**CODIAC: SELF-LEARNING AND QUIZ ENVIRONMENT**

*A*

*Mini Project Report*

*Submitted in partial fulfilment of the Requirements for the award of the Degree of*

**BACHELOR OF ENGINEERING**

IN

**INFORMATION TECHNOLOGY**

By

DEEPAK BHARDWAJ - 1602-19-737-129

NOMULA SAI KIRAN – 1602-19-737-159



**Department of Information Technology Vasavi College of Engineering (Autonomous) (Affiliated to Osmania University) Ibrahimbagh, Hyderabad-31**

**2020**

**Vasavi College of Engineering (Autonomous) (Affiliated to Osmania University)**

**2020**

**Hyderabad-500 031 Department of Information Technology**

**DECLARATION BY THE CANDIDATE**

We, DEEPAK BHARDWAJ and NOMULA SAI KIRAN bearing hall ticket numbers, 1602-19-737-129 and 1602-19-737-159 respectively, hereby declare that the project report entitled “CRIMINAL RECORD MANAGEMENT” is submitted in partial fulfilment of the requirement for the award of the degree of Bachelor of Engineering in Information Technology.

This is a record of bonafide work carried out by us and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

DEEPAK BHARDWAJ

1602-19-737-179

NOMULA SAI KIRAN

1602-19-737-159

(Faculty In-Charge) (Head, Dept. of IT)

**AKNOWLEDGEMENTS**

Our Mini Project would not have been successful without the help of several people. We are extremely thankful to our college, Vasavi College of Engineering, for providing the opportunity to implement our project, “CRIMINAL RECORD MANAGEMENT”.

We would like to express our gratitude to Ms. Divya Lingineni, Assistant Professor, Department of Information Technology, Vasavi College of Engineering and Dr. Ramesh Vasappanavara, Professor, Department of Information Technology, Vasavi College of Engineering, for their esteemed guidance, moral support and invaluable advice provided by them for the success of the Mini Project.

Sincerely,

**NOMULA SAIKIRAN 1602-19-737-159**

**DEEPAK BHARDWAJ 1602-19-737-129**

**ABSTRACT**

The Criminal Record Management system is primarily focused on storing data related to police officers , prisoners and people who file complaints .This Management system has features like registering the compliant ,deleting the compliant ,registering as police official and status of compliant. ID and password are provide for police officials, public and Higher Officials. The data can be viewed like compliant data, officials data and thief data .

**TABLE OF CONTENTS**

1. [Introduction 7](#_bookmark0)
   1. [About The Project 7](#_bookmark1)
   2. [Project Domain 7](#_bookmark2)
      1. [Technical Domain 7](#_bookmark3)
      2. [Functional Domain 7](#_bookmark4)
   3. [Features 8](#_bookmark5)
      1. [Admin 8](#_bookmark6)
      2. [Police Official 8](#_bookmark7)
      3. [User 8](#_bookmark8)
2. [Technology 10](#_bookmark10)
   1. [Software Requirements 10](#_bookmark11)
   2. [Hardware Requirements 10](#_bookmark12)
3. [Proposed Work 11](#_bookmark13)
   1. [Design 11](#_bookmark14)
      1. [User 11](#_bookmark15)
      2. [Admin 13](#_bookmark16)
      3. Police Official
   2. [Implementation 14](#_bookmark17)
      1. [Module-Wise Code 15](#_bookmark18)
      2. [Github/Folder Structure 39](#_bookmark19)
   3. [Testing 40](#_bookmark20)
      1. [Test Plan 40](#_bookmark21)
      2. [User Test Cases 40](#_bookmark22)
      3. [Admin Test Cases 45](#_bookmark23)
4. [Results 49](#_bookmark24)
   1. [User Test Case Results 49](#_bookmark25)
   2. [Admin Test Case Results 57](#_bookmark26)
5. [Additional Knowledge Acquired 62](#_bookmark27)
6. [Conclusion And Future Work 63](#_bookmark28)
7. [References 64](#_bookmark29)

# INTRODUCTION

## ABOUT THE PROJECT

“CRIMINAL RECORD MANAGEMENT SYSTEM” is a console-based C Project which acts as a database for police officials to solve cases registered ,add compliant s and view thief s data. The main moto behind this project is provide a better a easy way of storing and viewing data .

## PROJECT DOMAIN

The domain of the project is the targeted subject area of a computer program. It is a term most commonly used in software engineering. Formally, it represents the target subject of a specific programming project, whether narrowly or broadly defined. To be concise, a **domain** in the realm of software engineering commonly refers to the subject area on which the application is intended to apply. Within a domain, there are two categories:

* + 1. Technical Domain
    2. Functional Domain

### TECHNICAL DOMAIN

This project falls under the domain of a Console Application. A console application is a program designed to be used via a text-only computer interface, such as a text terminal, the command line interface of some operating systems or the text- based interface included with most GUI (Graphical User Interface) operating systems.

### FUNCTIONAL DOMAIN

CRIMINAL RECORD MANAGEMENT SYSTEM comes under the Storing and Viewing domain, because we aimed to produce a platform to users and police officials.

## FEATURES

The Vision of The Criminal Record Management system is primarily focused on storing data related to police officers , prisoners and people who file complaints .This Management system has features like registering the compliant ,deleting the compliant ,registering as police official and status of compliant. ID and password are provided for police officials, public and Higher Officials. The data can be viewed like compliant data, officials data and thief data .

### ADMIN

The Admin is the one who can add or delete a police official and can see records of police official. The Admin can login if he/she has the required credentials. The system first verifies whether the credentials entered match the Admin credentials. If they do, then the Admin gets complete access CRMS. If they don’t match, the required message is displayed and fix it.

### POLICE OFFICIAL

The police can add or remove compliant and update the status of compliant. The Police official can login if he/she has the required credentials. The system first verifies whether the credentials entered match the Police official credentials. If they do, then the police gets complete access CRMS. If they don’t match, the required message is displayed and fix it.

### USER

The user can Register a compliant ,withdraw a compliant, check status of compliant . The User can login if he/she has the required credentials. The system first verifies whether the credentials entered match the Usercredentials. If they do, then the User gets complete access CRMS. If they don’t match, the required message is displayed and fix it.

# TECHNOLOGY

All computer software needs certain hardware components or other software resources to be present, in order for computers to be used efficiently. These prerequisites are known as System Requirements. Within this, we have two types – Software Requirements and Hardware Requirements.

## SOFTWARE REQUIREMENTS

Software Requirements deal with defining the software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. These preconditions are generally not included in the software installation package and need to be installed separately.

In order to use CODIAC, one should have the following:

* **Operating System:** Windows 7 and above
* **C Compiler:** GNU Compiler Collection (GCC)
* **Editor:** Any text editor (preferably Visual Studio Code)

## HARDWARE REQUIREMENTS

Hardware requirements refer to the common set requirements defined by any operating system or software application and are usually the physical computer resources. In this, we look into the architecture, processing power, memory, secondary memory, display adapter and peripherals.

In order to use CODIAC, one should have the following:

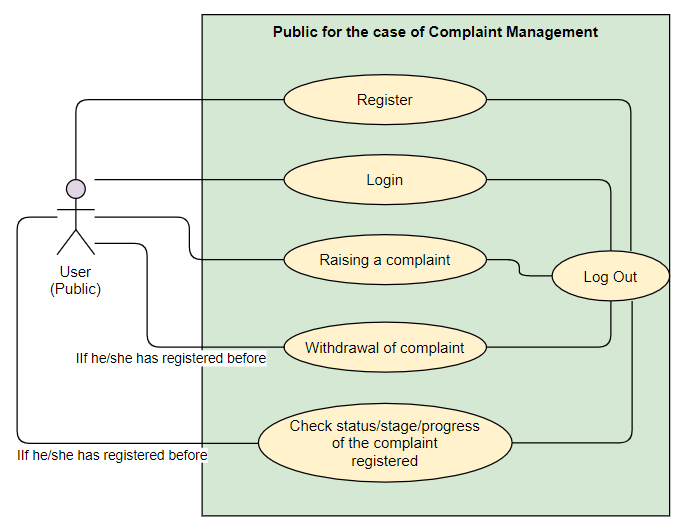
* + - **Processor:** Intel Core i3 and above
    - **Memory:** 8 GB RAM
  1. **DESIGN**

# PROPOSED WORK

Our approach in designing CRMS was to divide our users into three groups – the users, police official and the Admin. The user is a day-to-day consumer whose only goal is to check status or register a compliant . The Admin’s goal is to check police official work and add or delete police official. The police official job is to view a compliant, add or remove compliant and update status of compliant.

* + 1. **USER USE CASES**

The user is a person who has limited access to CRMS. He/she will have 3 functionalities: Register Compliant,Withdraw and status.



* + - 1. **REGISTER**

Public can register as a user in CRMS . First, the system prompts the user to enter his/her details. Once, the user enters the required details (first name, last name, username and password), the system checks whether the username checks or not and whether the password meets the prerequisites. If the username exists, the system displays a message to choose another username. If the password is not valid, it will display the respective message of what to fix.

**3.1.1.1 WITHDRAW**

User can withdraw a compliant by specified reason and compliant will be deleted .

* + - 1. **STATUS**

Once you login as a user status option is used to check status of compliant whether it is solved or not by entering user id which is used to search and if its found it will show status as solved or not else it will show complain not found.

* + - 1. **REGISTER COMPLIANT**

After Registering his/her details user get to register his complaint in form of a letter and stores in a form of text document.

* + 1. **ADMIN**

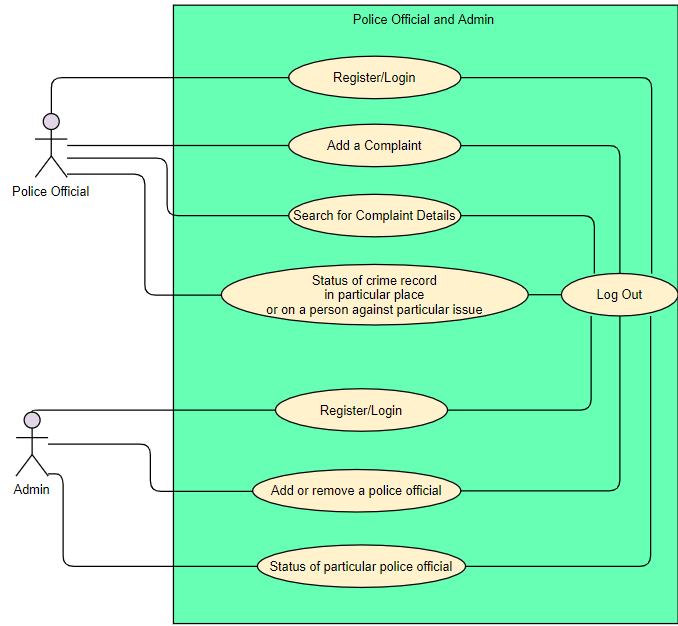
The Admin is the higher official who keeps eye on records of police and he can add or delete a police official.

* + - 1. **LOGIN**

The Admin can login if he/she has the required credentials. The system first verifies whether the credentials entered match the Admin credentials. If they do, then the Admin gets complete access CRMS. If they don’t match, the required message is displayed and fix it.

* + - 1. **VIEW POLICE DETAILS**

After the Admin logs in, he/she can view all the information of all registered police official .By using their registered ID. The system first verifies whether the credentials entered match the police credentials. If they do, then the Admin gets complete access CRMS. If they don’t match, the required message is displayed and fix it.



* + - 1. **DELETE OR ADD POLICE**

Only the Admin has the authority to delete a police official. The Admin may have different reasons for deleting a police official. He as an authority to add police official also.

* + 1. **POLICE OFFICIAL**

The Police official has charge to add a compliant and remove a compliant ,view a compliant . The system first verifies whether the credentials entered match the police credentials. If they do, then the Admin gets complete access CRMS. If they don’t match, the required message is displayed and fix it.

**3.1.3.1 ADD A COMPLIANT**

The police official can add a compliant by entering details of and returns to main menu.

**3.1.3.2 DELETE A COMPLIANT**

The police official can delete the compliant by entering compliant details if you enter wrong data , the error message will pop up and u should fix it.

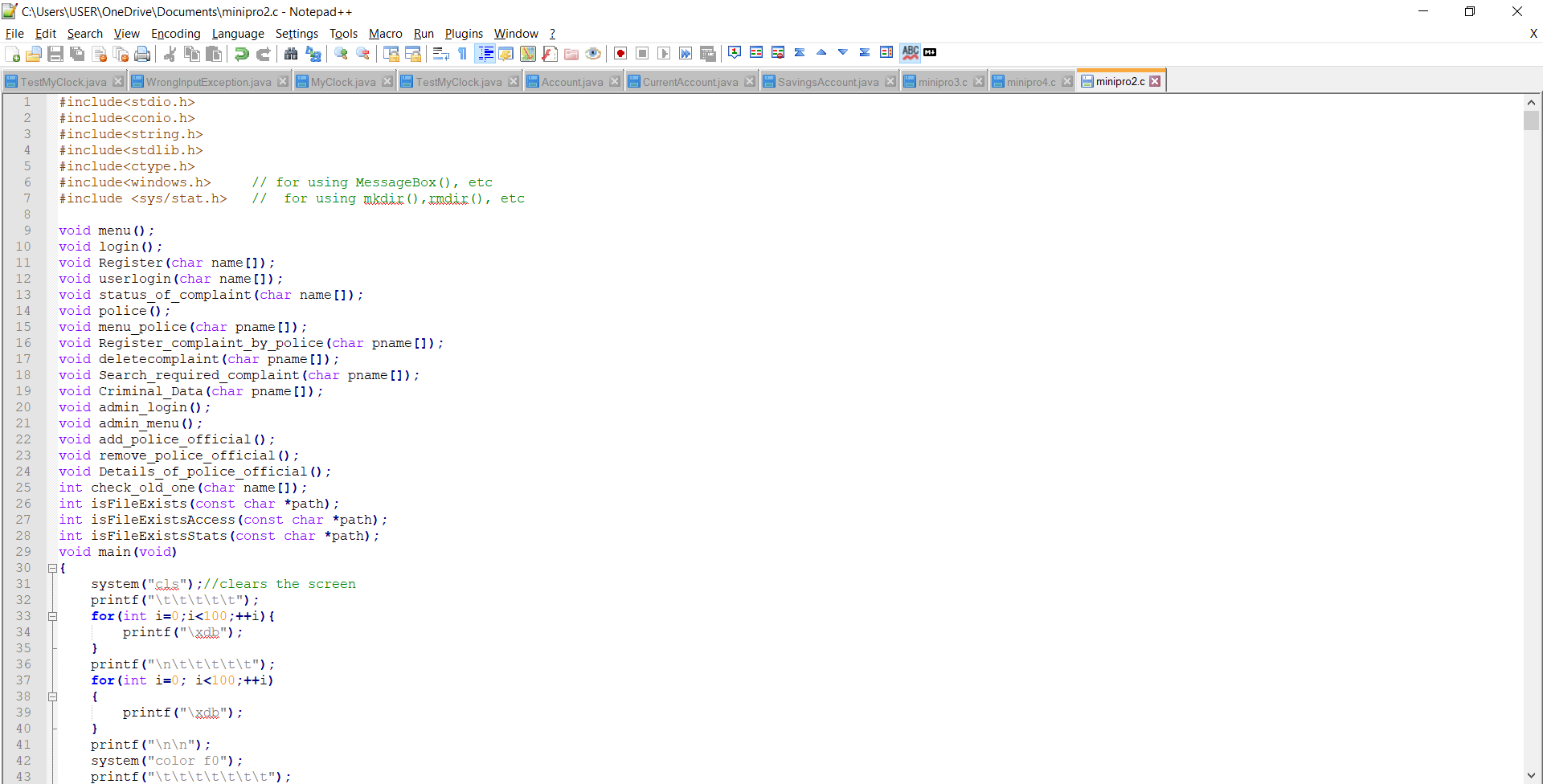
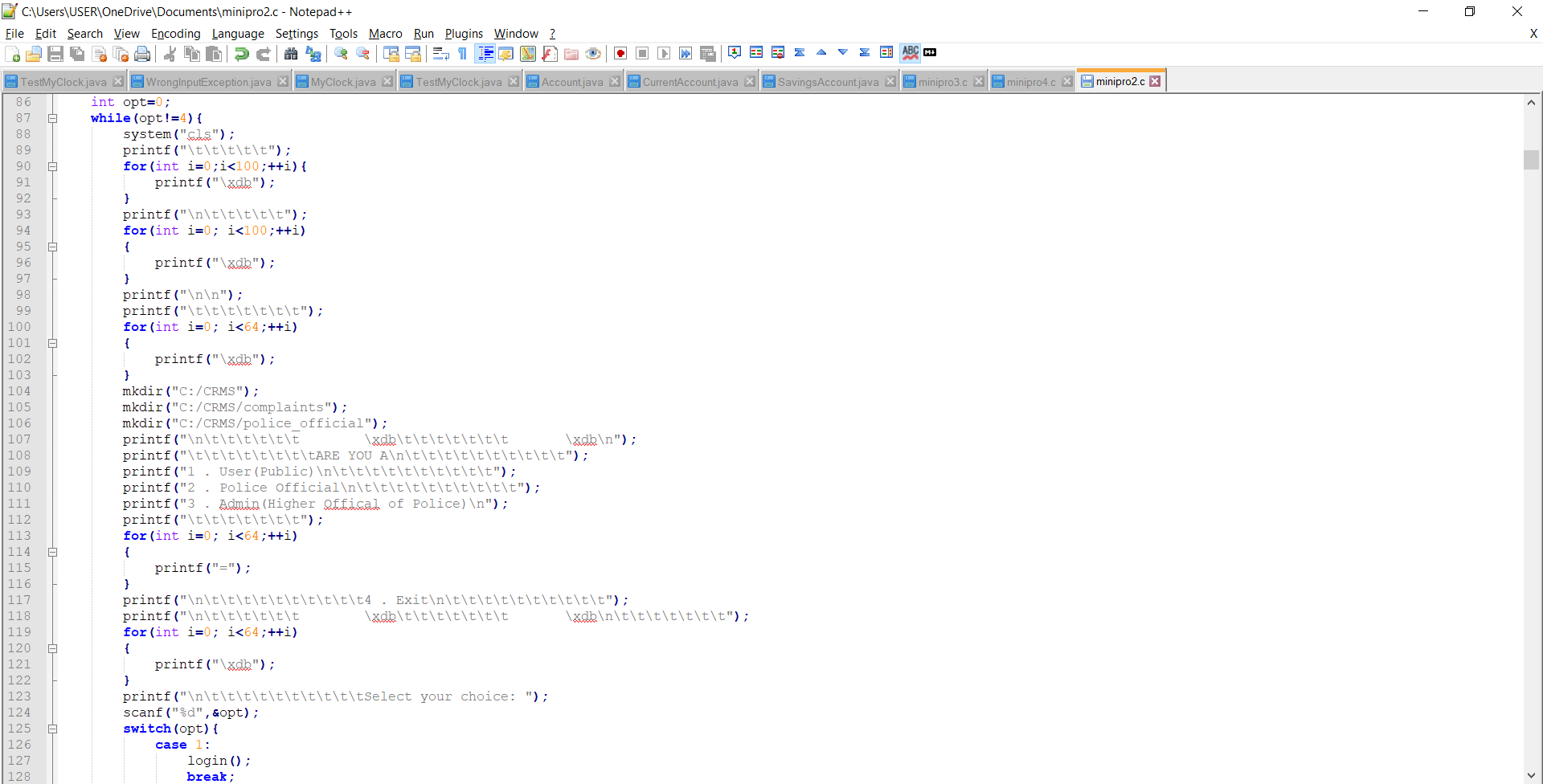
**3.1.3.3 STATUS OF COMPLIANT**

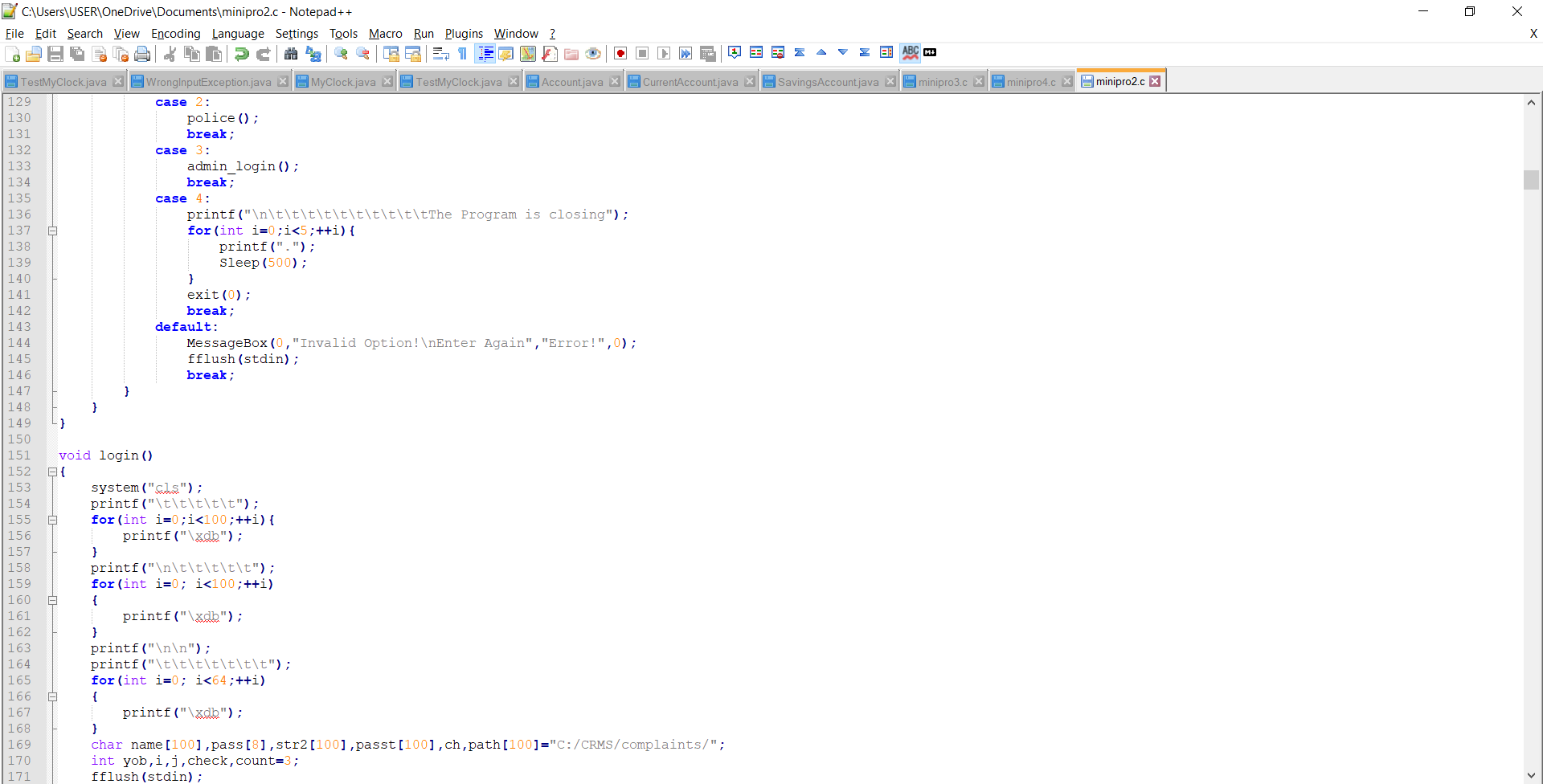
The police official can change the status of compliant selecting the option called view compliant and update status. The police official can access complaint by entering ID of compliant.

## IMPLEMENTATION

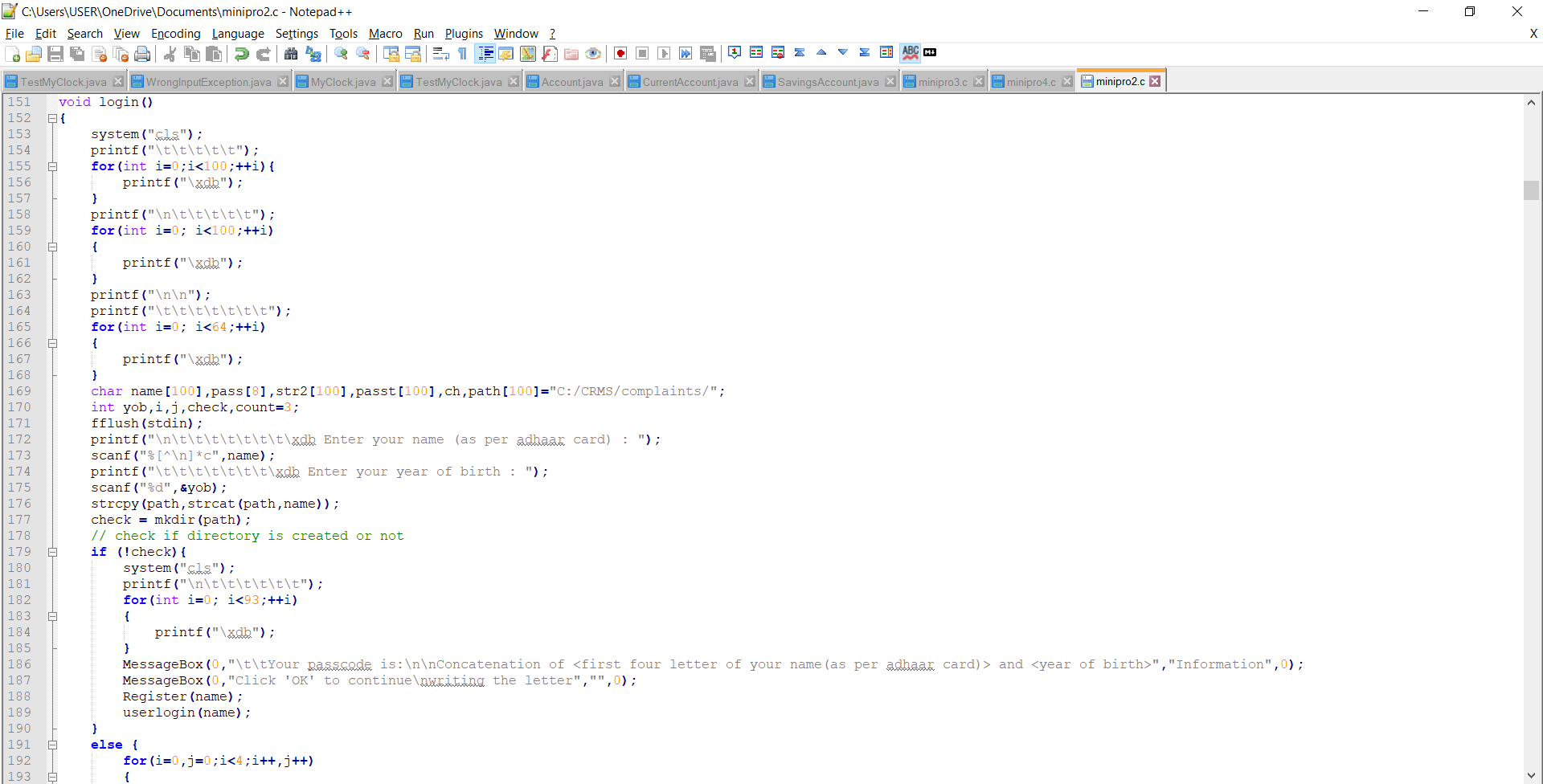
Based on the use cases, we have implemented this project by dividing the work into 6 modules – Register and Login , Admin ,Student Police Official and Banner. The Banner module is for stylistically printing our product’s name when we run our console application.

