(ptr1);

}

fclose;

}

if (a == q)

{

return 1;

}

else

{

return 0;

}

}

void password(char passcode[20]){

int check;

char pass\_code[20];

admin();

fflush(stdin);

int passcode\_number = 0;

if( strcmp(passcode, "project@c") == 0)

{

passcode\_number = 1;

}

else if(strcmp(passcode, "BANS@PROJECT") == 0)

{

passcode\_number = 2;

}

printf("Enter the passcode-%d:-\t", passcode\_number);

scanf("%[^\n]s",&pass\_code);

check=strcmp(pass\_code,passcode);

while(check!=0){

printf("!!!!! Wrong passcode !!!!!!!\n");

printf("Re - enter passcode :-- ");

fflush(stdin);

scanf("%[^\n]s",&pass\_code);

check=strcmp(pass\_code,passcode);

}

while(check==0)

return;

}

void admin(){

system("cls");

printf("\t\t--Admin section only.--\n\t!!!!! AUTHORISED USER ONLY ALLOWED!!--\n");

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

}

int students\_scores()

{

int j;

j=students\_details();

DIR \*students;

students = opendir("C:\\IOEEPADMK\_DATABASE\\mocktest\\scoresheets");

struct dirent \*std;

std =readdir(students);

std =readdir(students);

while((std= readdir(students)) != NULL)

{

printf("%d. %s\n",j, std->d\_name);

j++;

}

closedir(students);

students = opendir("C:\\IOEEPADMK\_DATABASE\\practiseset\\scoresheets");

std =readdir(students);

std =readdir(students);

while((std= readdir(students)) != NULL)

{

printf("%d. %s\n",j, std->d\_name);

j++;

}

closedir(students);

return j;

}

int students\_details()

{

int i=1;

DIR \*students;

students = opendir("C:\\IOEEPADMK\_DATABASE\\studentinfo");

struct dirent \*std;

std =readdir(students);

std =readdir(students);

while((std= readdir(students)) != NULL)

{

printf("%d. %s\n",i, std->d\_name);

i++;

}

closedir(students);

return i;

}

bool folder\_checker( char path[100], char name[20] )

{

DIR \*FOLDER;

bool repeat\_checker;

FOLDER = opendir(path);

struct dirent \*folder\_reader;

folder\_reader = readdir(FOLDER);

if((folder\_reader = readdir(FOLDER)) == NULL)

{

return false;

}

while(folder\_reader != NULL)

{

if(strcmp(folder\_reader->d\_name, name) == 0)

{

repeat\_checker = true;

break;

}

else

{

repeat\_checker = false;

folder\_reader = readdir(FOLDER);

}

}

return repeat\_checker;

}

void Directory\_Maker()

{

bool repeat\_checker = folder\_checker("C:\\", "IOEEPADMK\_DATABASE");

if(repeat\_checker == false)

{

mkdir("C:\\IOEEPADMK\_DATABASE");

}

repeat\_checker = folder\_checker("C:\\IOEEPADMK\_DATABASE", "mocktest");

if(repeat\_checker == false)

{

mkdir("C:\\IOEEPADMK\_DATABASE\\mocktest");

mkdir("C:\\IOEEPADMK\_DATABASE\\mocktest\\Scoresheets");

}

repeat\_checker = folder\_checker("C:\\IOEEPADMK\_DATABASE", "practiseset");

if(repeat\_checker == false)

{

mkdir("C:\\IOEEPADMK\_DATABASE\\practiseset");

mkdir("C:\\IOEEPADMK\_DATABASE\\practiseset\\Scoresheets");

}

repeat\_checker = folder\_checker("C:\\IOEEPADMK\_DATABASE", "studentinfo");

if(repeat\_checker == false)

{

mkdir("C:\\IOEEPADMK\_DATABASE\\studentinfo");

}

}

//used to count the existing number of question in set

int question\_counter(int set\_no){

int Qnum=0,Lines=0;

char Qsetname[100],character;

FILE\* Qset;

sprintf(Qsetname, "mocktest\\QuestionSets\\Set-%d.txt", set\_no);

Qset=fopen(Qsetname,"r");

if (Qset == NULL)

{

printf("The file doesn't exist ! Please check again .");

Exit();

}

while((character=fgetc(Qset))!=EOF) //calculating the existing number of question in given set

{

if(character=='\n')

{

Lines+=1;

if(Lines==7)

{

Qnum+=1;

Lines=0;

}

}

}

fclose(Qset);

return Qnum;

}