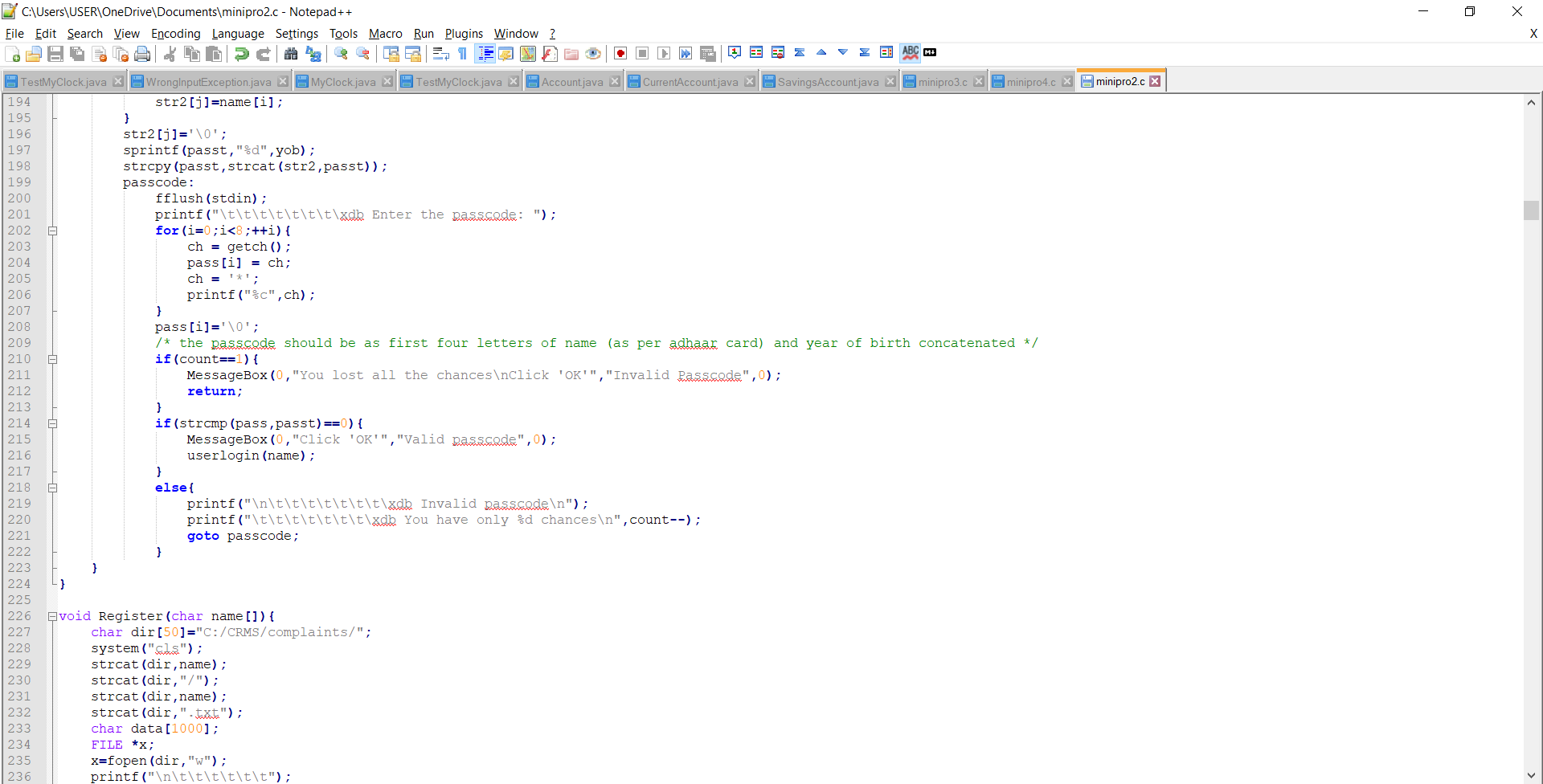
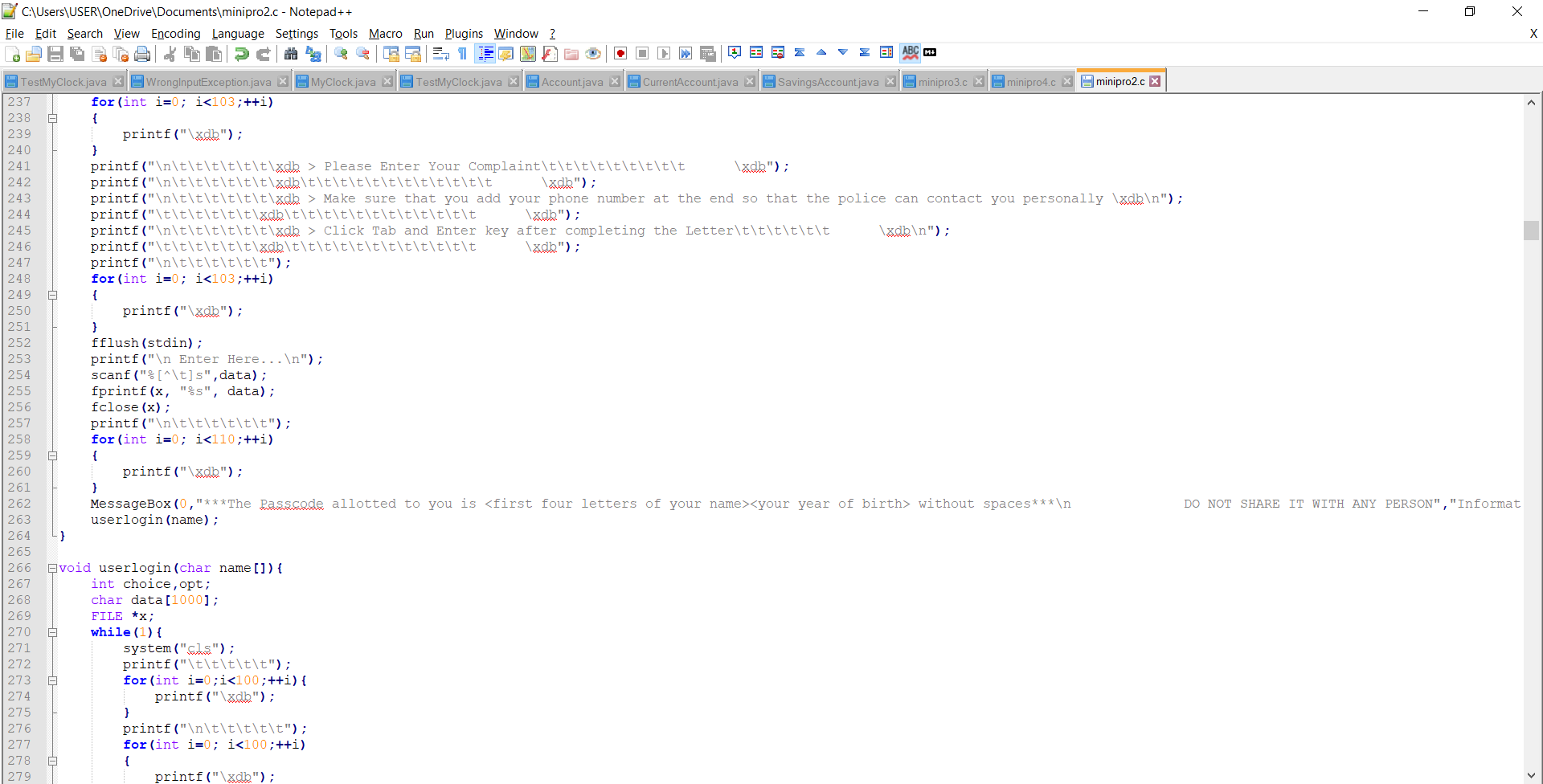
* + 1. **MODULE-WISE CODE**

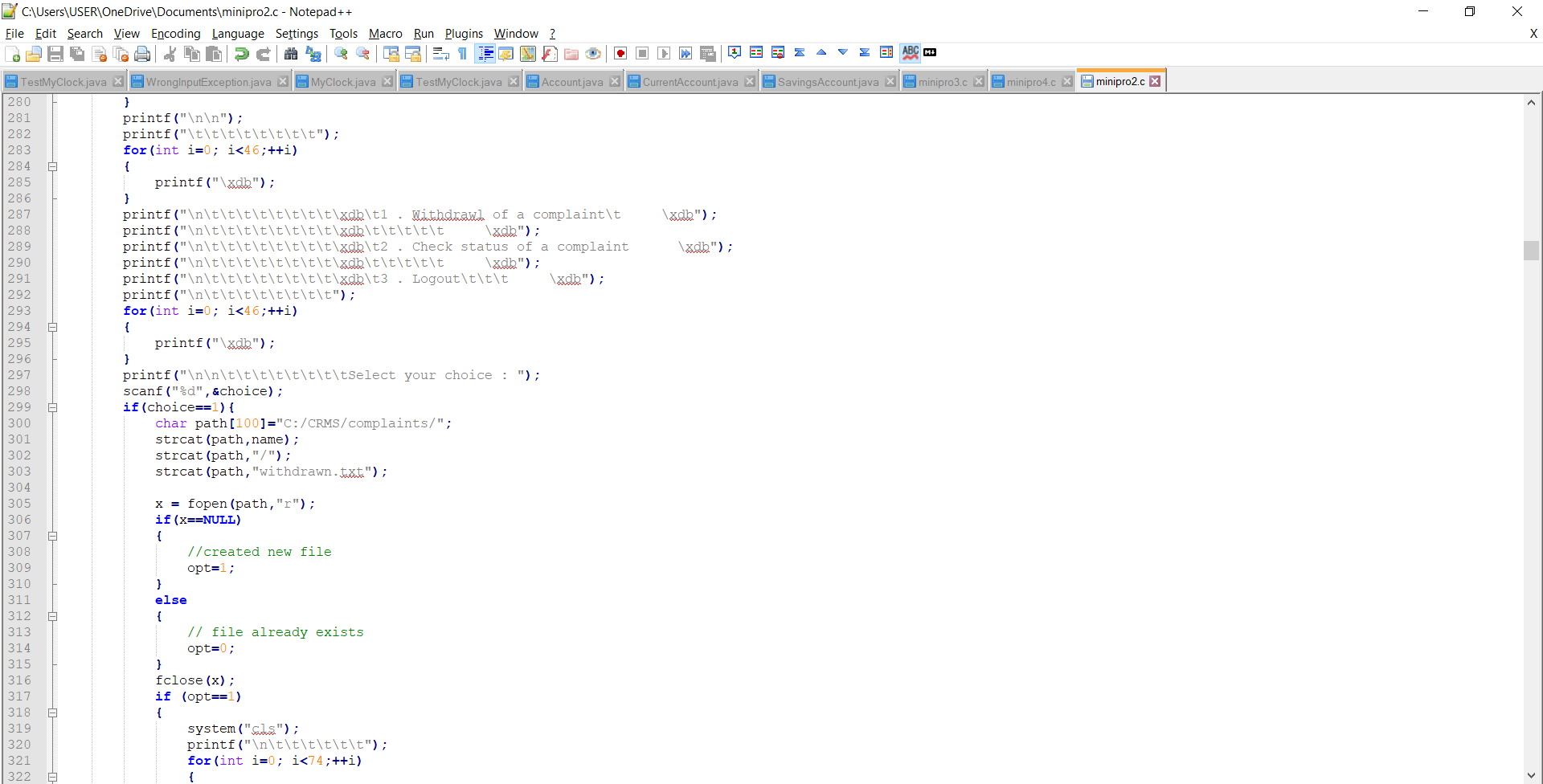
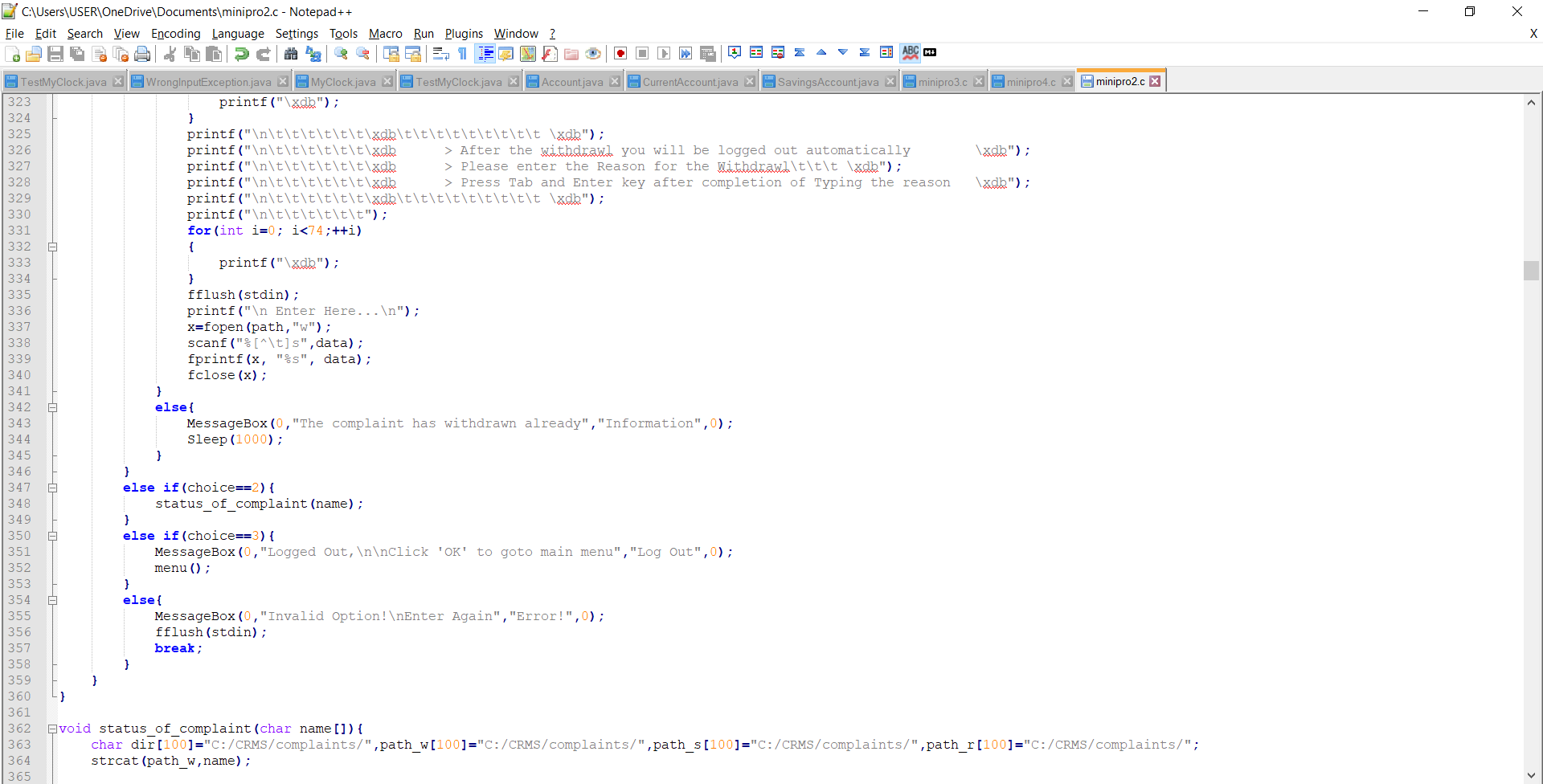
  

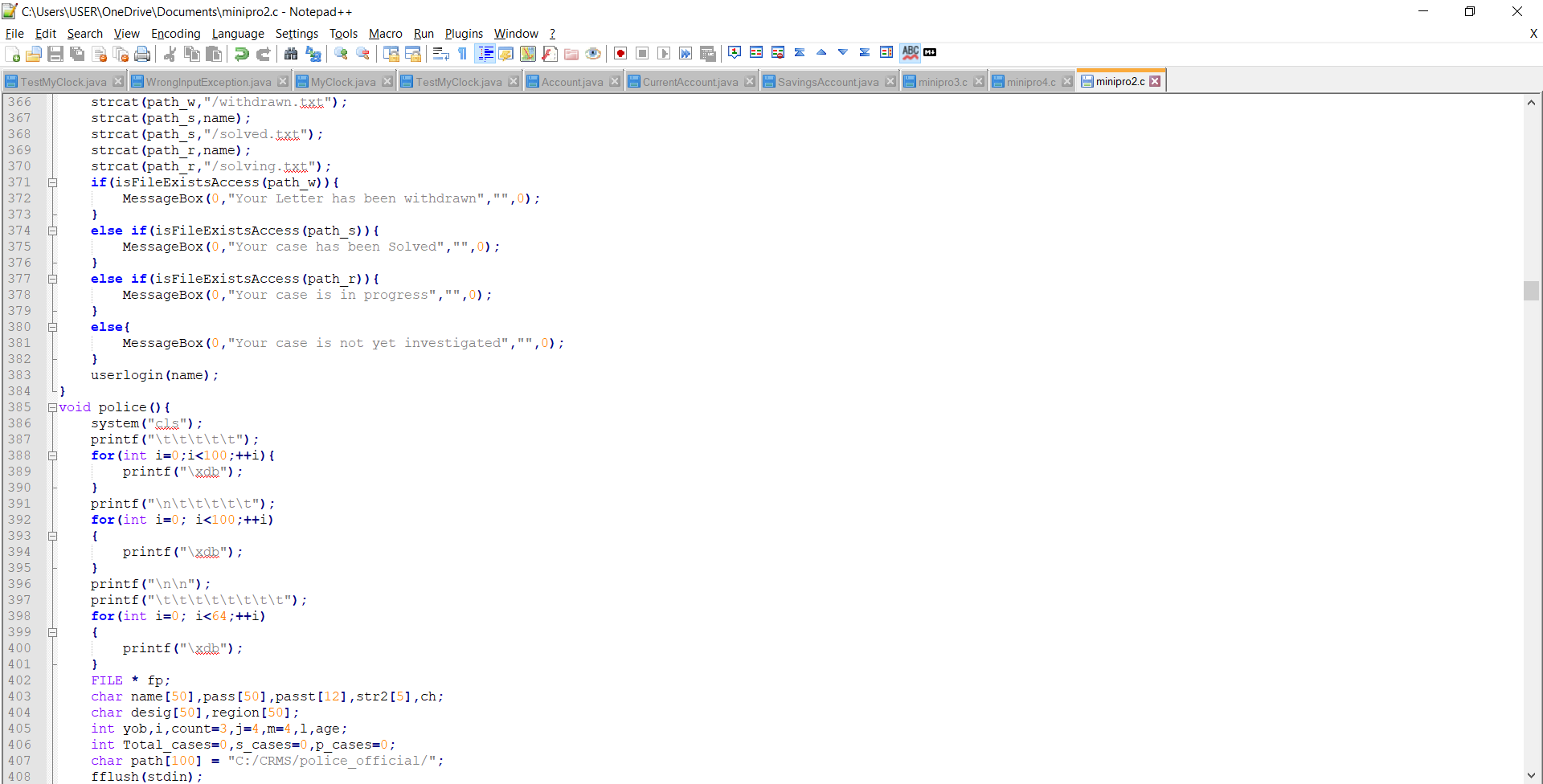


* + - 1. **USER(PUBLIC) REGISTER AND LOGIN**

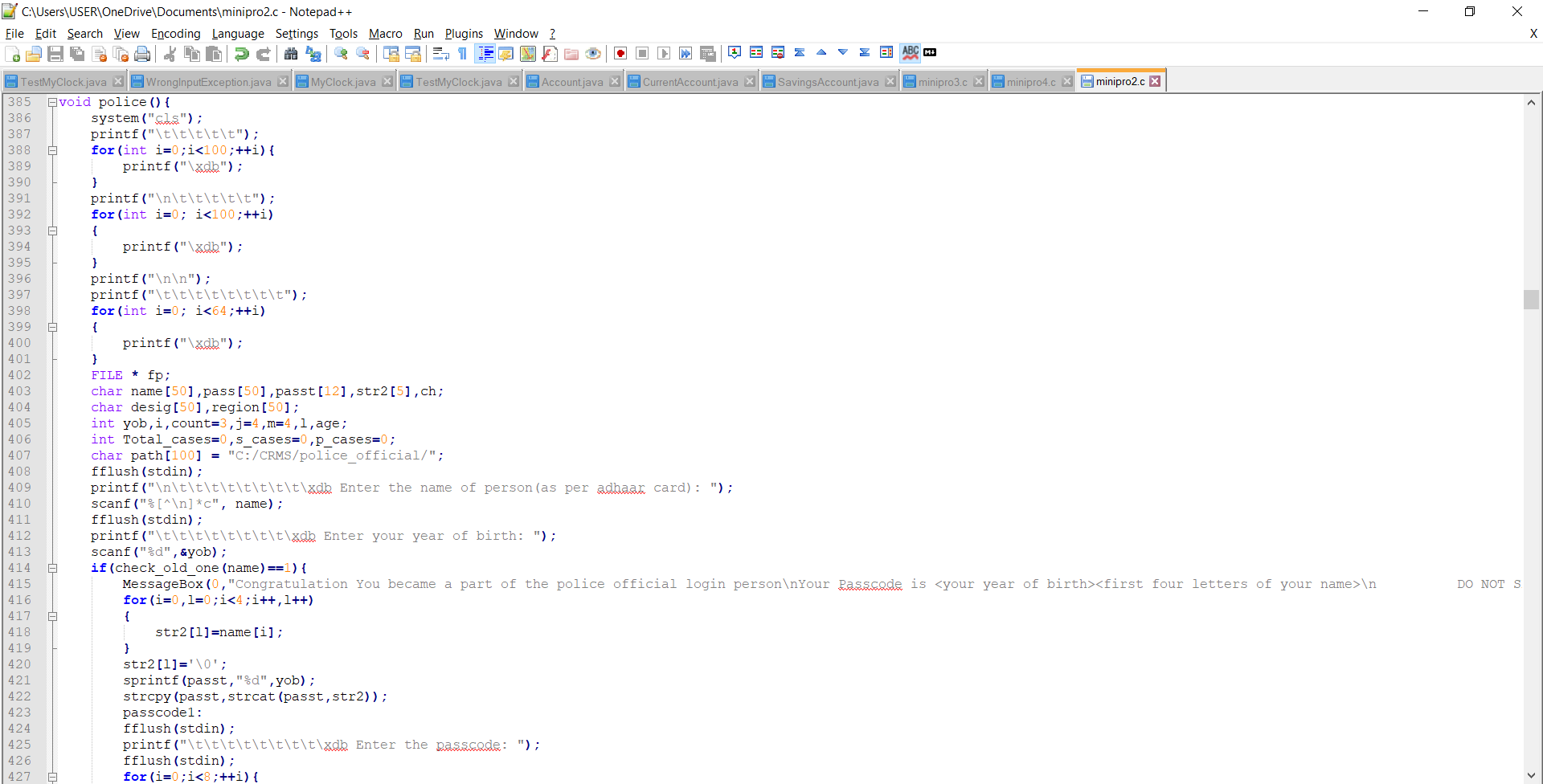


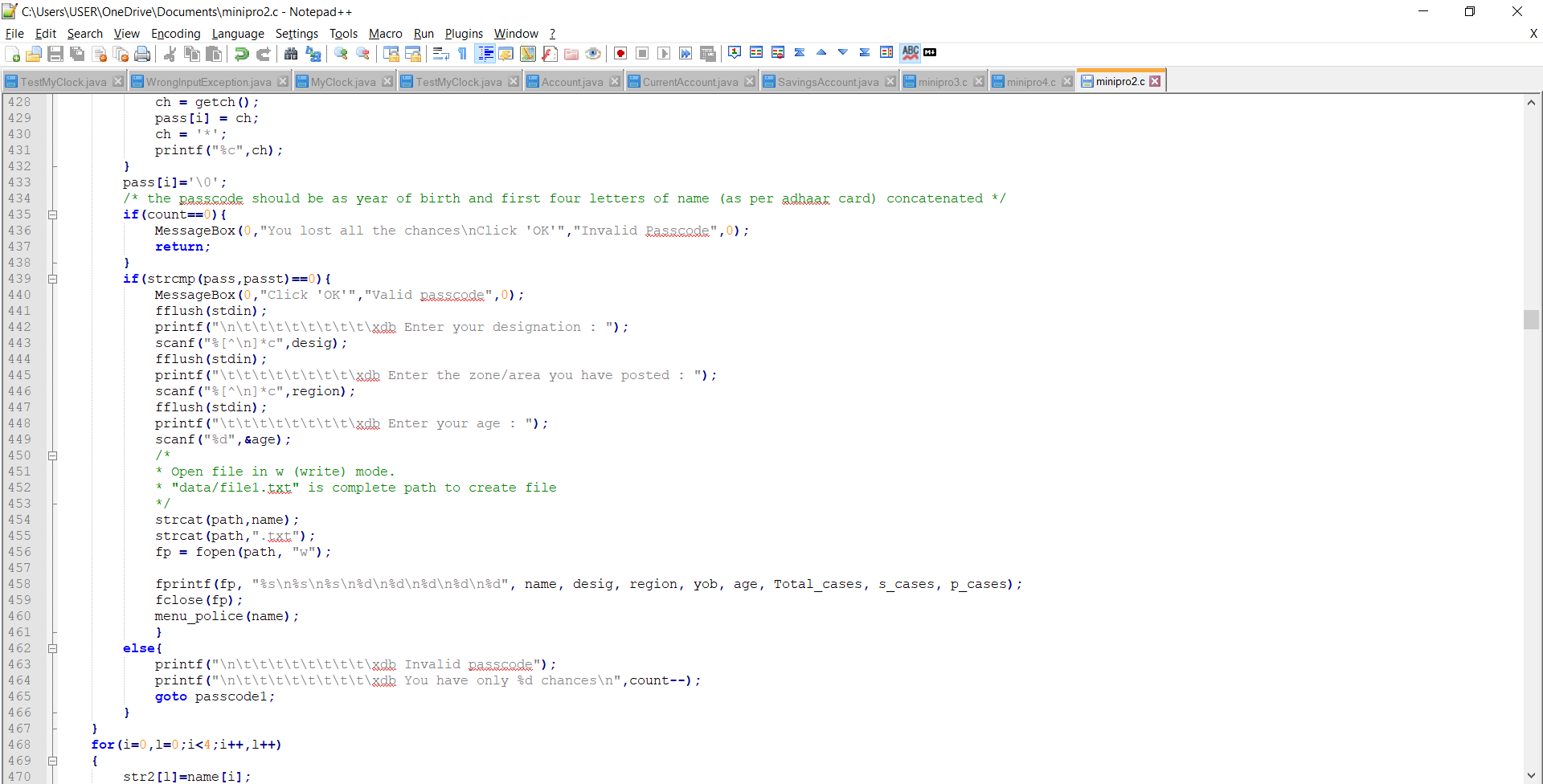
 

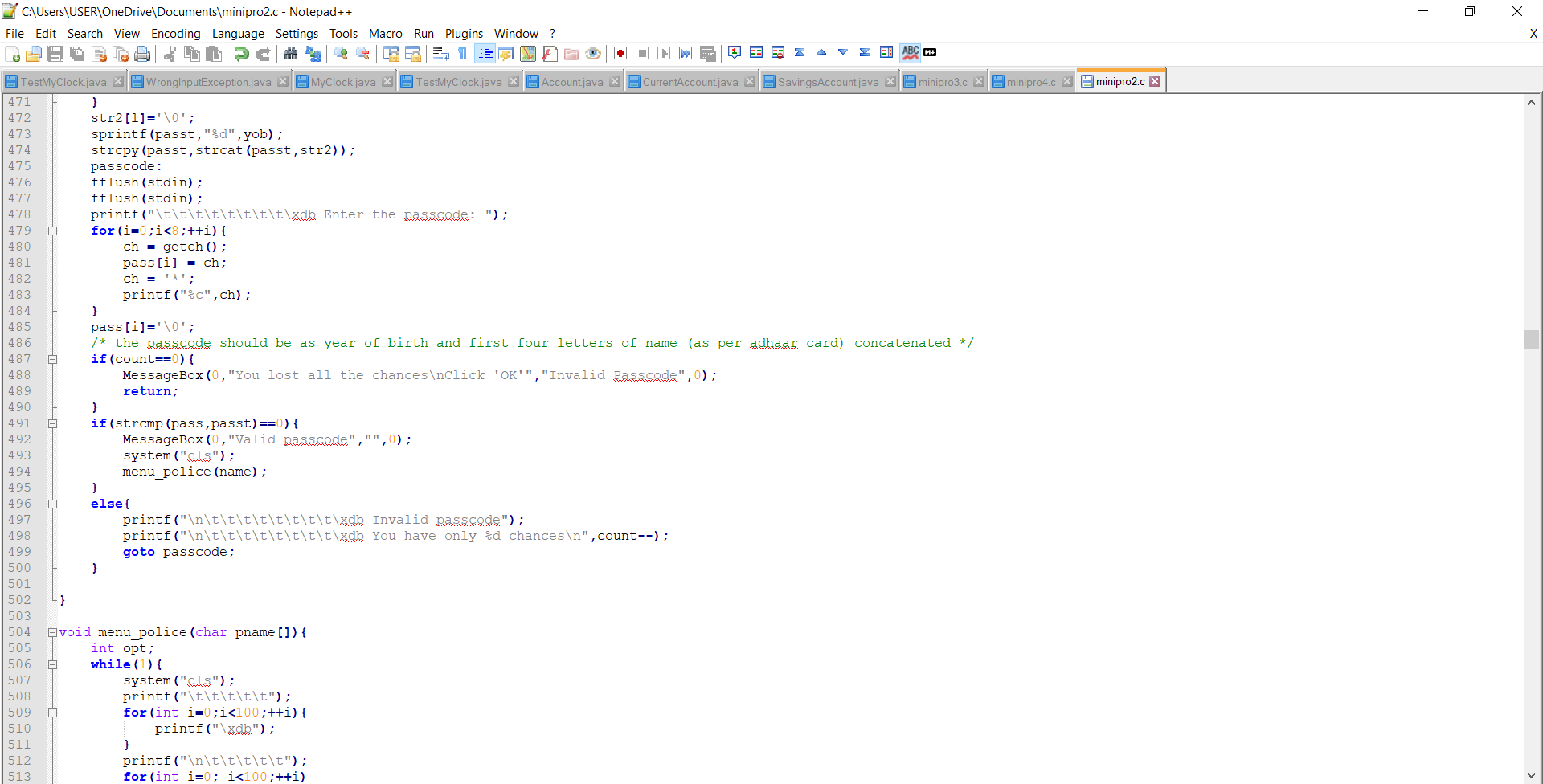
 

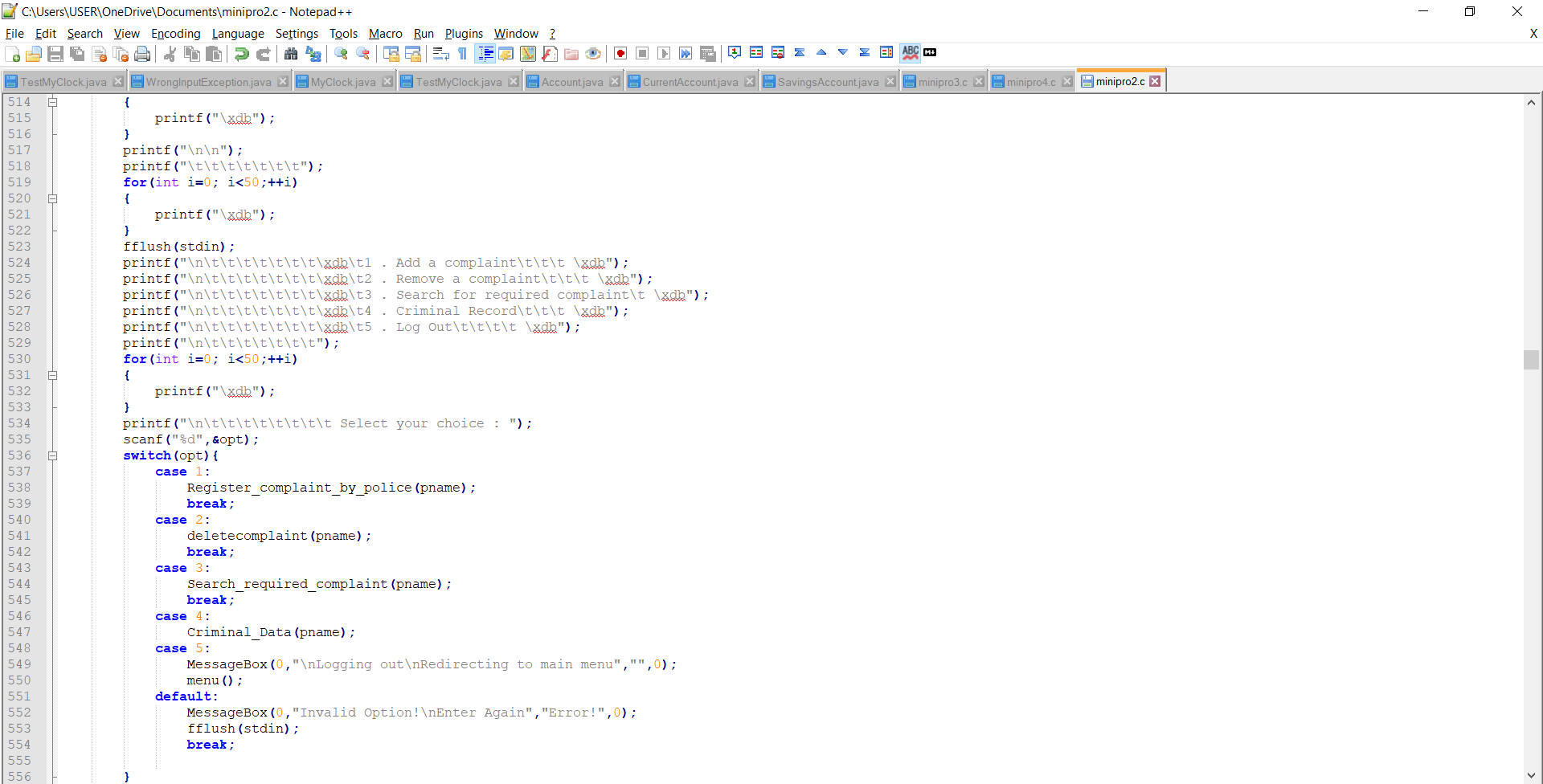


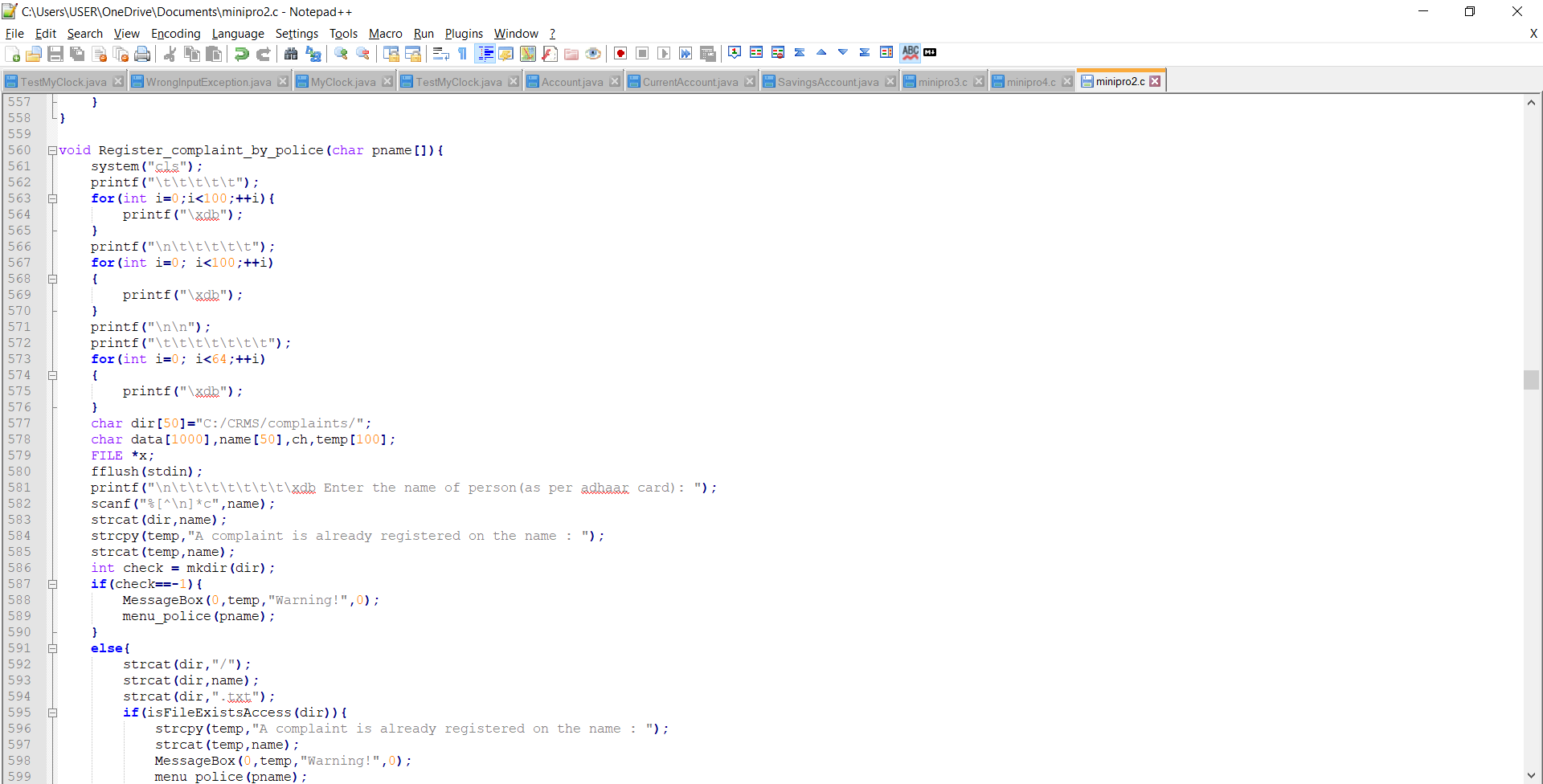
* + - 1. **Police Official Source Code**

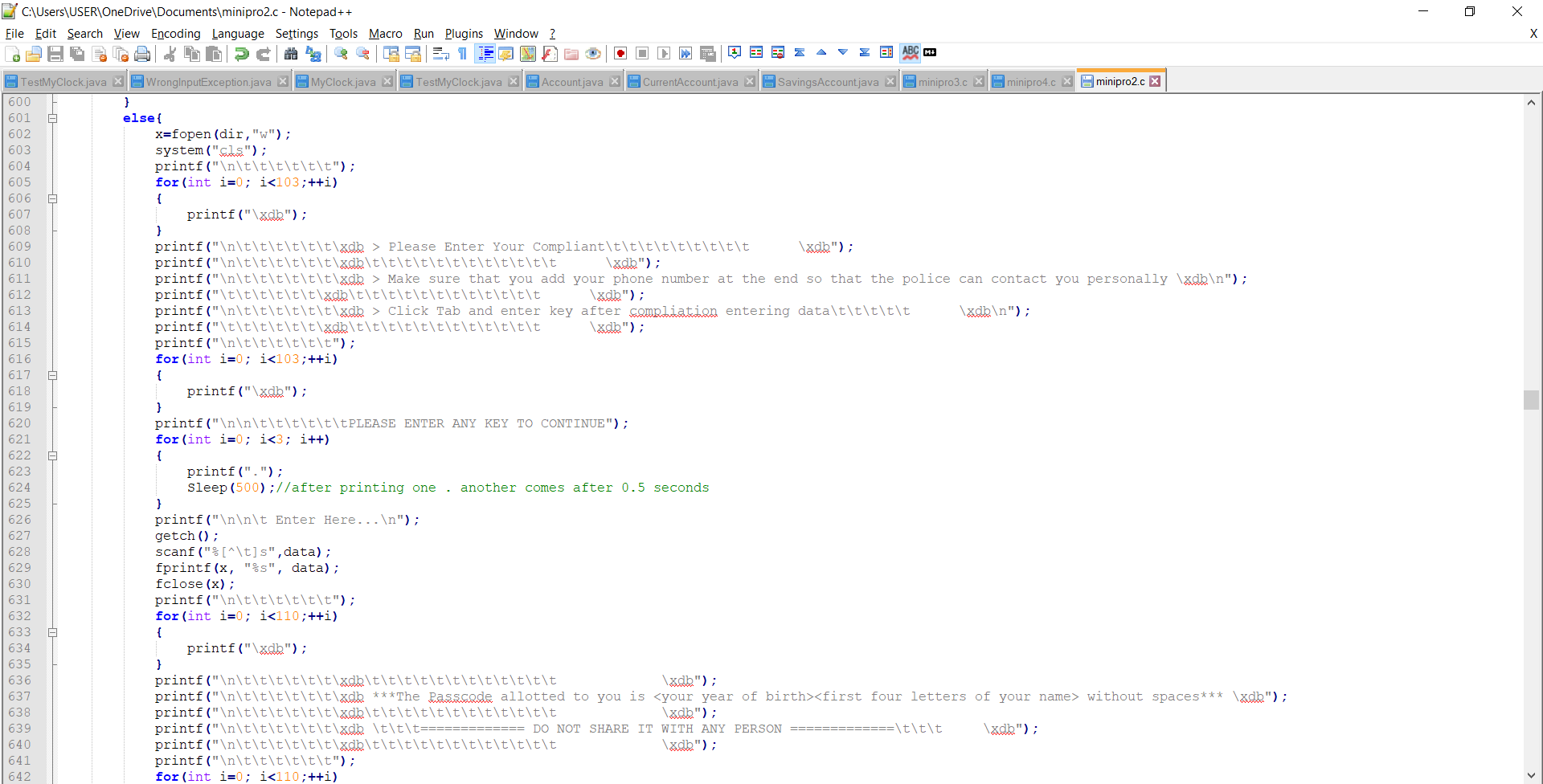


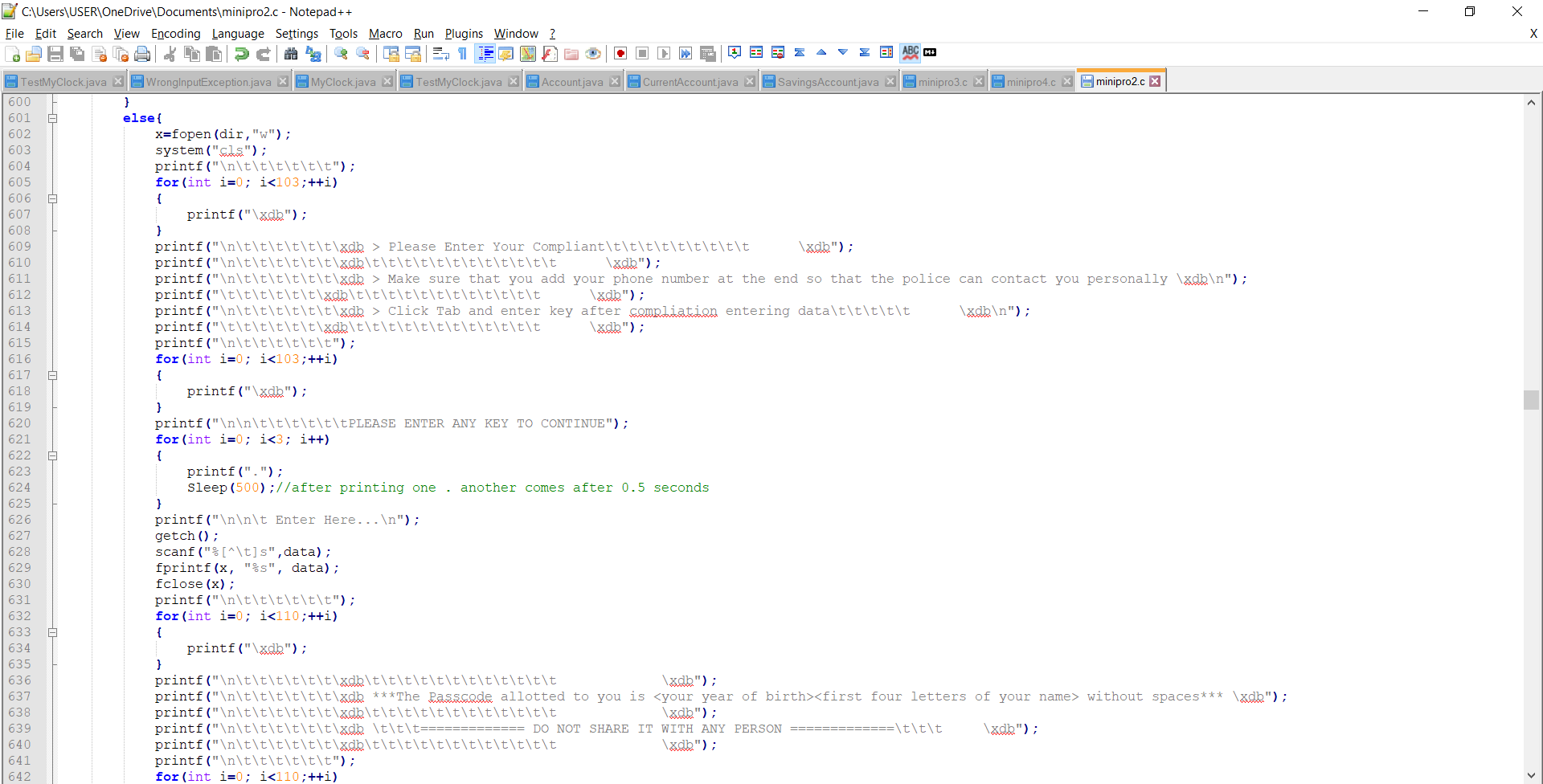


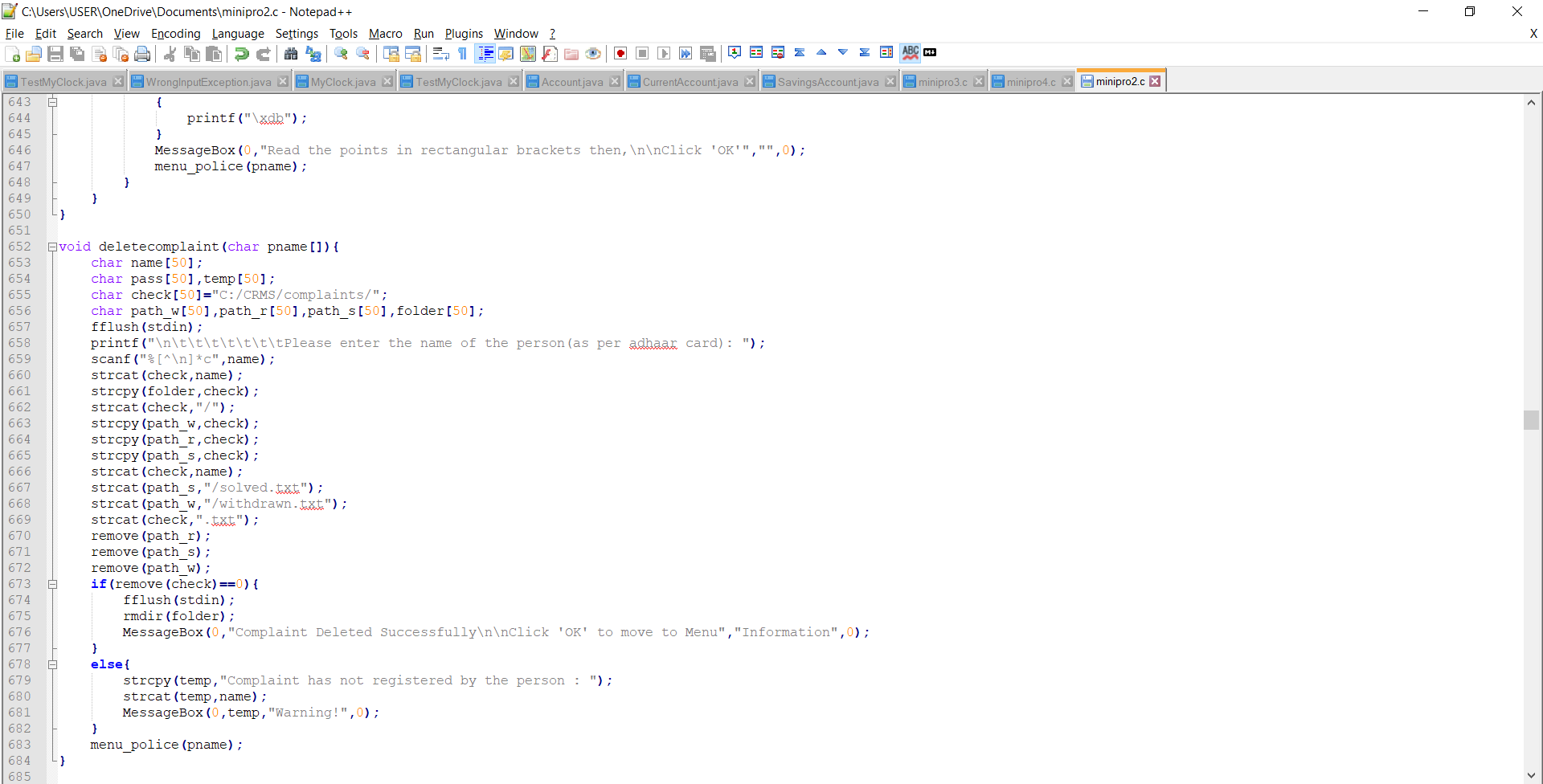


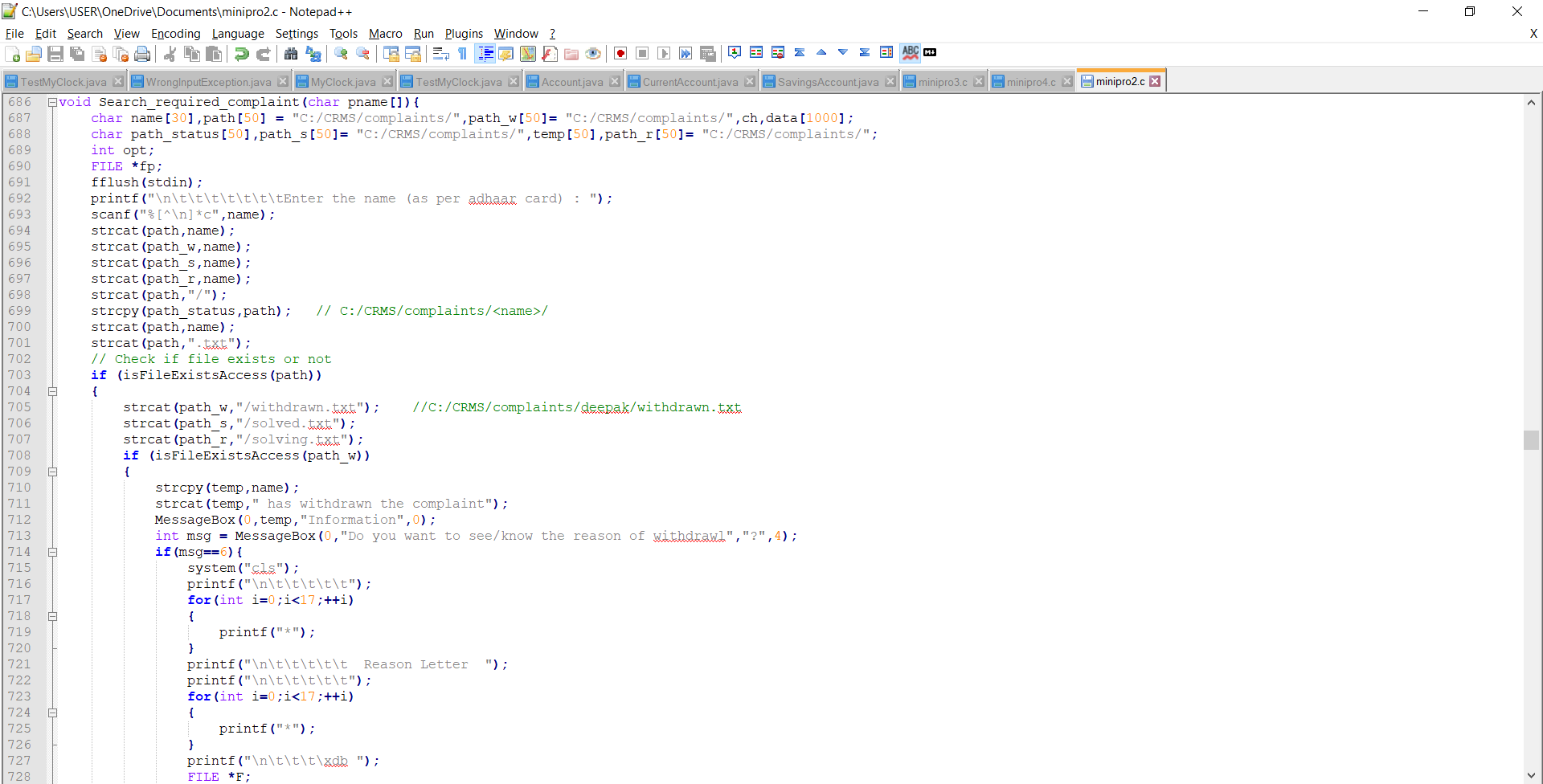


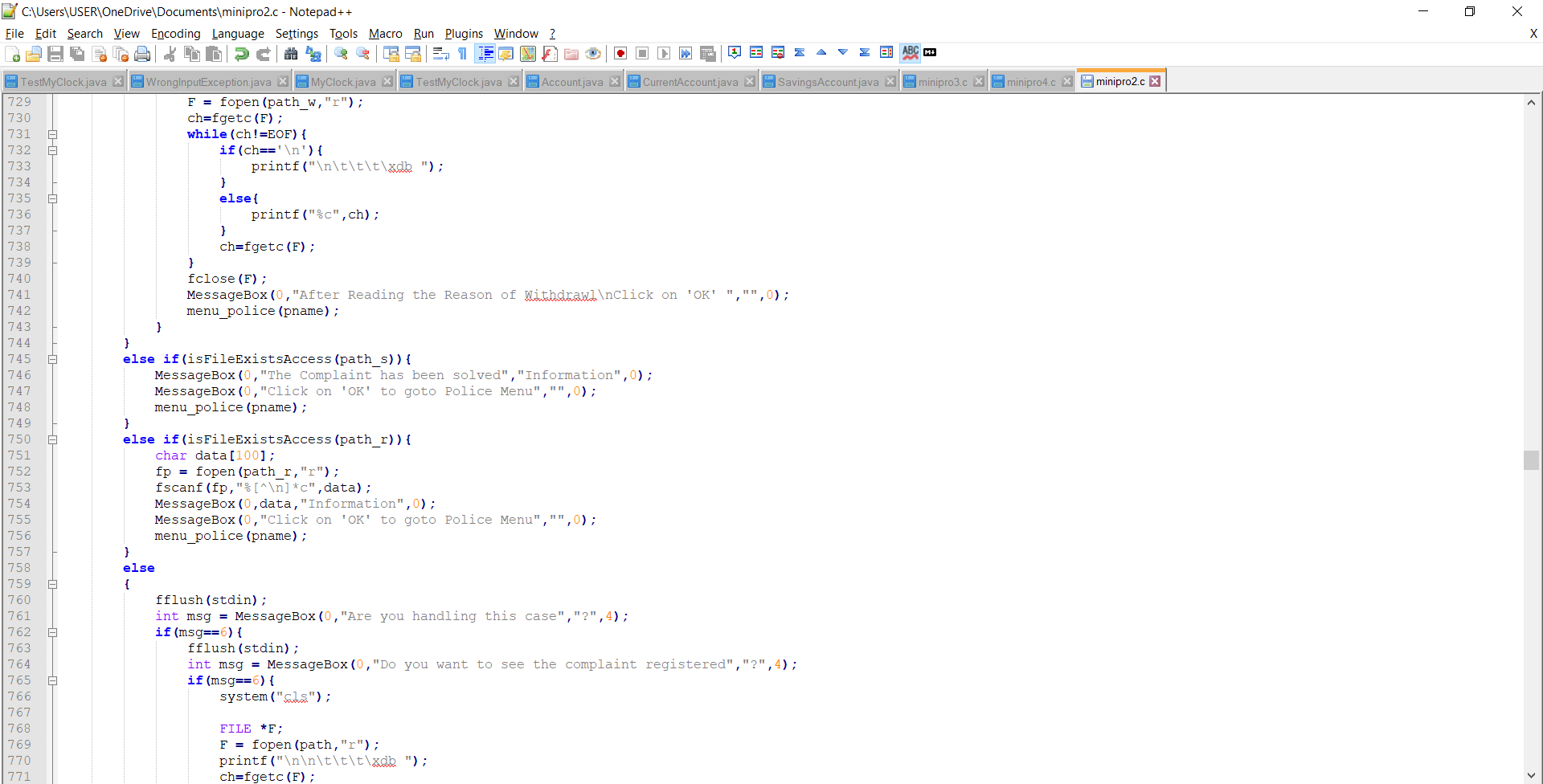


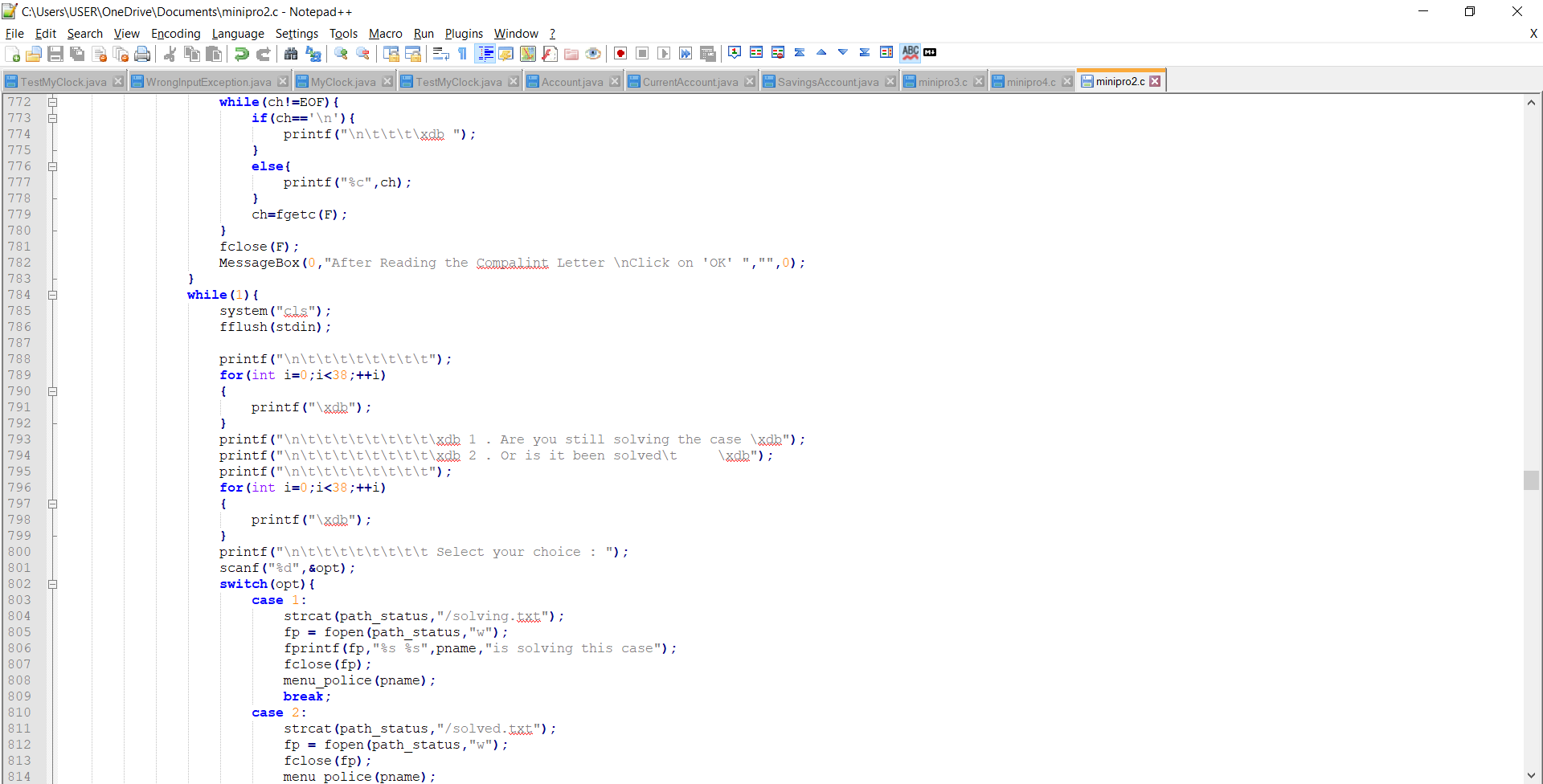


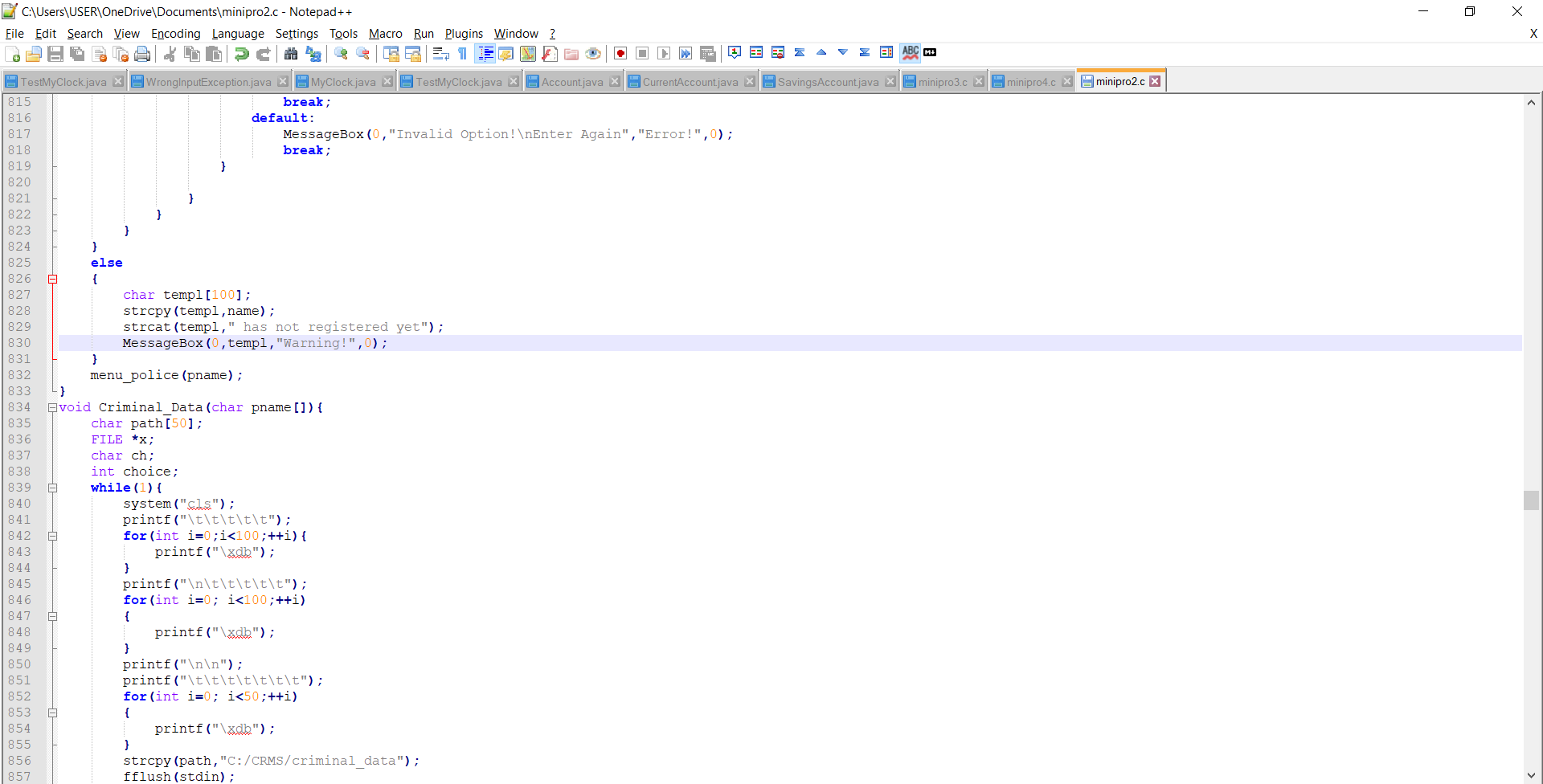


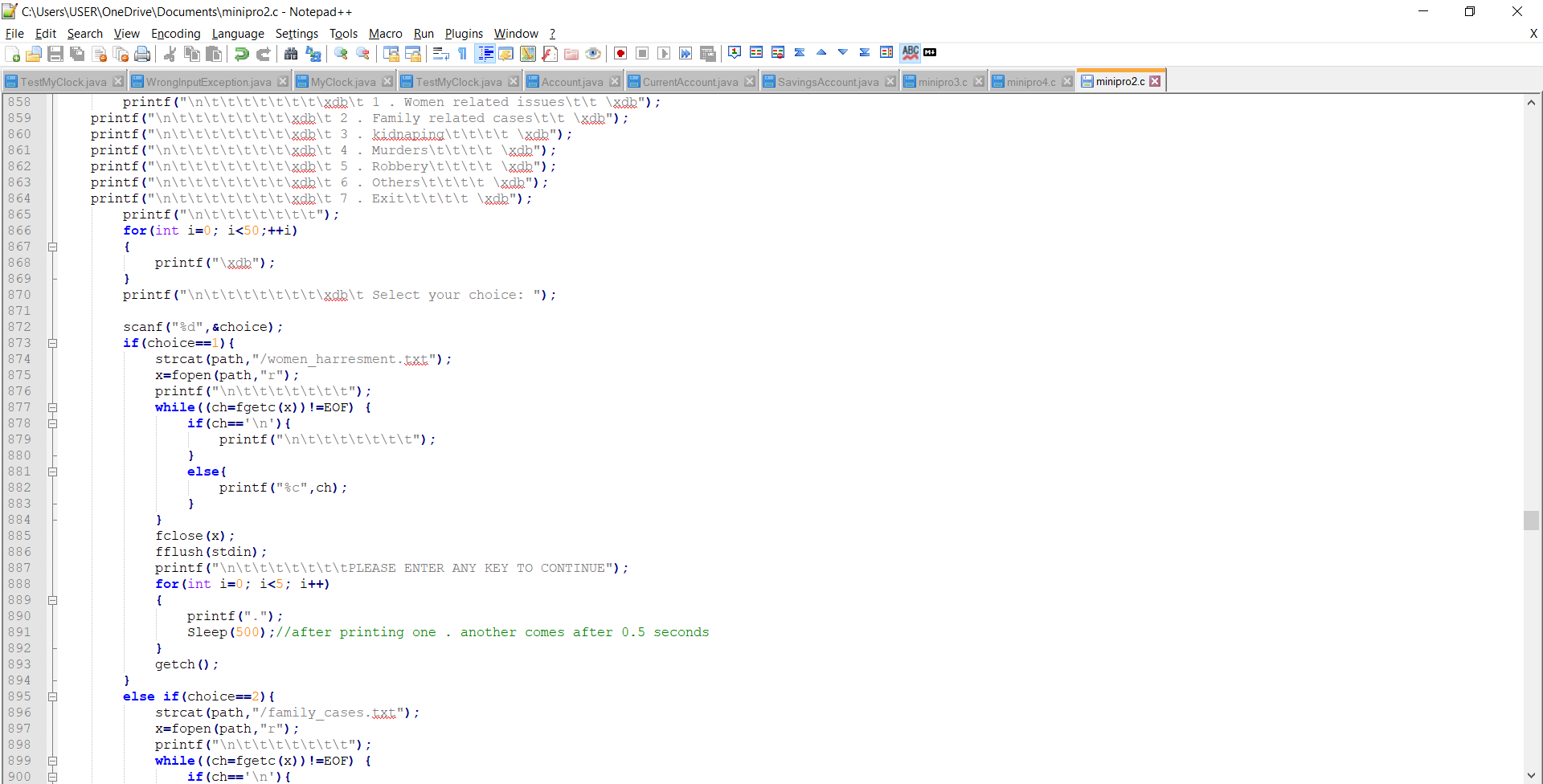


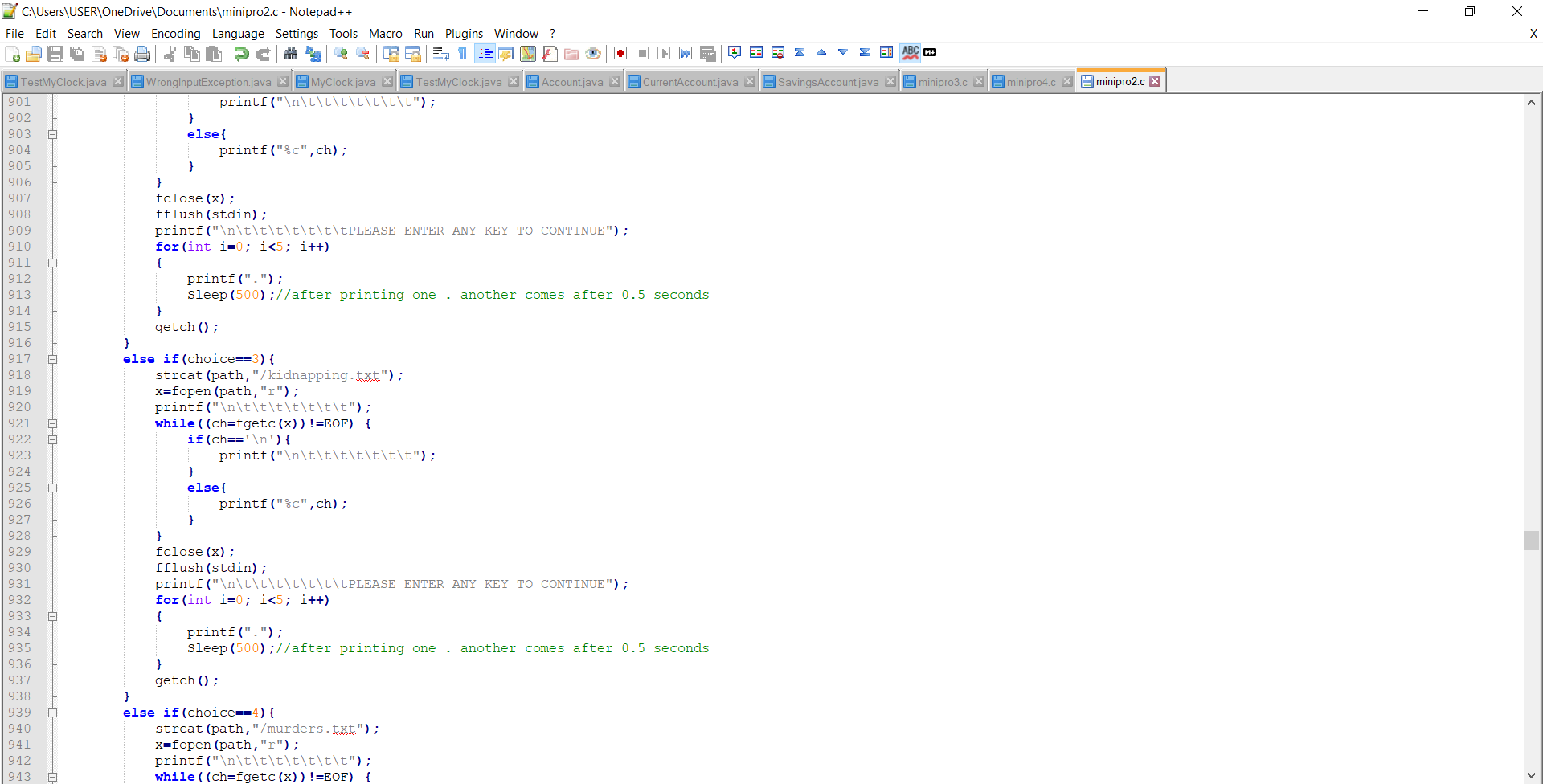


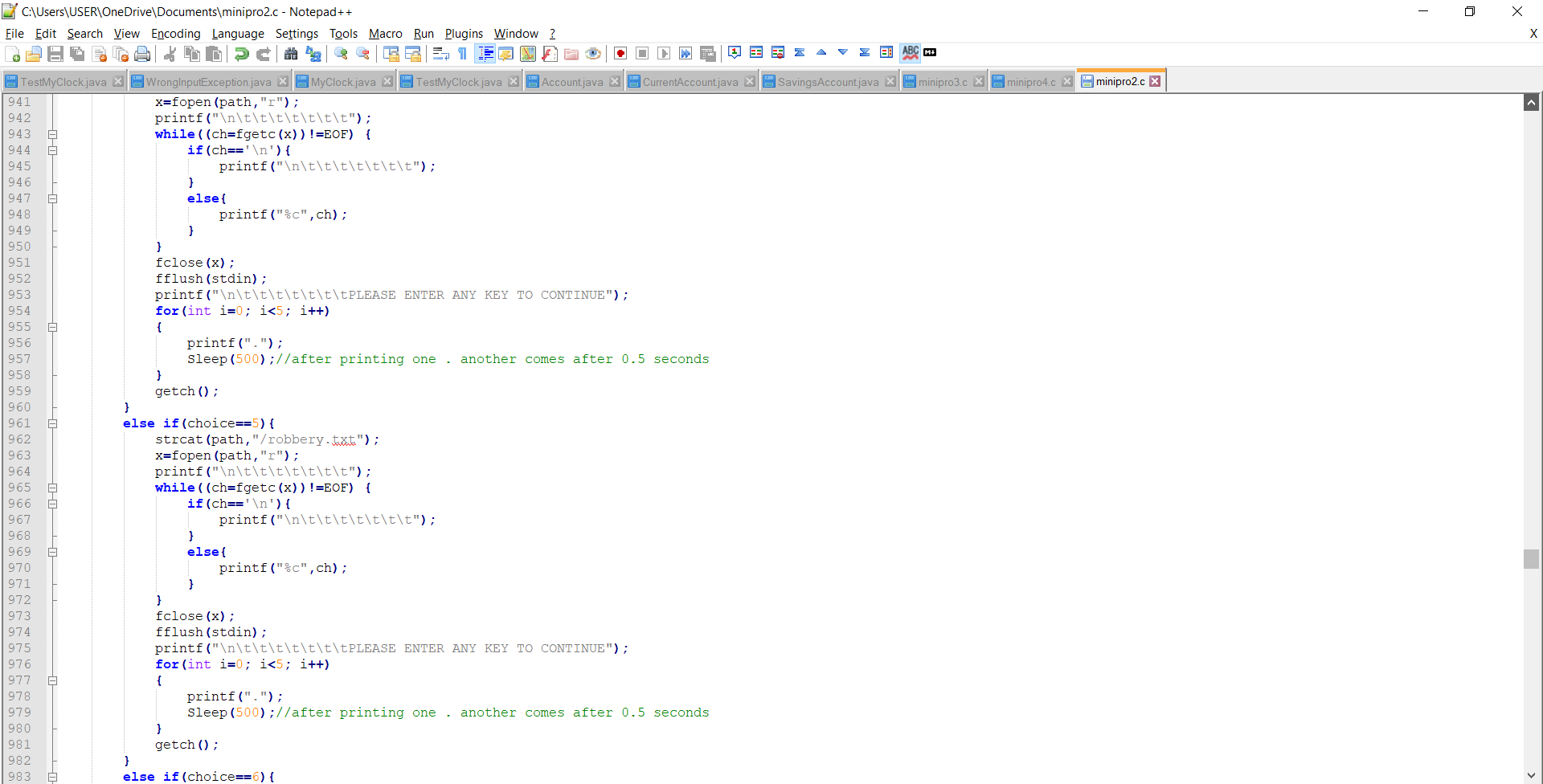


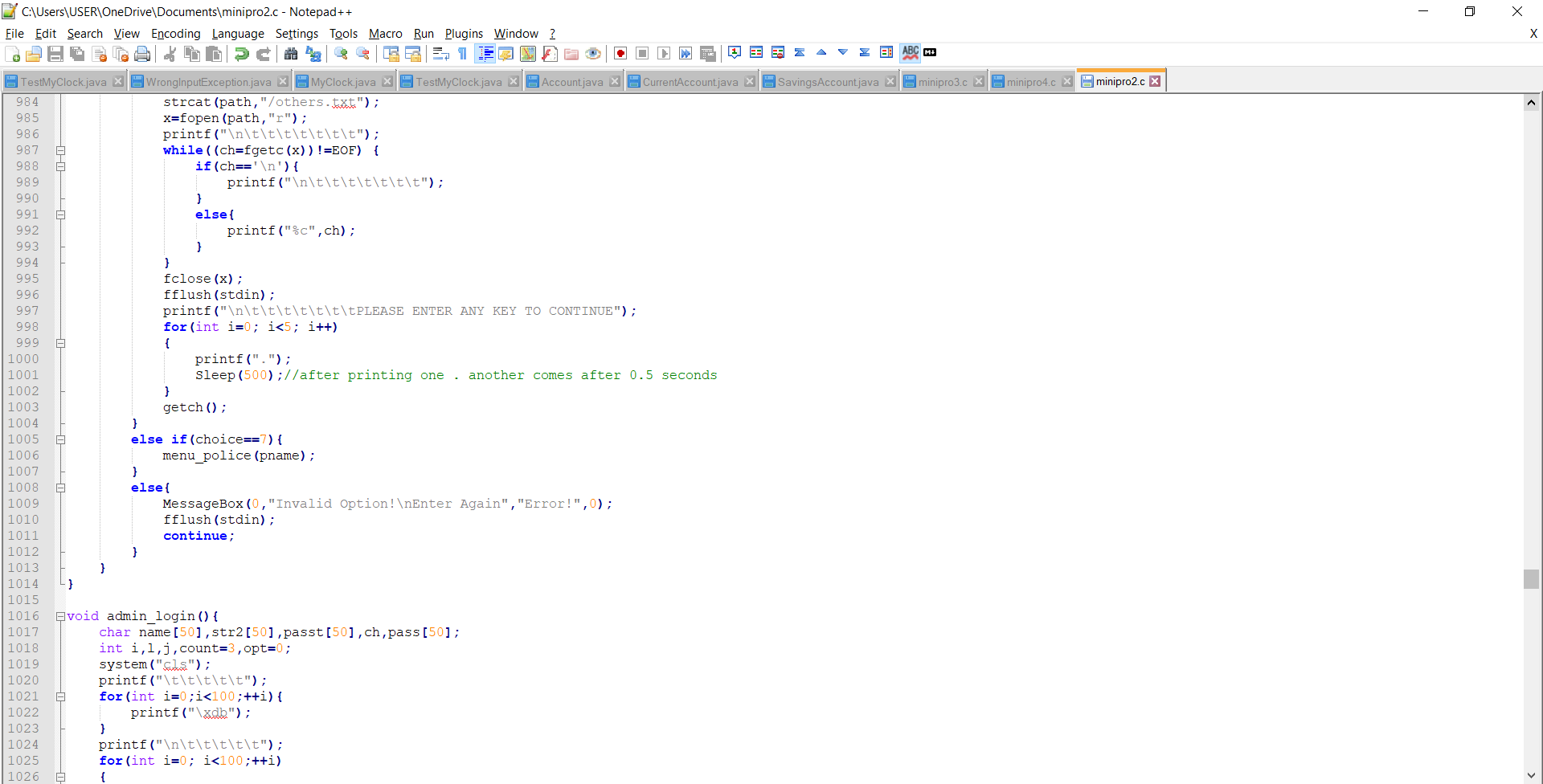




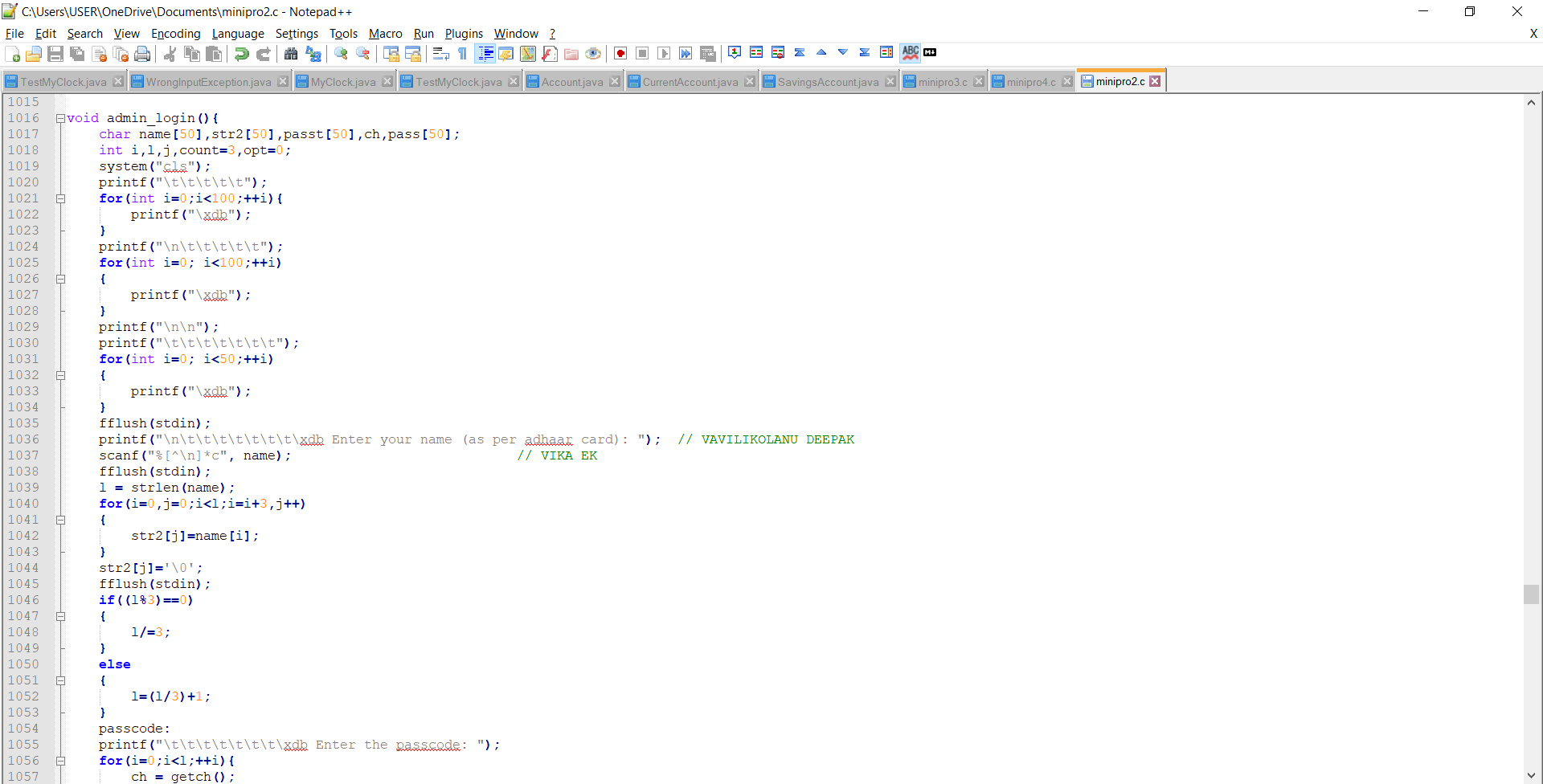


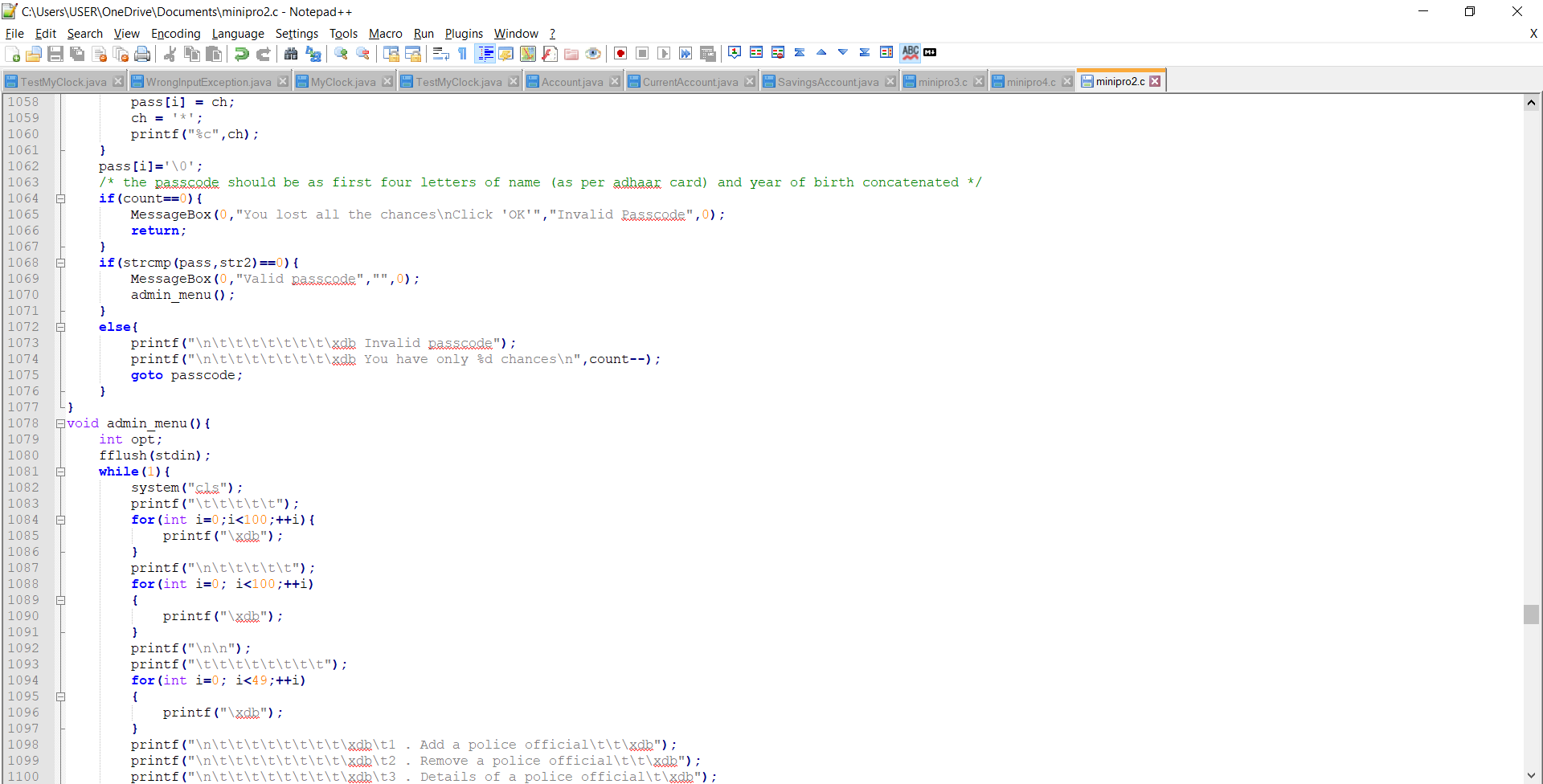


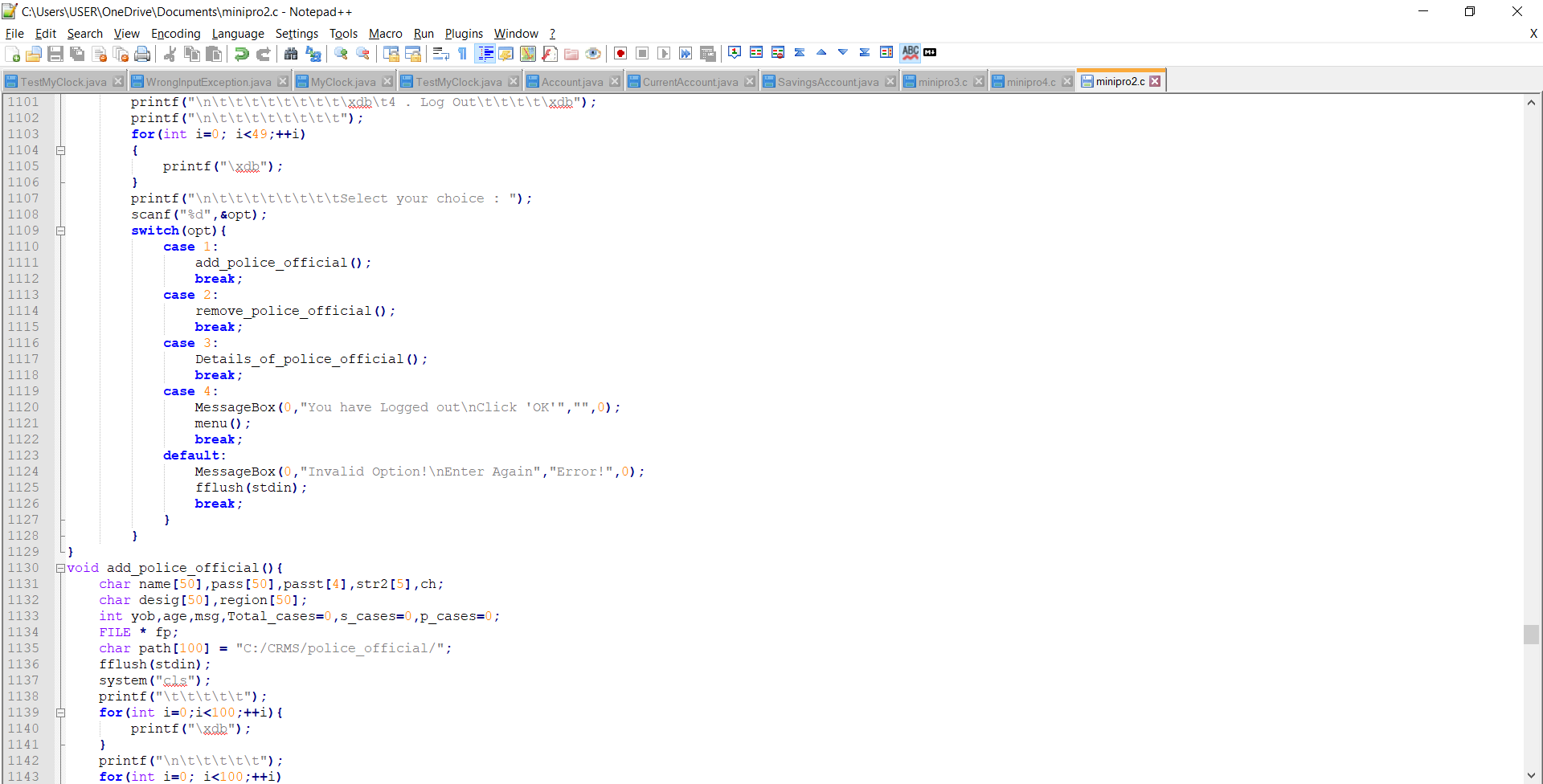


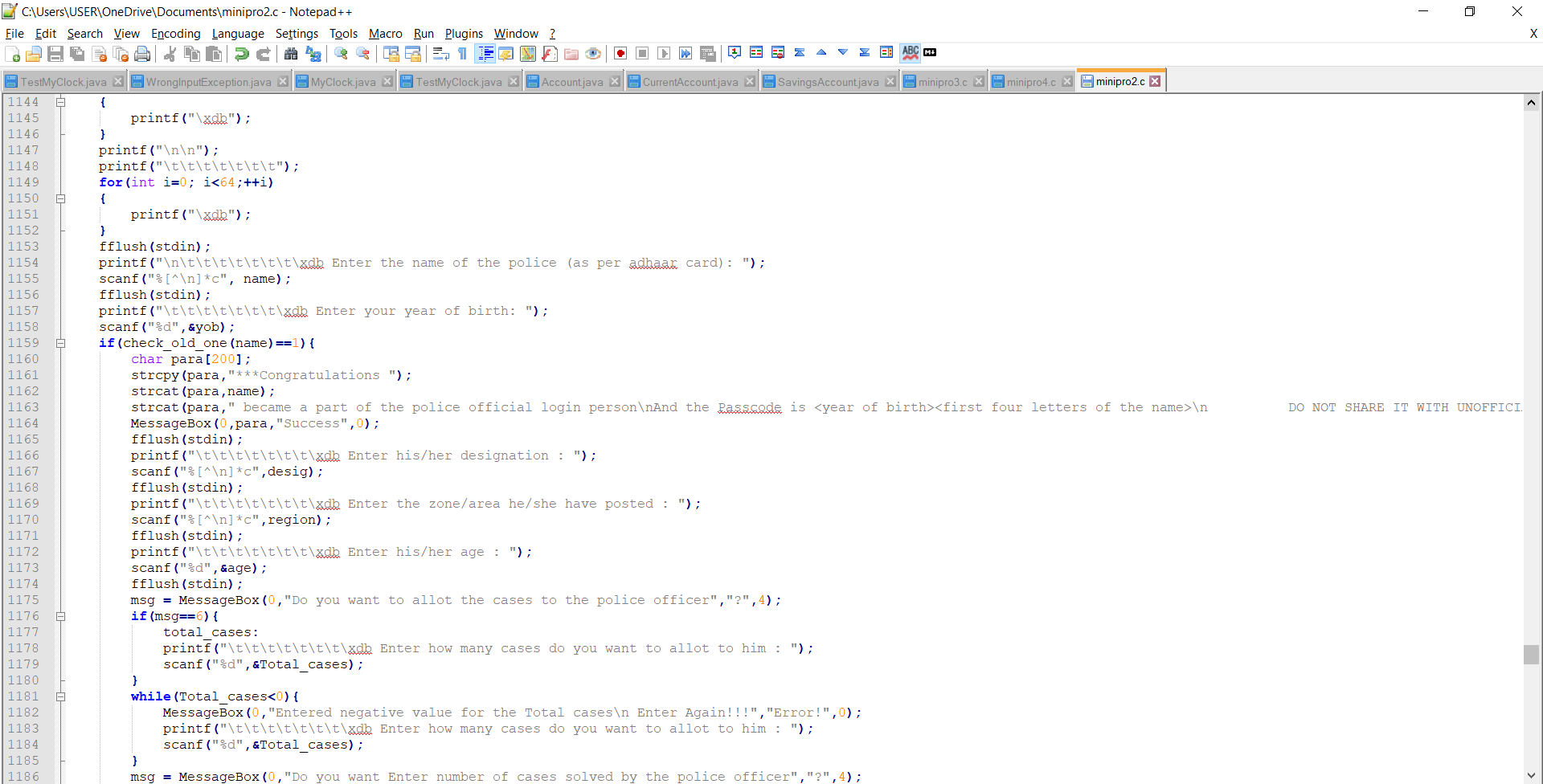


* + - 1. **ADMIN**

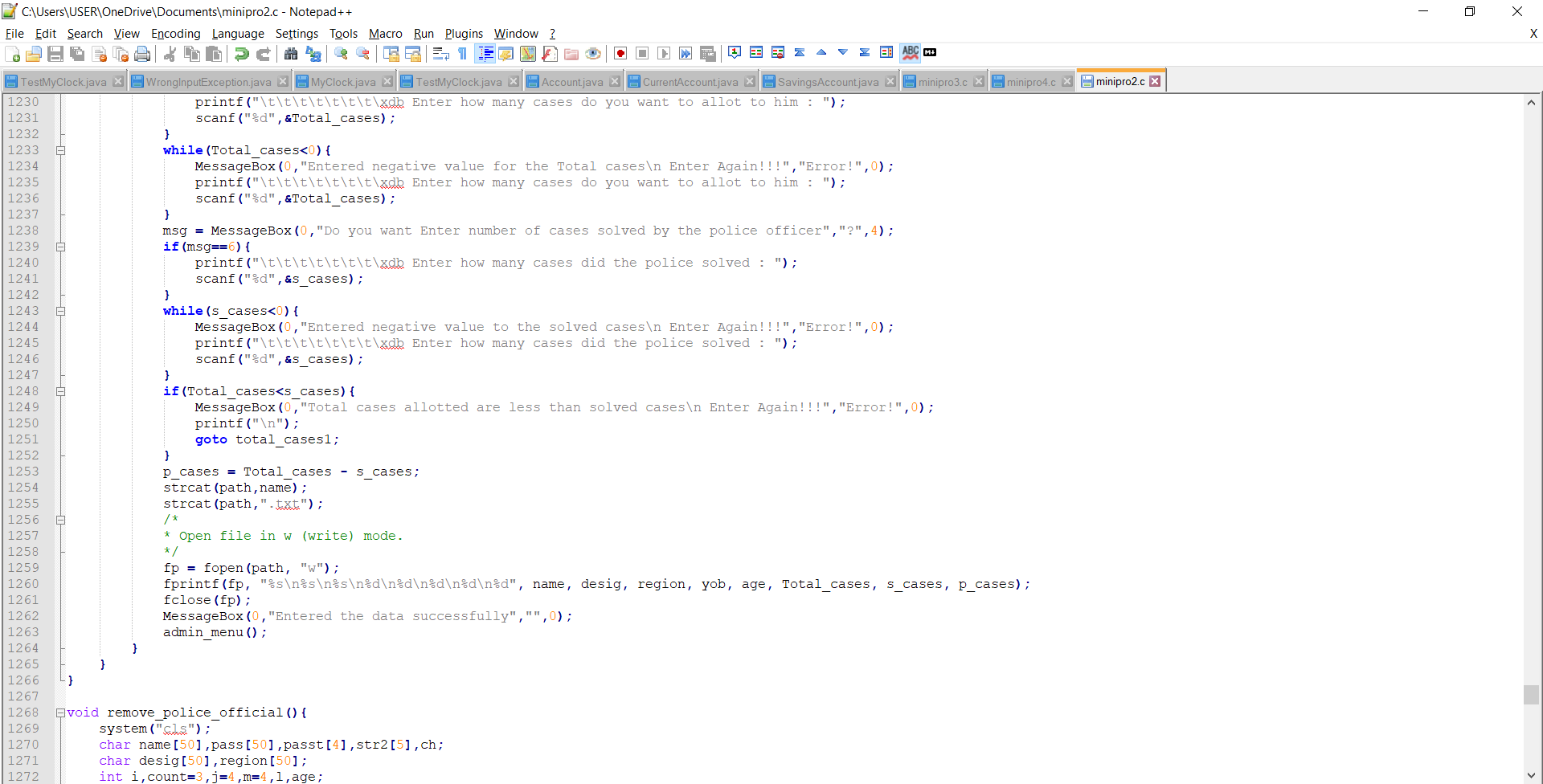


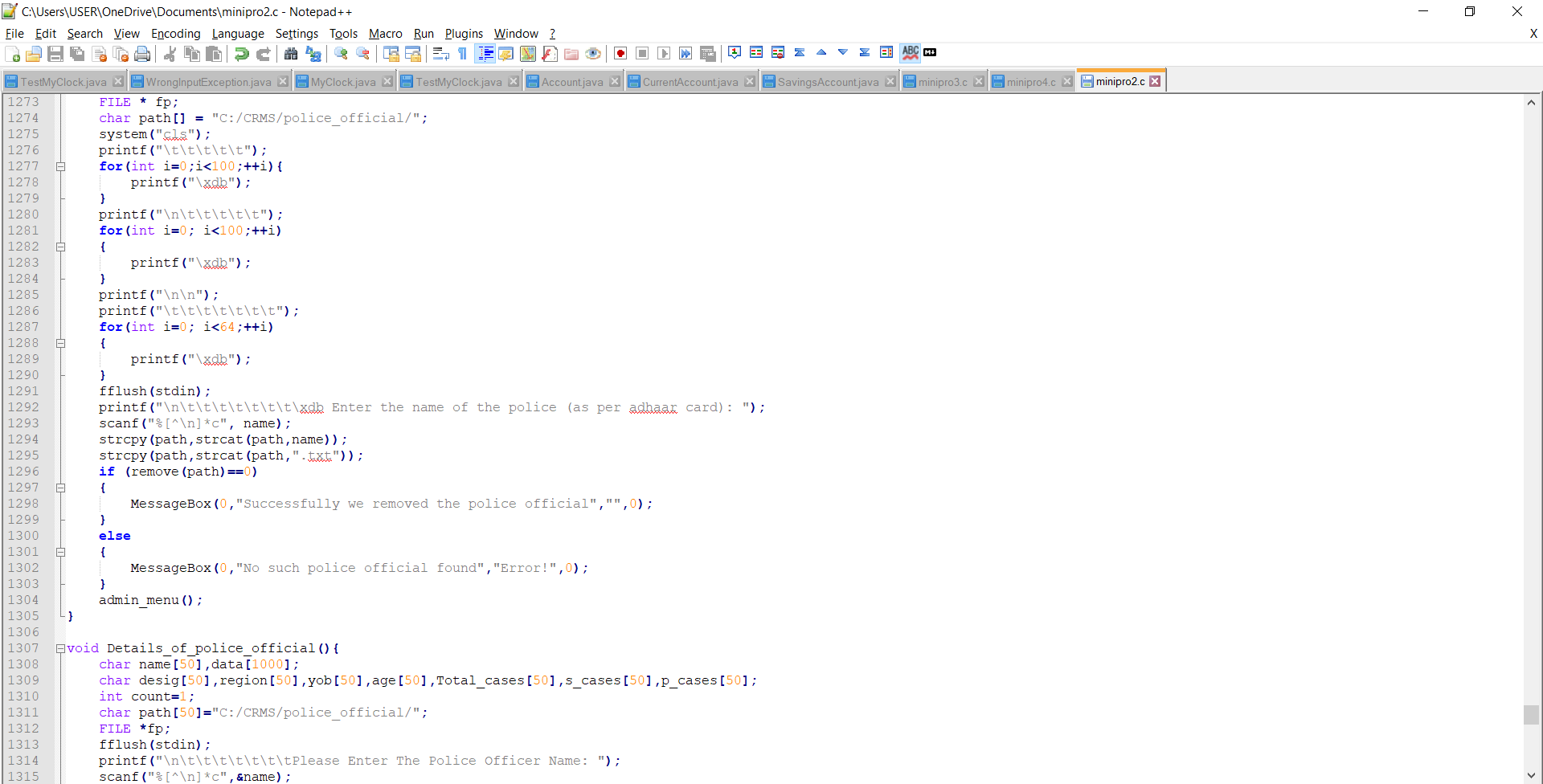


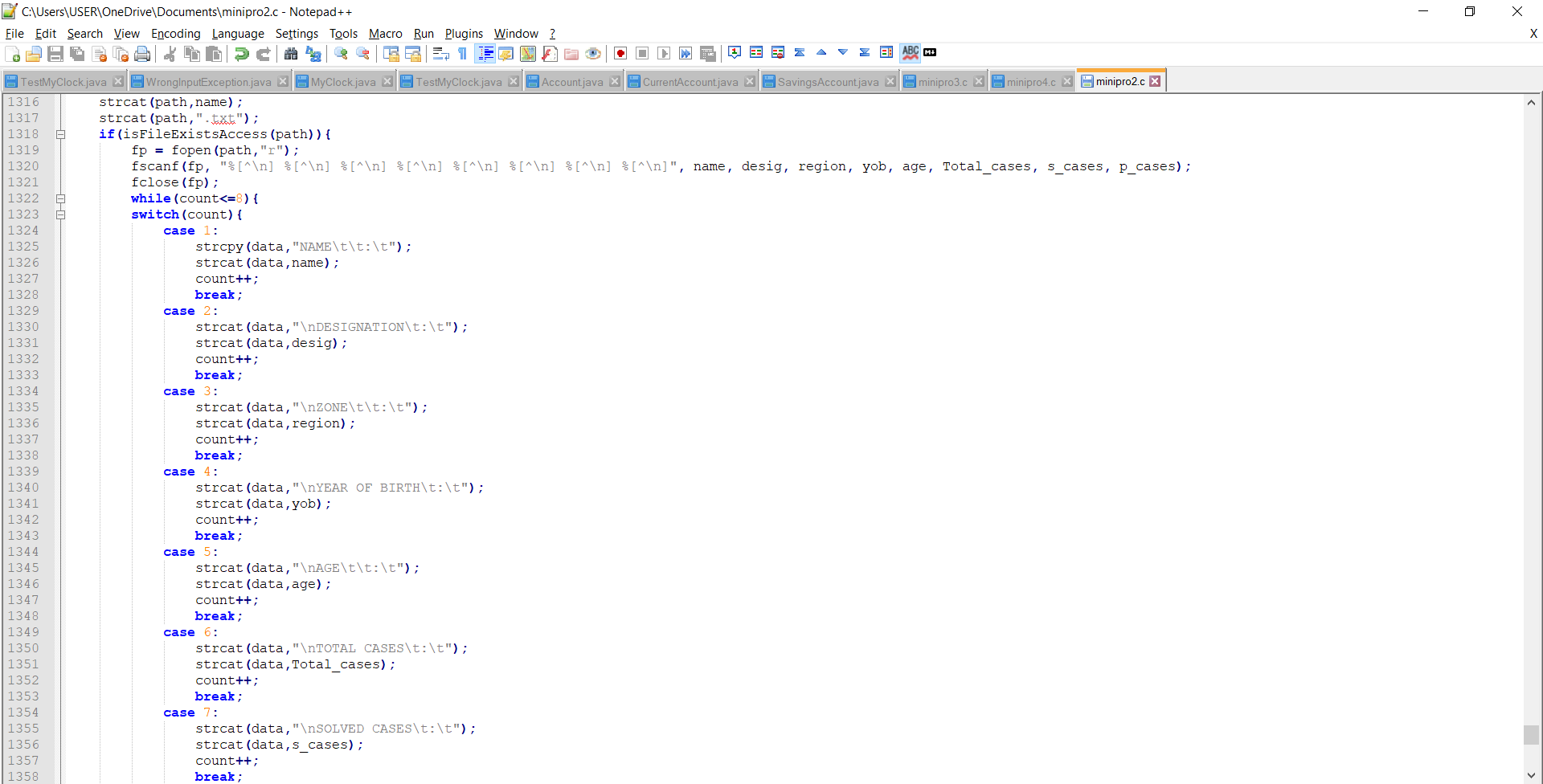


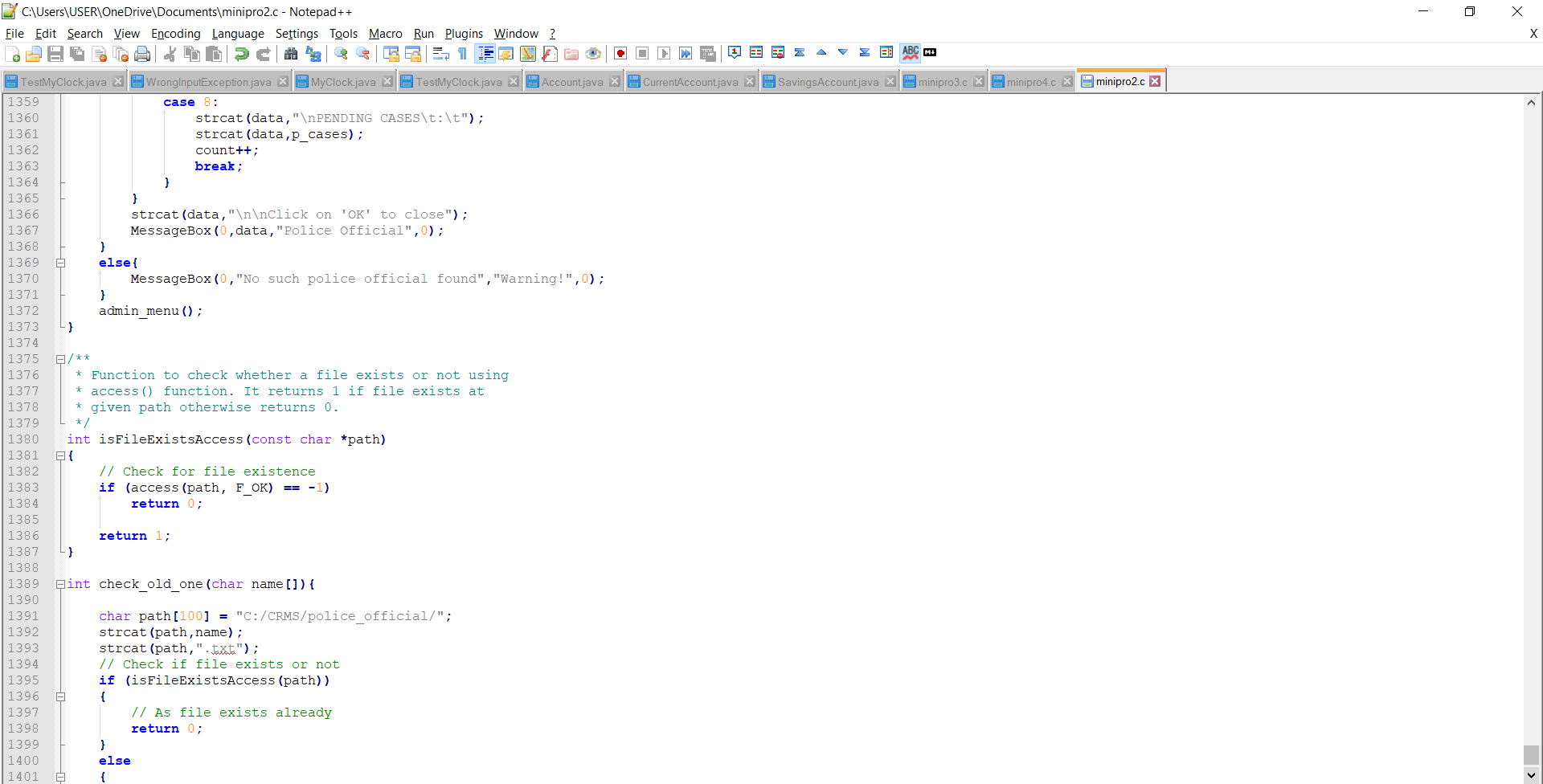


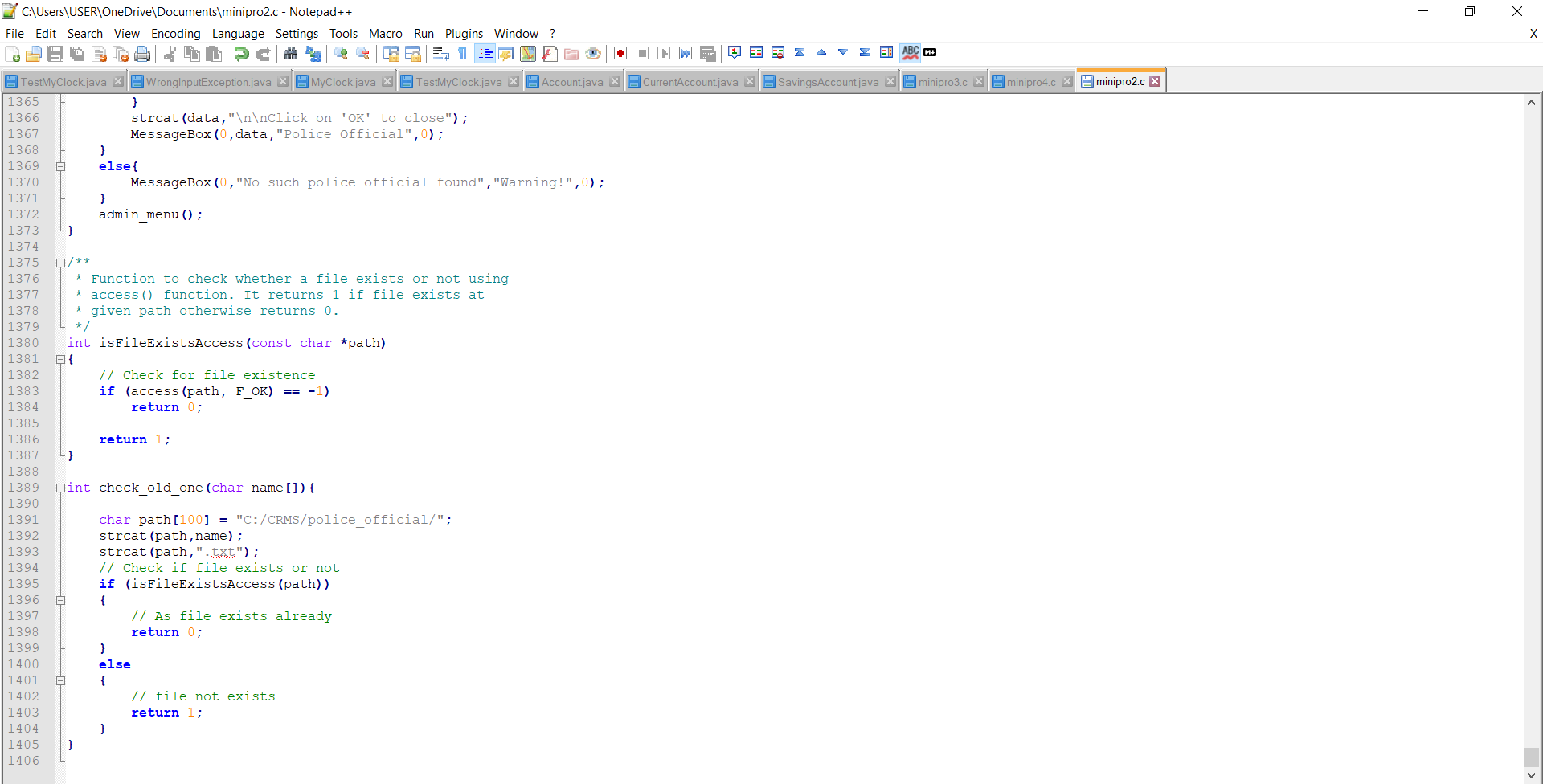






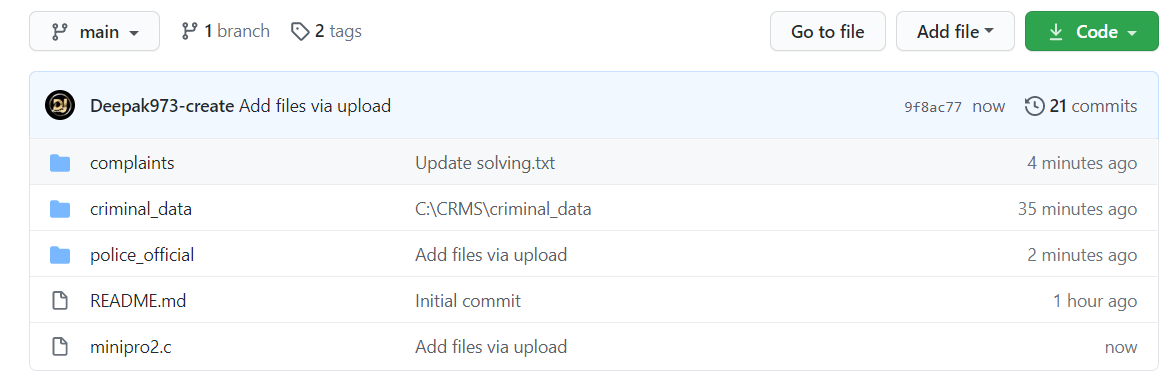


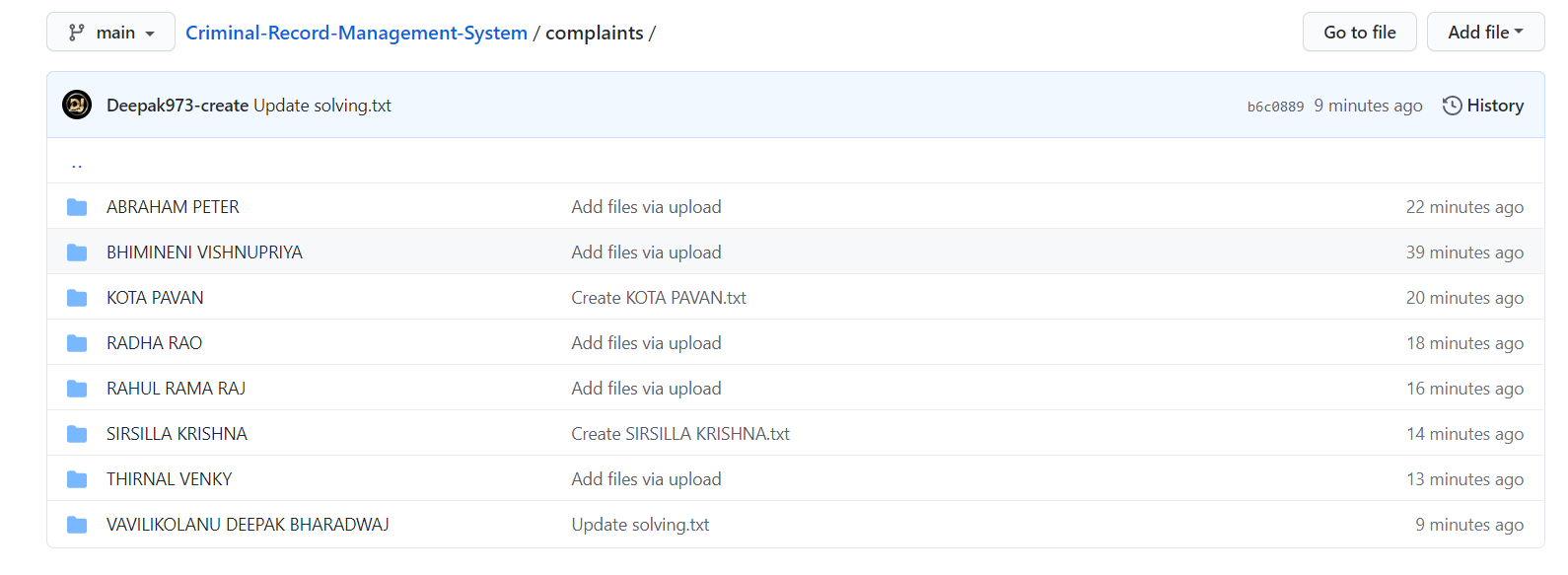




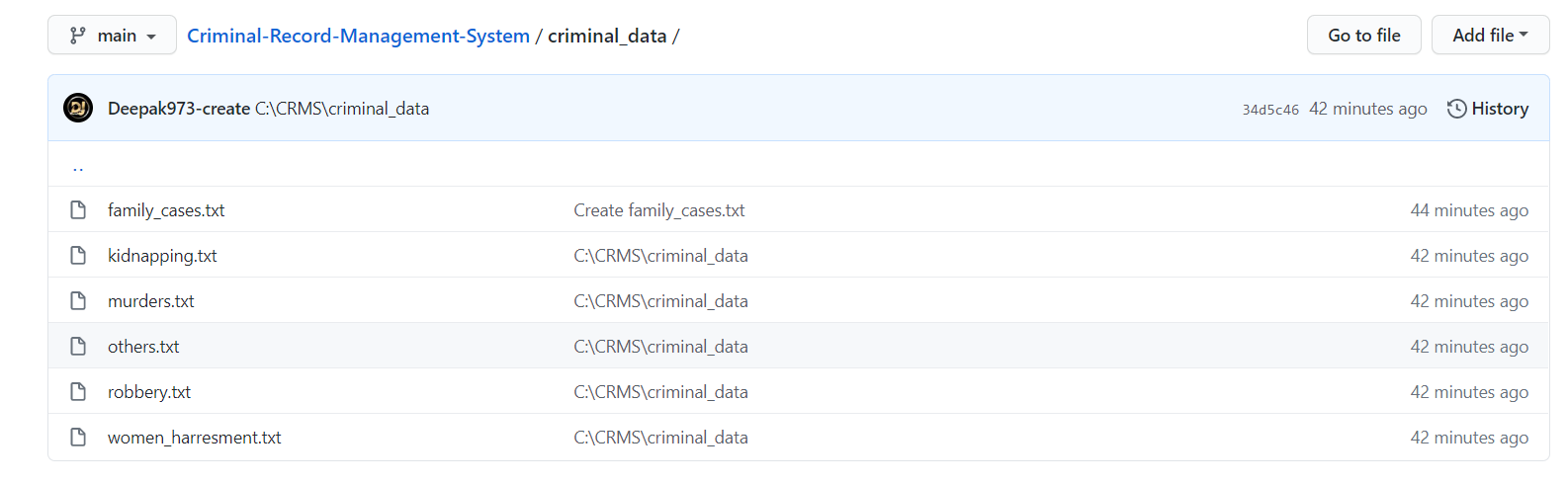
* + 1. **GITHUB/FOLDER STRUCTURE**

We have segregated the files we have used based on the usage. We made 3 folders, one contains all the complaints, next contains all the criminal record data, and other contains all the police official data. We also have a README which has a brief description of our project. The minipro2.c contains the source code of our project.

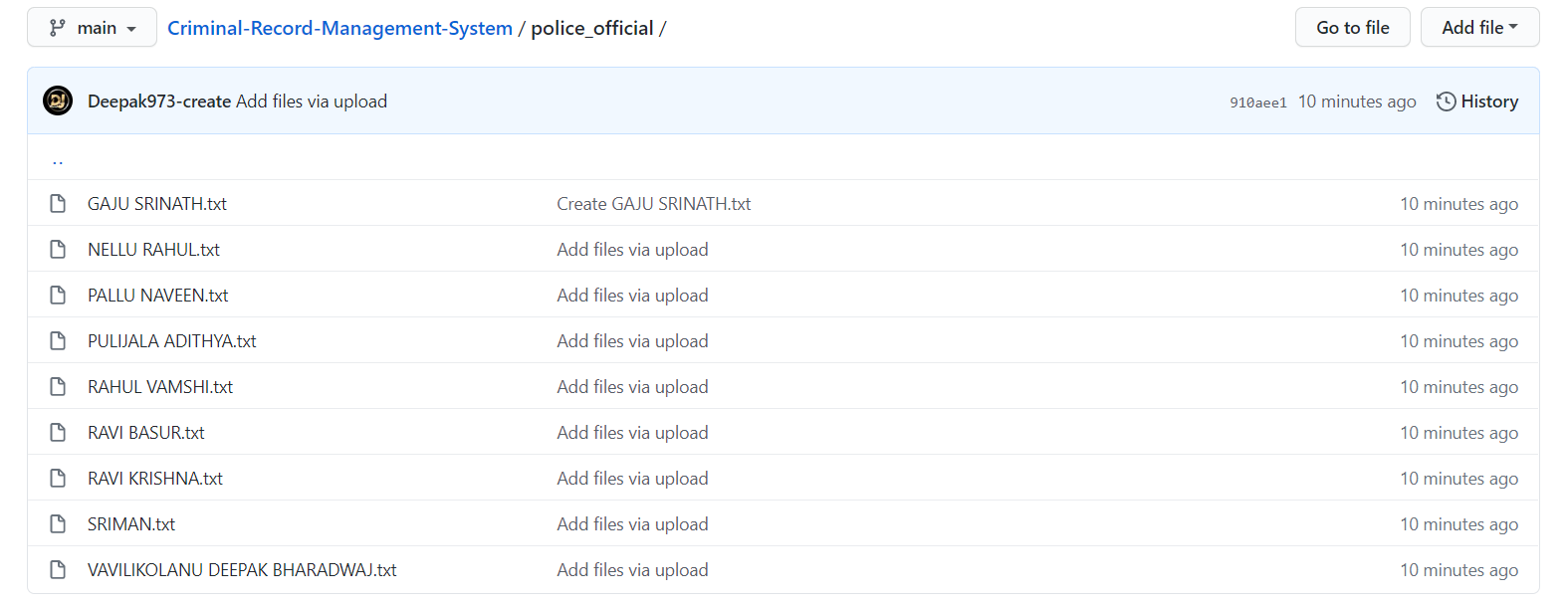


Within the complaints, we have

Within the criminal\_data, we have:



Within the police\_official, we have:



## TESTING

Testing is a method to check whether the actual product matches the expected requirements and to ensure that the product is defect-free. This process involves execution of various parts of the product either using manual or automated tools. The purpose is to identify errors, gaps or missing requirements in contrast to the actual requirements.

* + 1. **TEST PLAN**

We approached testing our console application by analysing each module separately. First, we coded the requirements and then manually tested each feature present in the module to cover any gaps that might occur.

* + 1. **USER TEST CASES**

The user has 3 major functionalities: User(public), Police Official and Admin. Below are the testcases which we have compiled together manually.

|  |
| --- |
| **USE CASE DESCRIPTIONS** |
| * Use Case ID : CB01 * Name : Register Compliant. * Actor : User(Public). * Description : To Register A Complaint. * Pre-Conditions : NA * Post-Conditions : A Complaint Is Registered And ID Is Given. |

|  |  |
| --- | --- |
| **User(Public)** | **System** |
| 1)If user is new then  First he/she should Register complaint. |  |
|  | 2)Asks to write a letter to register the complaint. |
| 3)Writes a complaint letter. |  |
|  | 4)Validates data and registers compliant. |

|  |
| --- |
| **USE CASE DESCRIPTIONS** |

|  |
| --- |
| * Use Case ID : CB02 * Name : Status Of Complaint. * Actor : User(Public). * Description : To Check Status Of Complaint. * Pre-Conditions : Enter The ID and Passcode Given. * Post-Conditions : The Status Of Compliant Is Registered. * Main Flow: |

|  |  |
| --- | --- |
| **User(Public)** | **System** |
| 1)Choose Status of Complaint Option. |  |
|  | 2)Validates Data And Displays Data If The Entered ID And Passcode Is Right |
|  | 3)If No Redirects To Menu. |

|  |
| --- |
| **USE CASE DESCRIPTIONS** |
| * Use Case ID : CB03 * Name : Withdraw Of Complaint. * Actor : User(Public). * Description : Withdrawal Of Complaint. * Pre-Conditions : Enter The ID and Passcode Given. * Post-Conditions : Create a withdrawn.txt file and user entered reason   is written into it |

|  |  |
| --- | --- |
| **User(Public)** | **System** |
| 1)Choose Withdrawal of complaint Option. |  |
| 2) Enter the Reason of the Withdrawal |  |
|  | 3)Validates Data and redirects to menu. |

|  |
| --- |
| **USE CASE DESCRIPTIONS** |
| * Use Case ID : CB04 * Name : Police Compliant Search. * Actor : Police Official. * Description : Complaint Search For Investigation. * Pre-Conditions : Enter There Official ID And Complaint ID By Selecting Type of Complaints. * Post-Conditions : The Status Of Compliant Is Registered. |

|  |  |
| --- | --- |
| **Police Official** | **System** |
| 1)Selects The Search the Complaint Option. |  |
|  | 2) Asks the name of the user |
| 3) Enters the name of User |  |
|  | 4) Checks the complaint is registered or not  If Yes Displays Data Or else show the warning message redirects to menu. |

|  |
| --- |
| **USE CASE DESCRIPTIONS** |
| * Use Case ID : CB05 * Name : Criminal search. * Actor : Police Official. * Description : By type of Complaint search the criminal. * Pre-Conditions : Enter The ID and Passcode Given and enter type of complaint. * Post-Conditions : List of Criminals with their data like Address Proof, for past 6 months. |

|  |  |
| --- | --- |
| **Police Official** | **System** |
| 1)Choose type of crime and criminal list Option. |  |
|  | 2)Validates Data And Displays Data if the type of data is there |
|  | 3) If No Redirects To Main. |

|  |
| --- |
| **USE CASE DESCRIPTIONS** |
| * Use Case ID : CB06 * Name : Higher Authority Status Check. * Actor : Admin. * Description: Complaints solved And Pending * Pre-Conditions : Enter There ID And Passcode. * Post-Conditions : Displays The Data Of Solved And Pending Cases And Gives Message Delivery Option. |

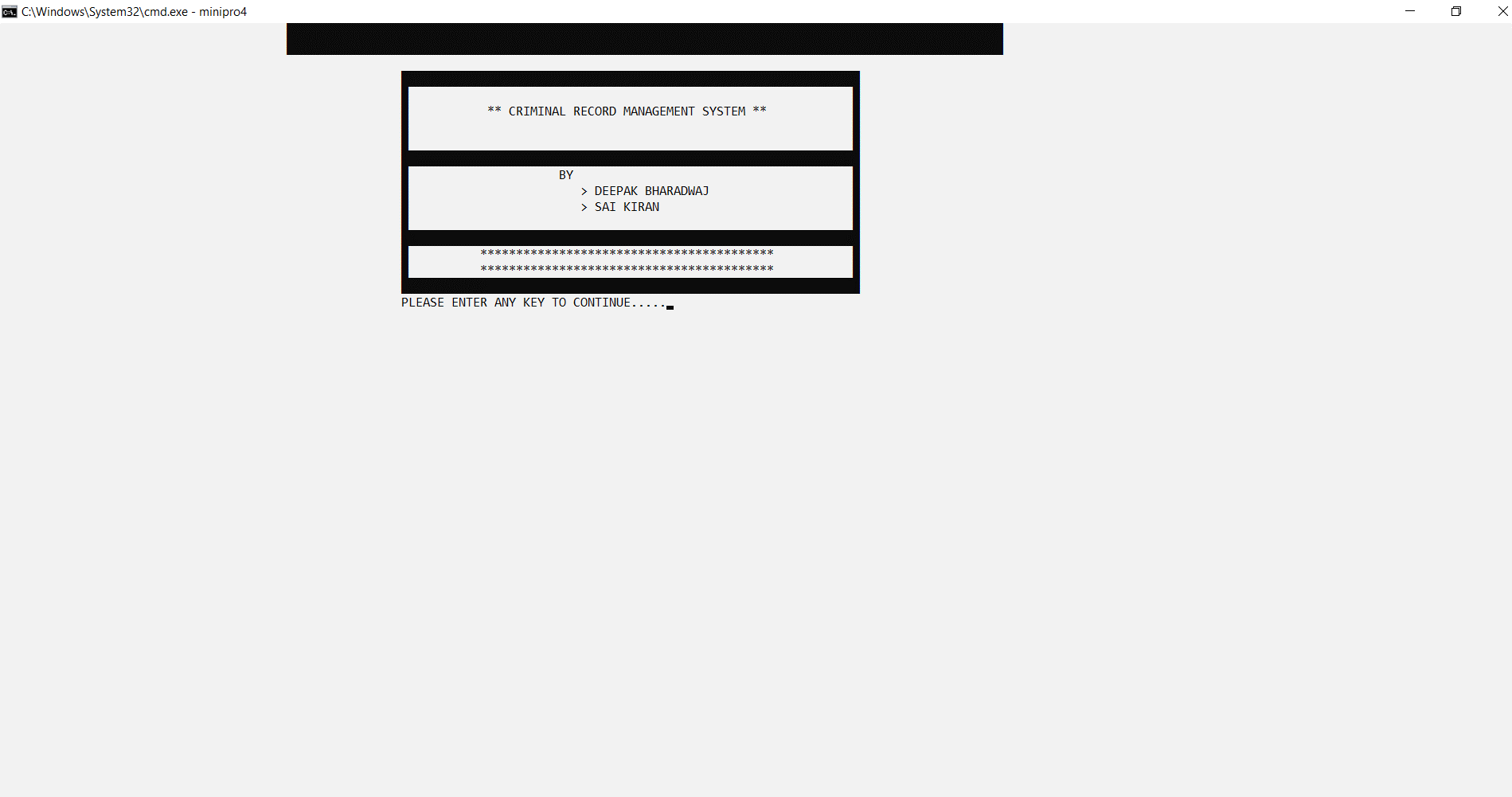
|  |  |
| --- | --- |
| **Admin** | **System** |
| 1)Enter The ID And Passcode |  |
|  | 2)Displays The Pending And Solved Cases. |
|  | 3)And Gives Message Option |

# RESULTS

We have successfully developed a platform for users of all age groups to utilize and to be benefited from this program. And also we made this platform for both the police official and admin to co-ordinate the registered complaints and the police officials respectively. Below are the output screenshots of the testcases mentioned.

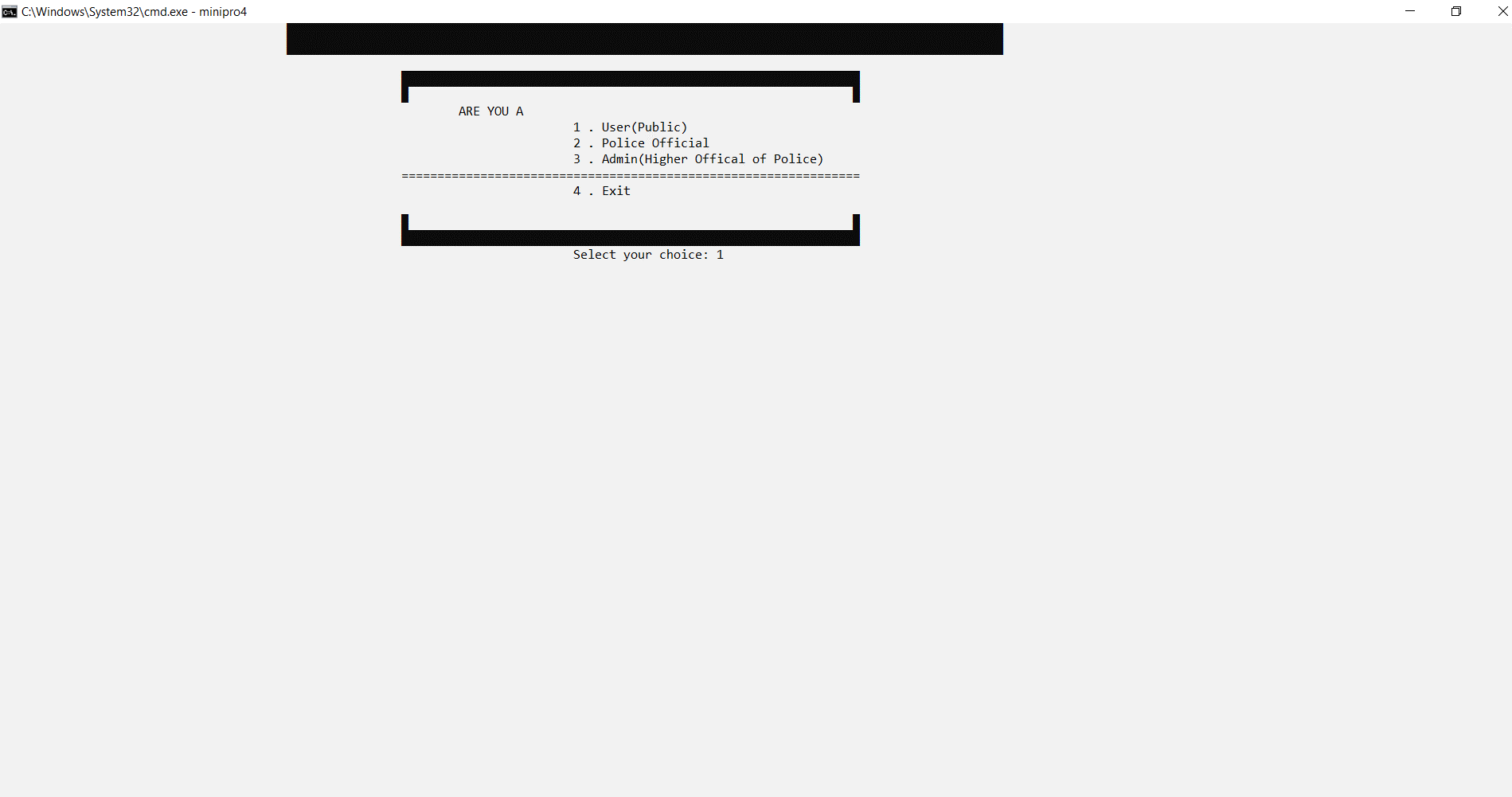
## USER TEST CASE RESULTS

**Test Case 1:**



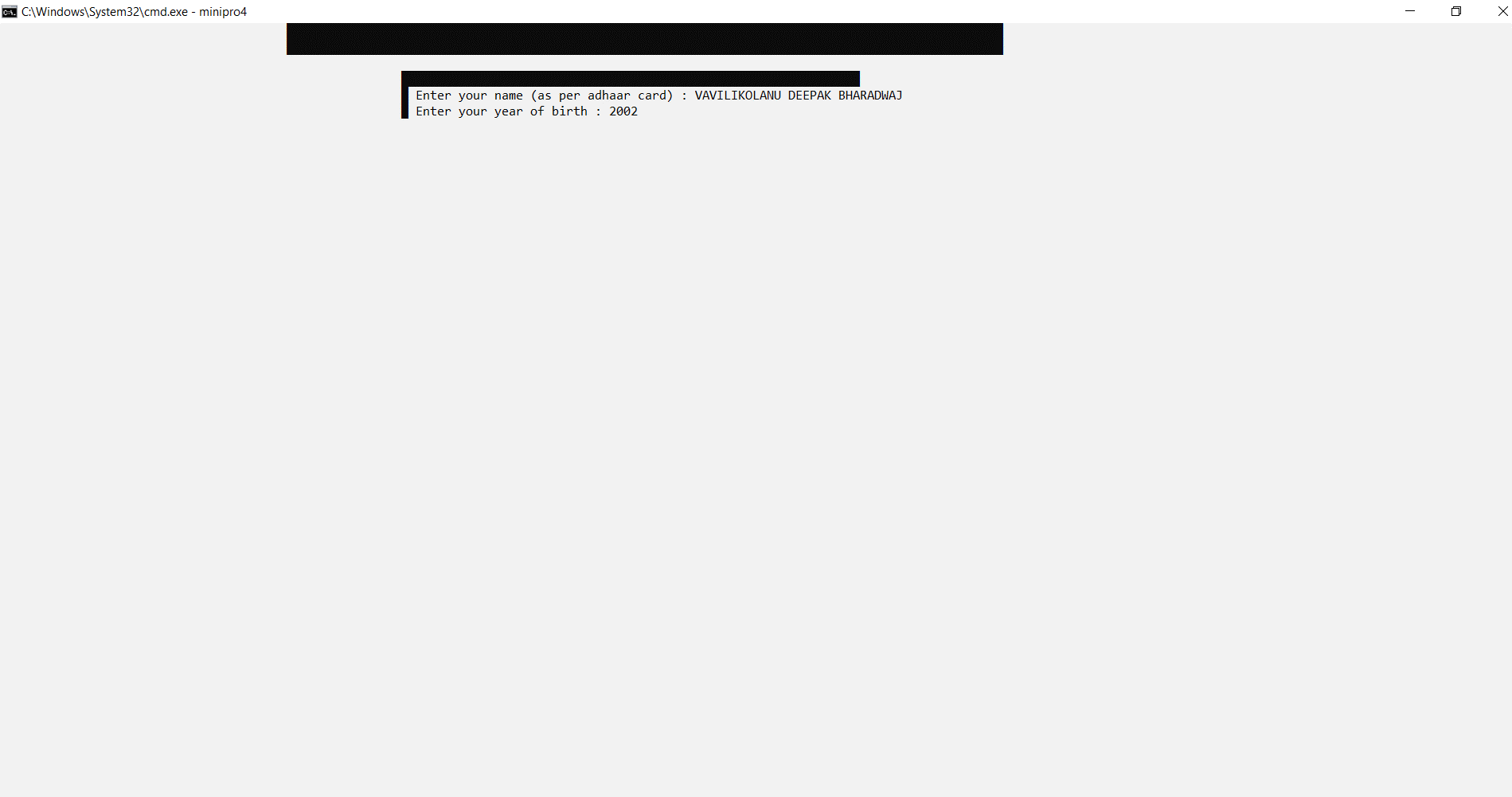
**Test Case 2:**

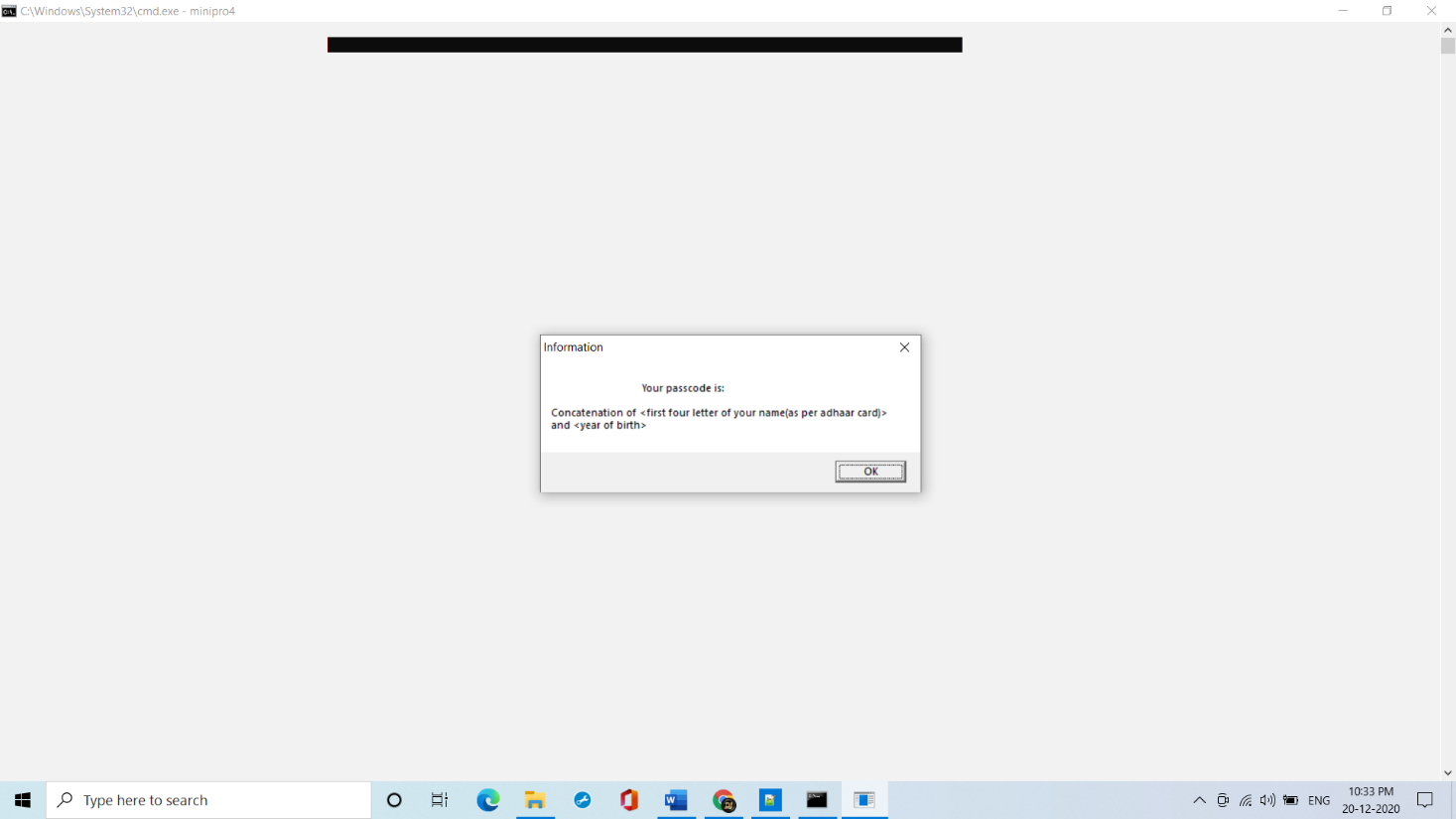
Main Menu of the program

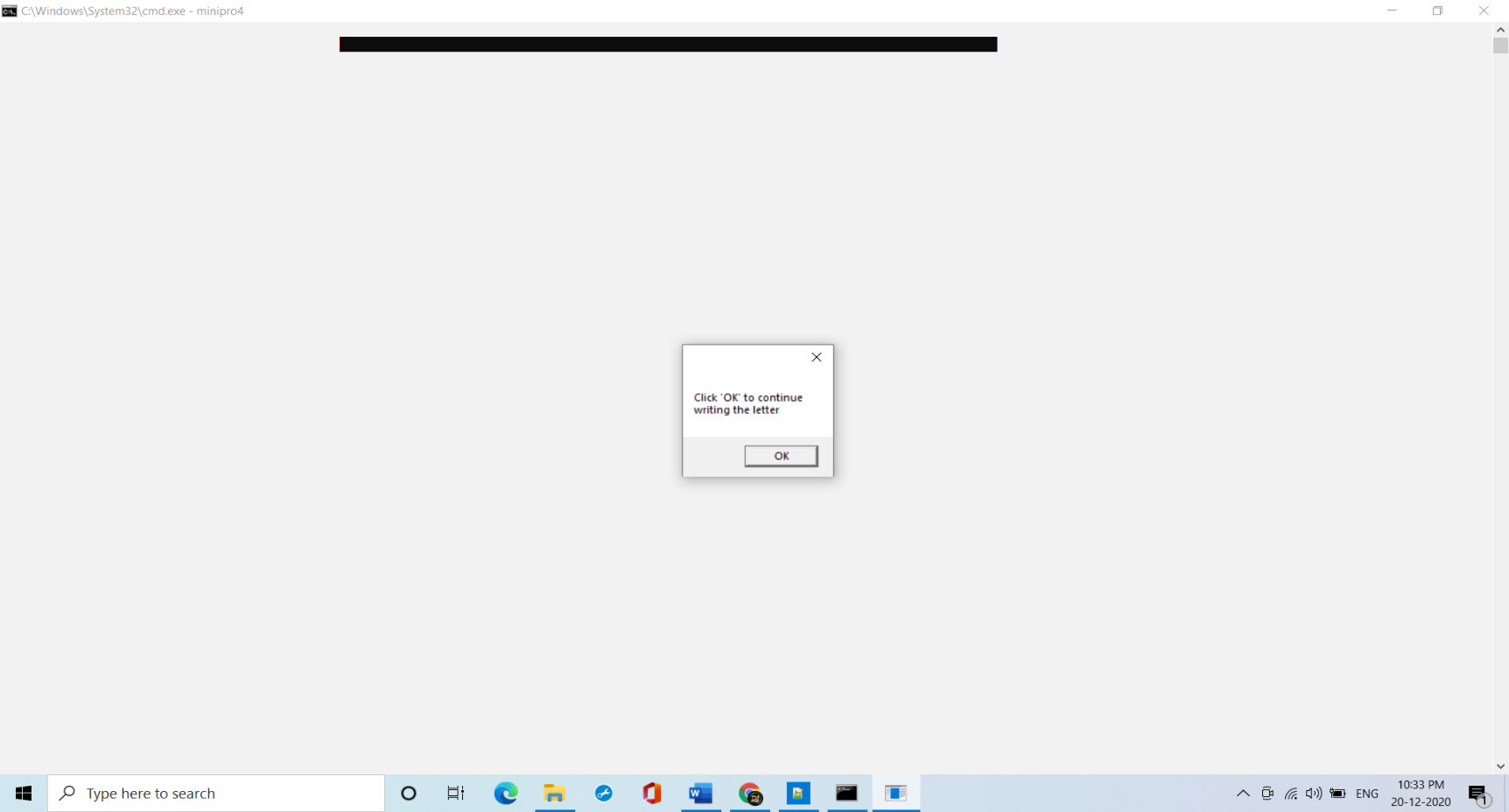


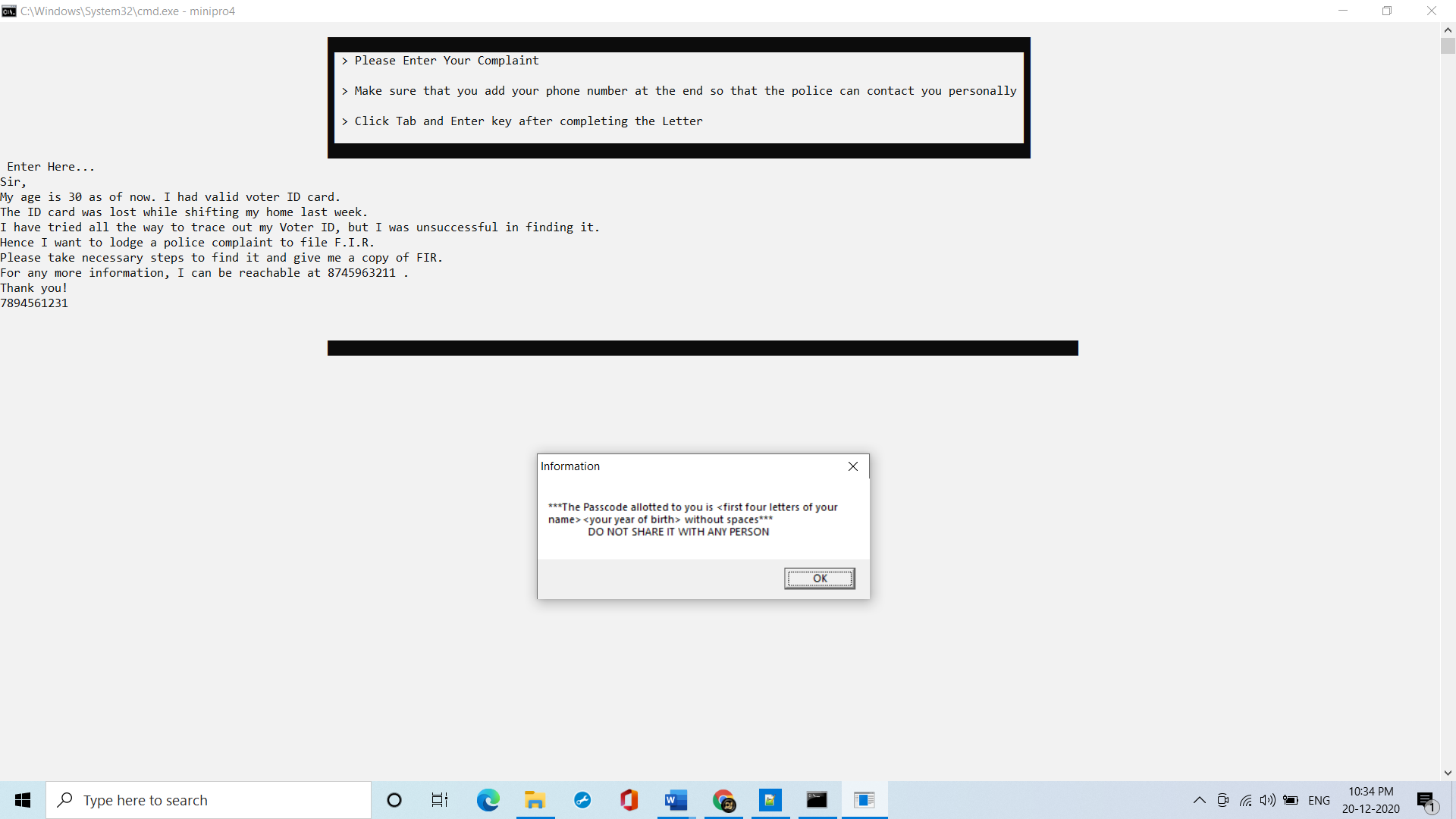
**Test Case 3:**

User registration (new user) of complaint



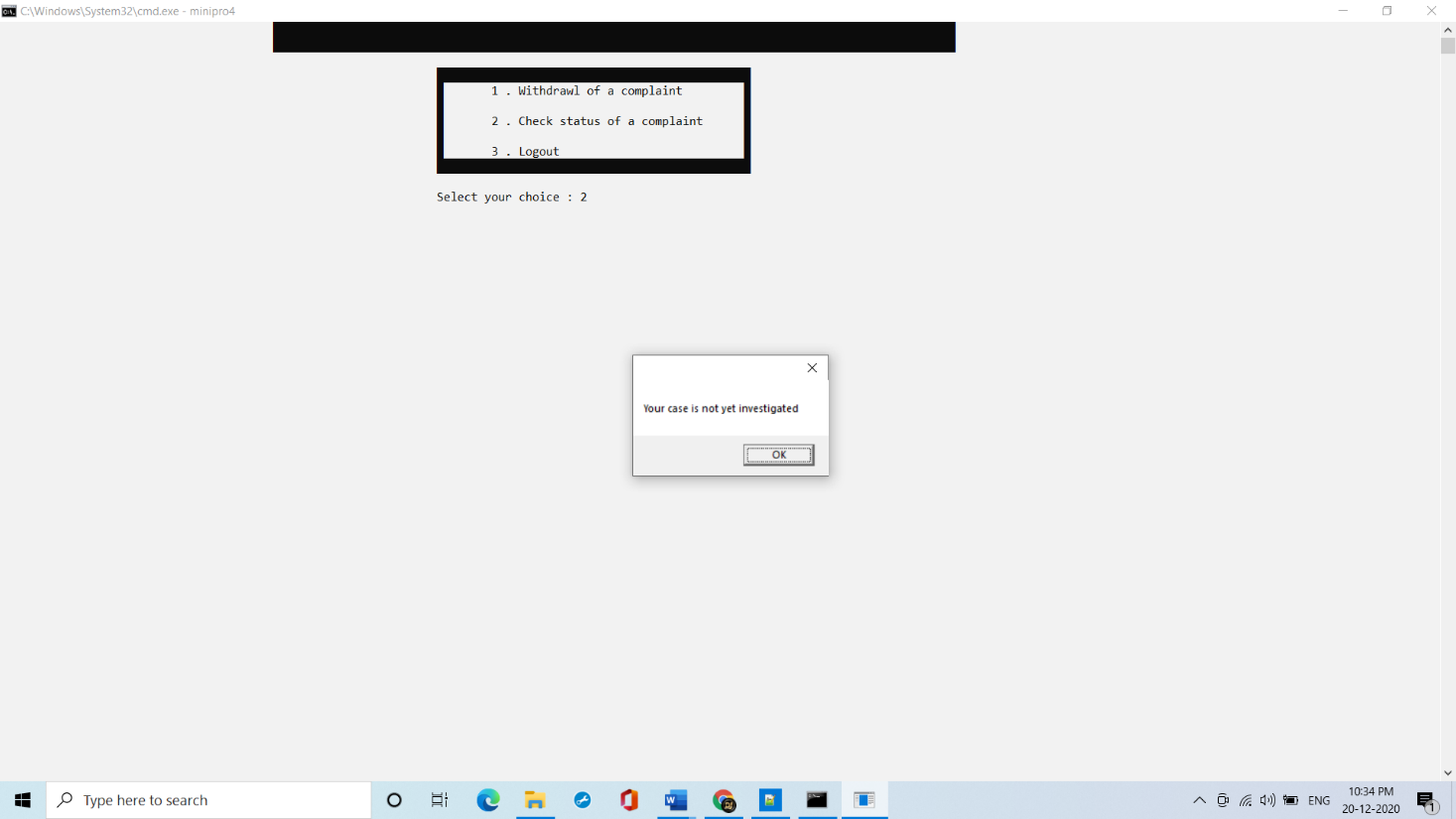




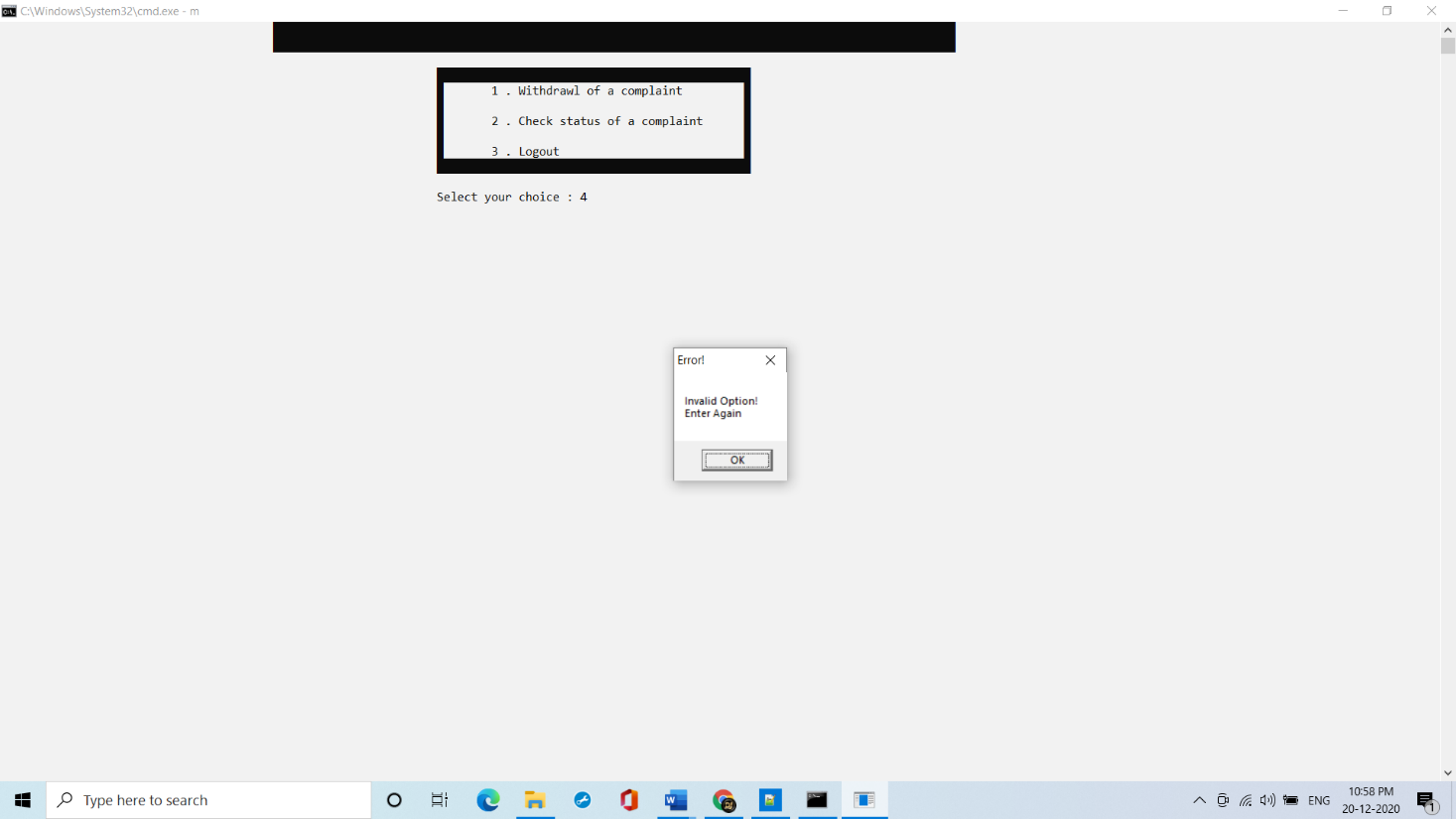


**Test Case 4:**

Checking the status of the complaint which was registered

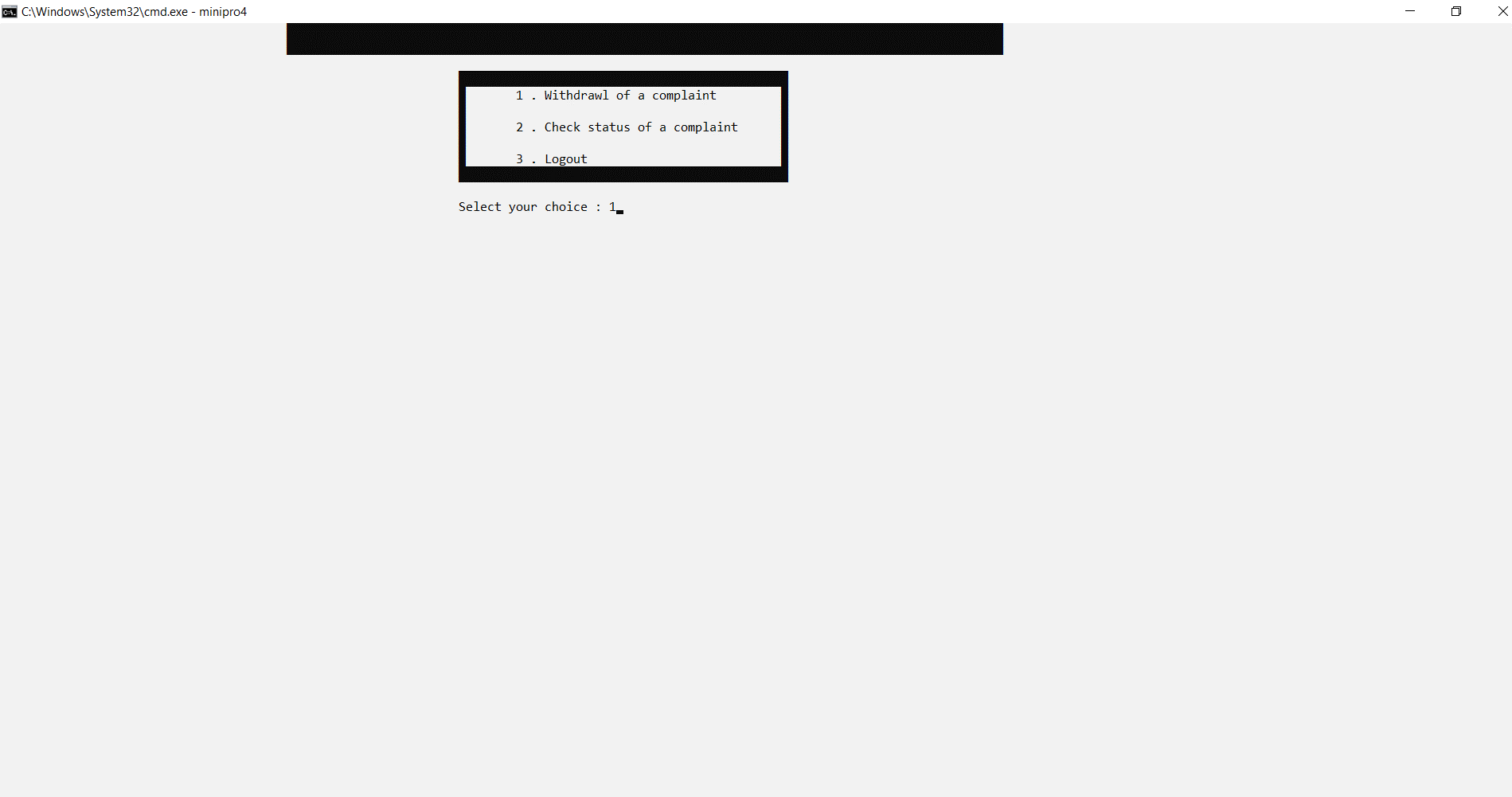


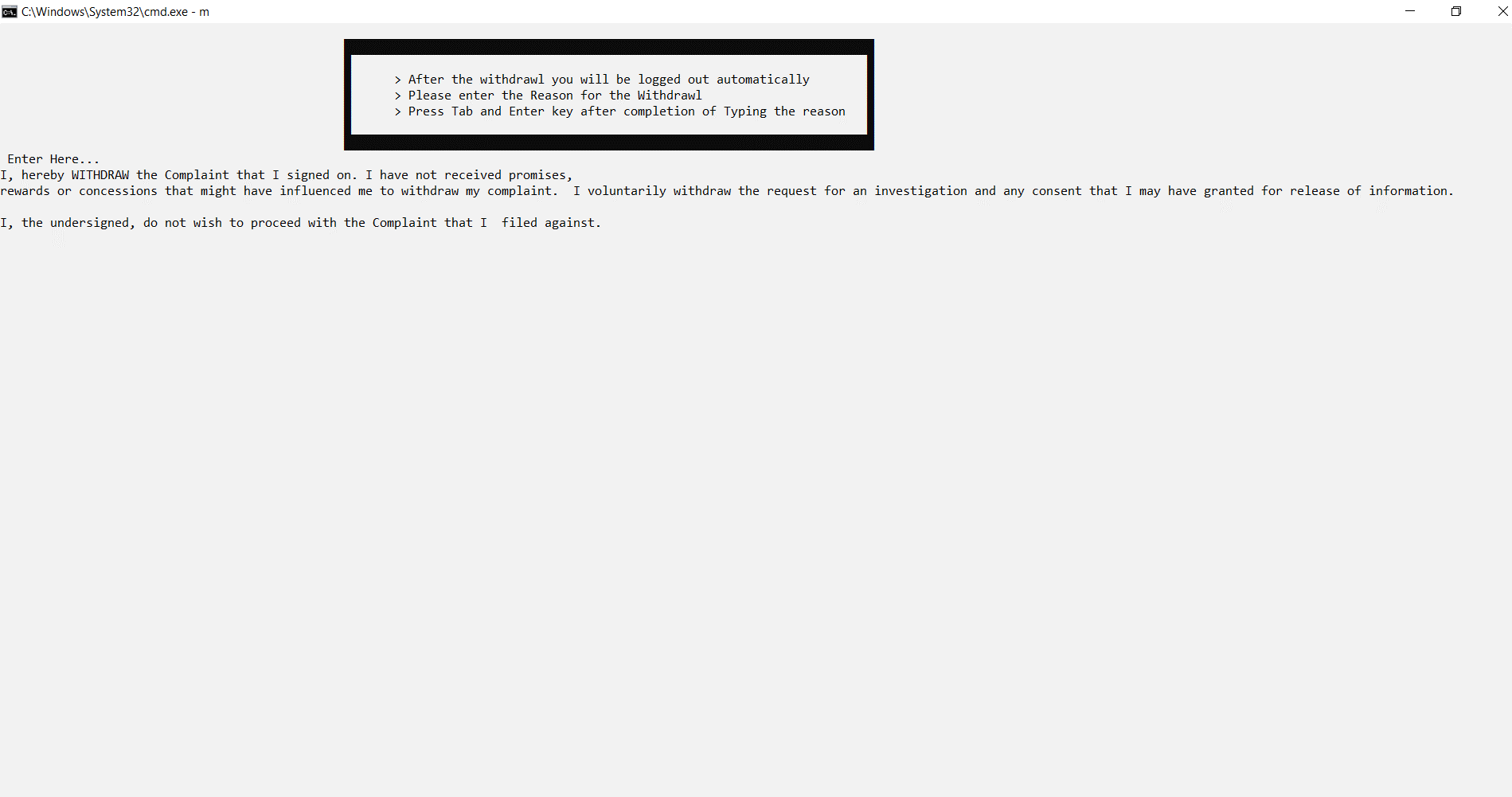
**Test Case 5:**



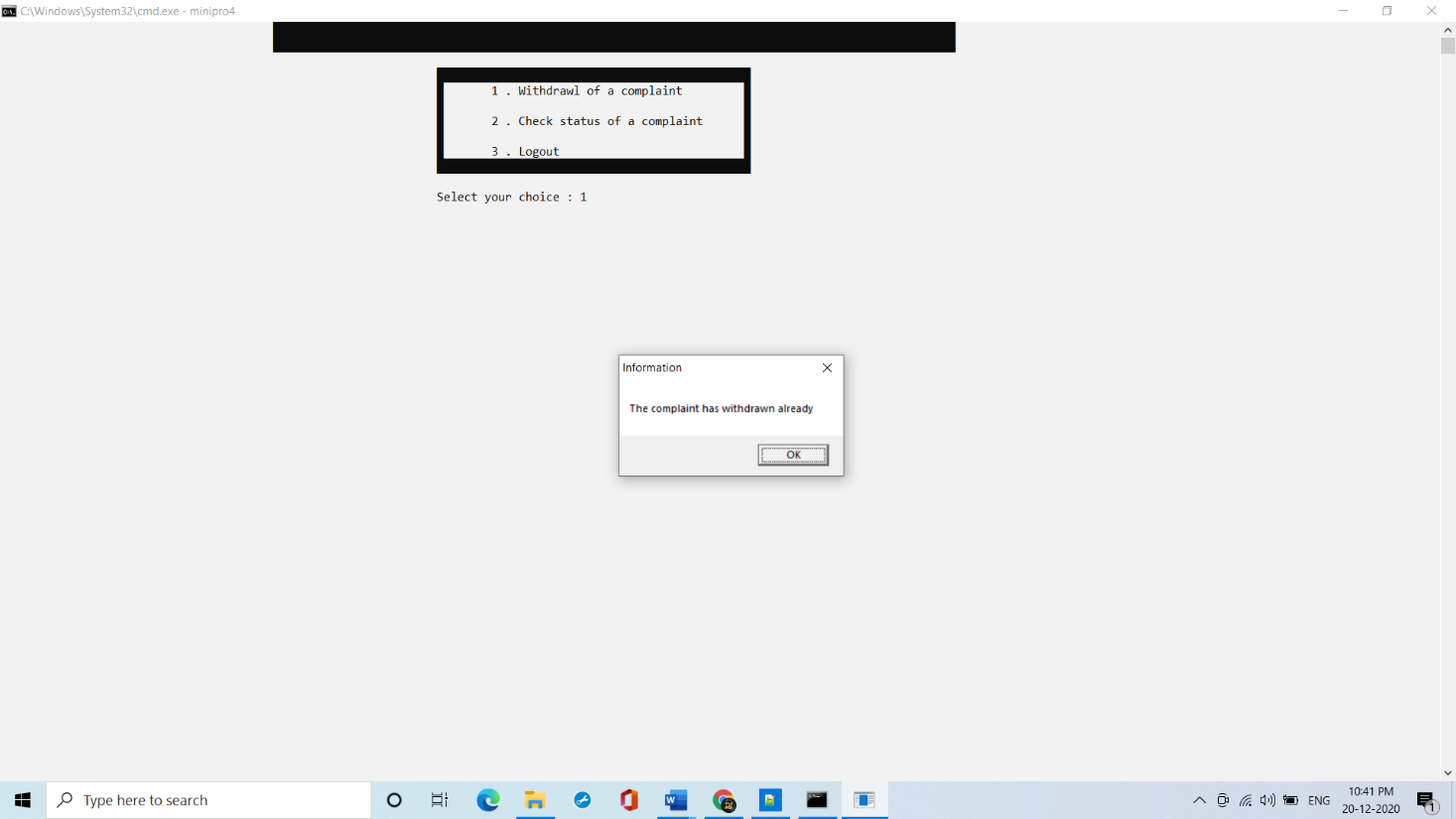
**Test Case 6:**

Withdrawing of the complaint

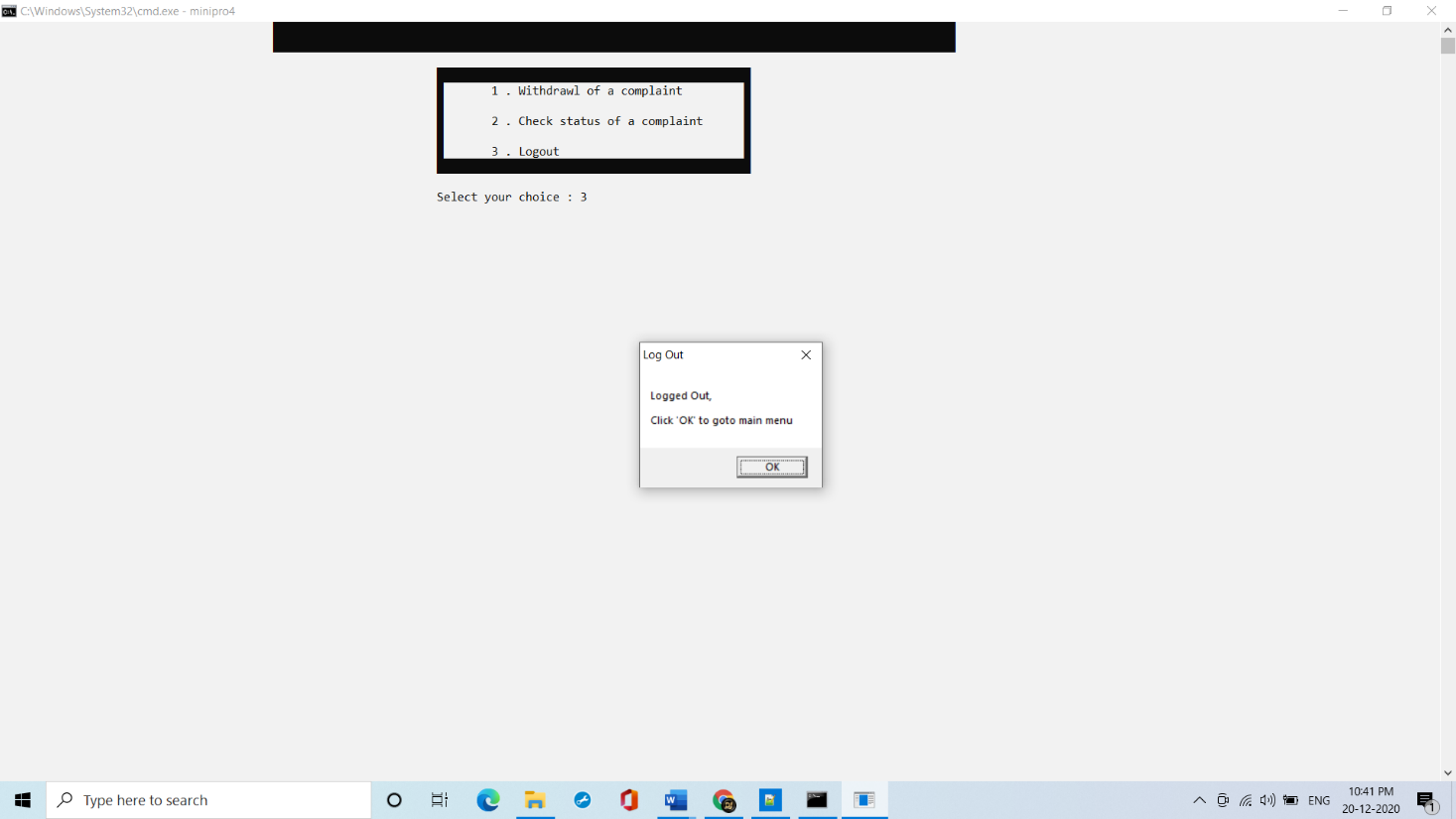




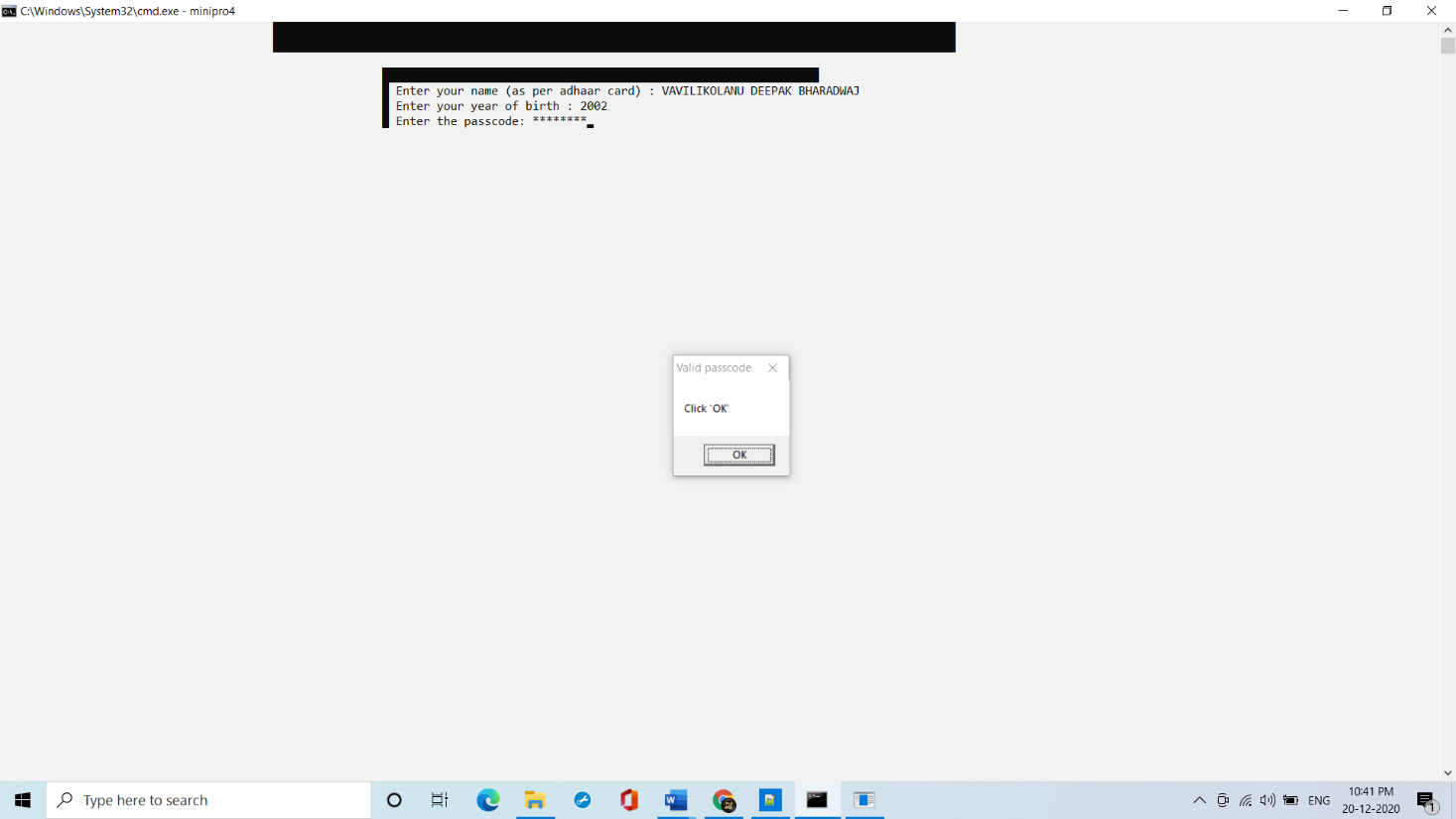
**Test Case 7:**



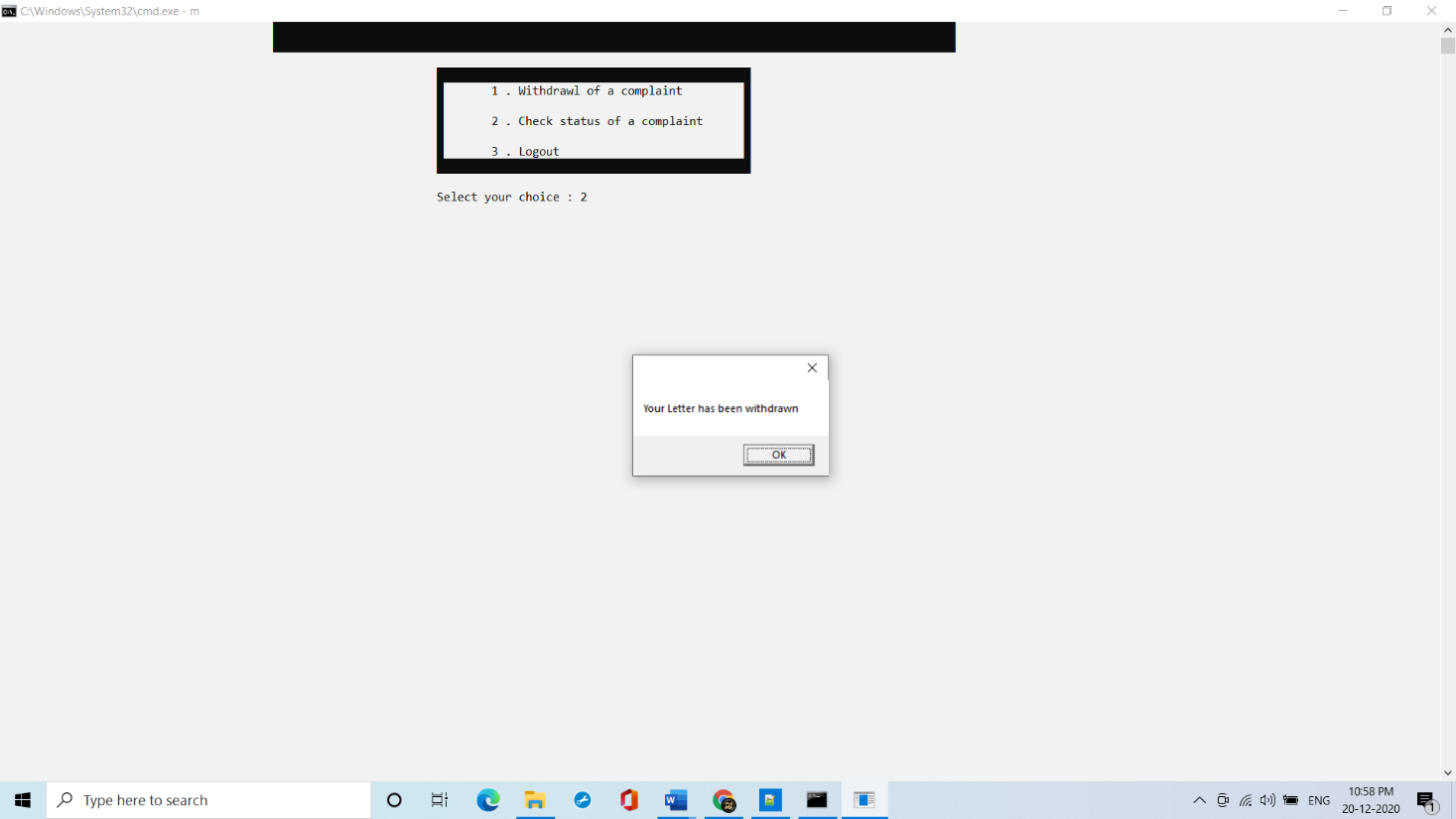
**Test Case 8:**



**Test Case 9:**

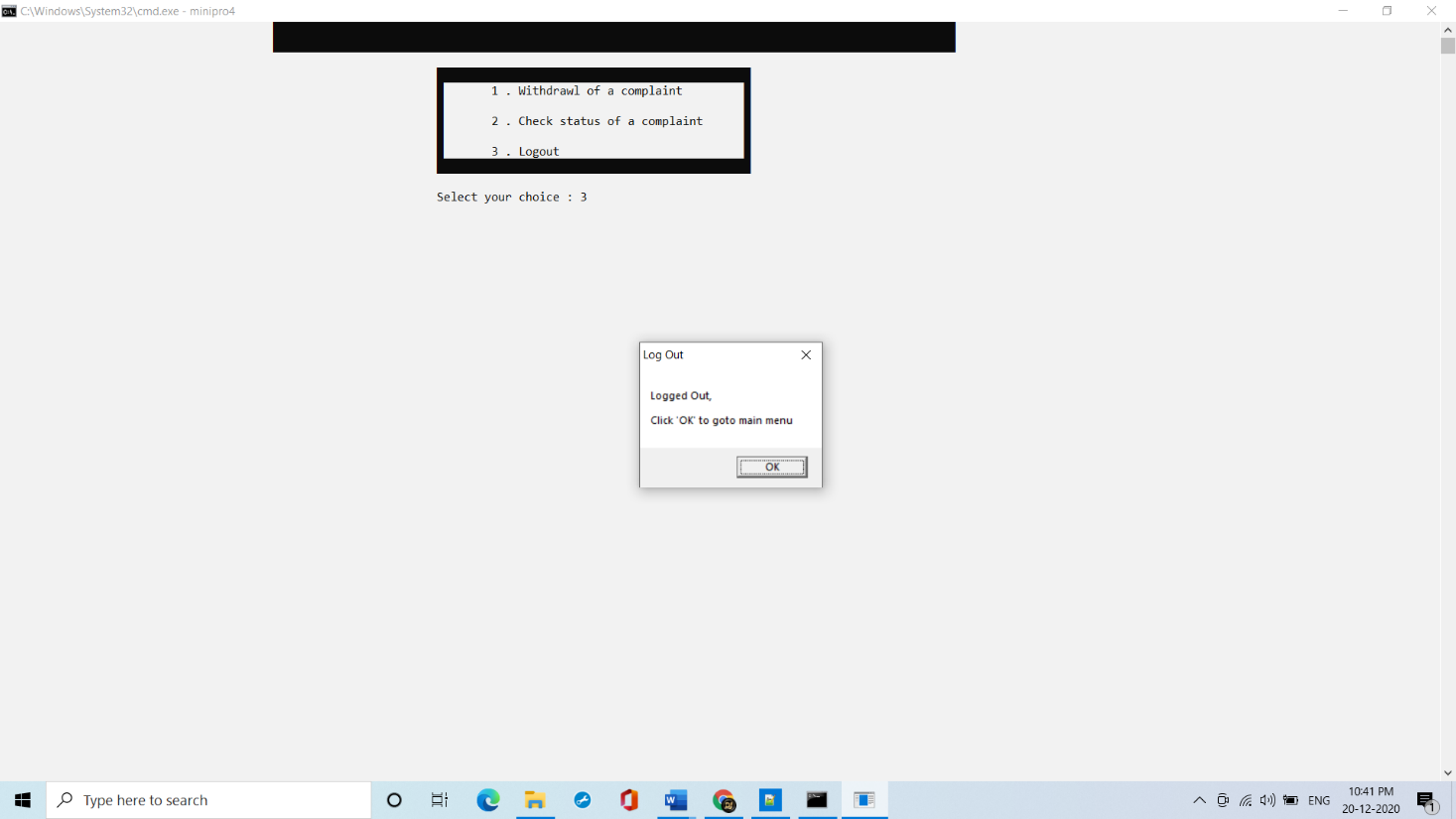


**Test Case 10:**



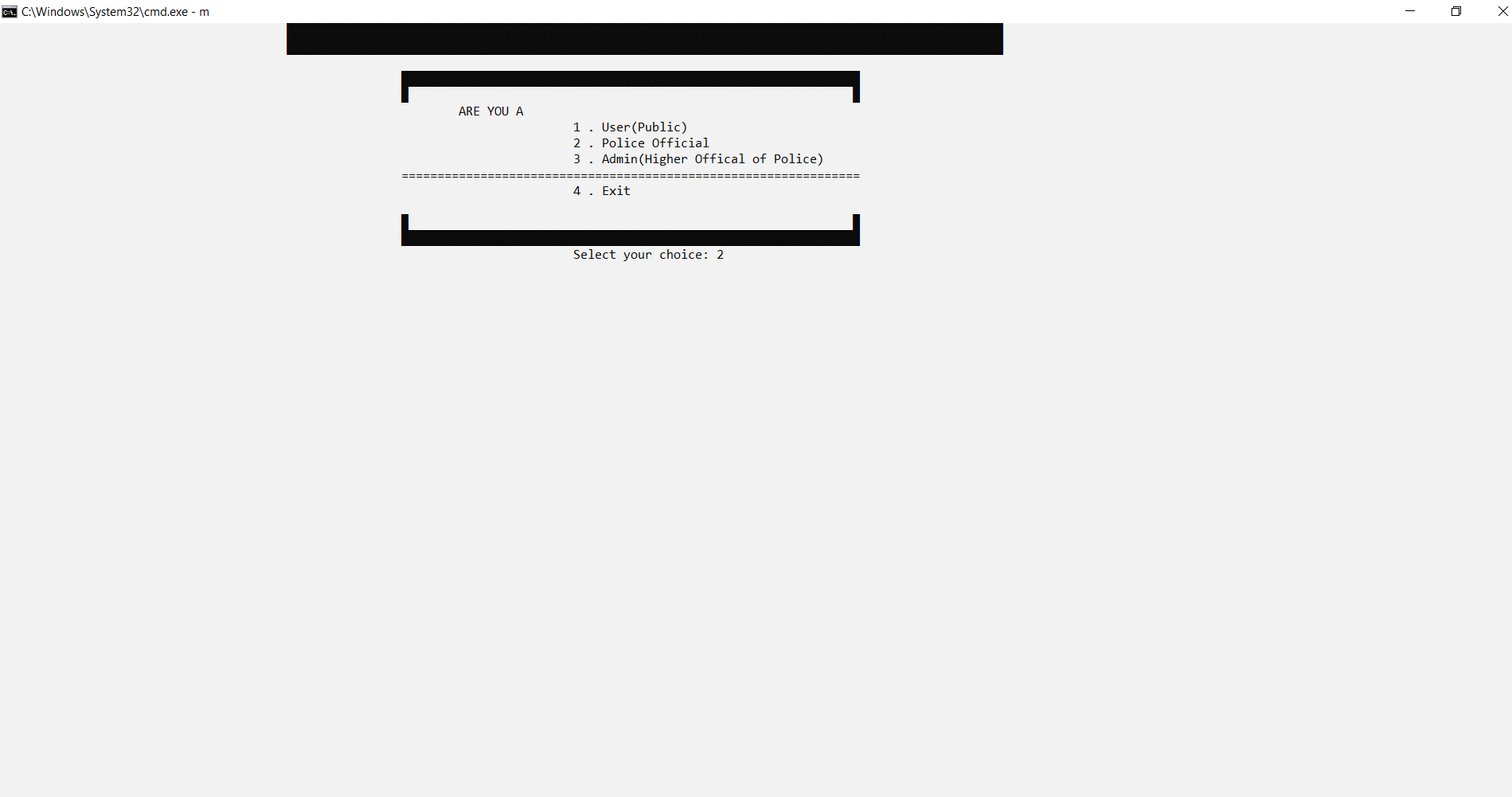
**Test Case 11:**

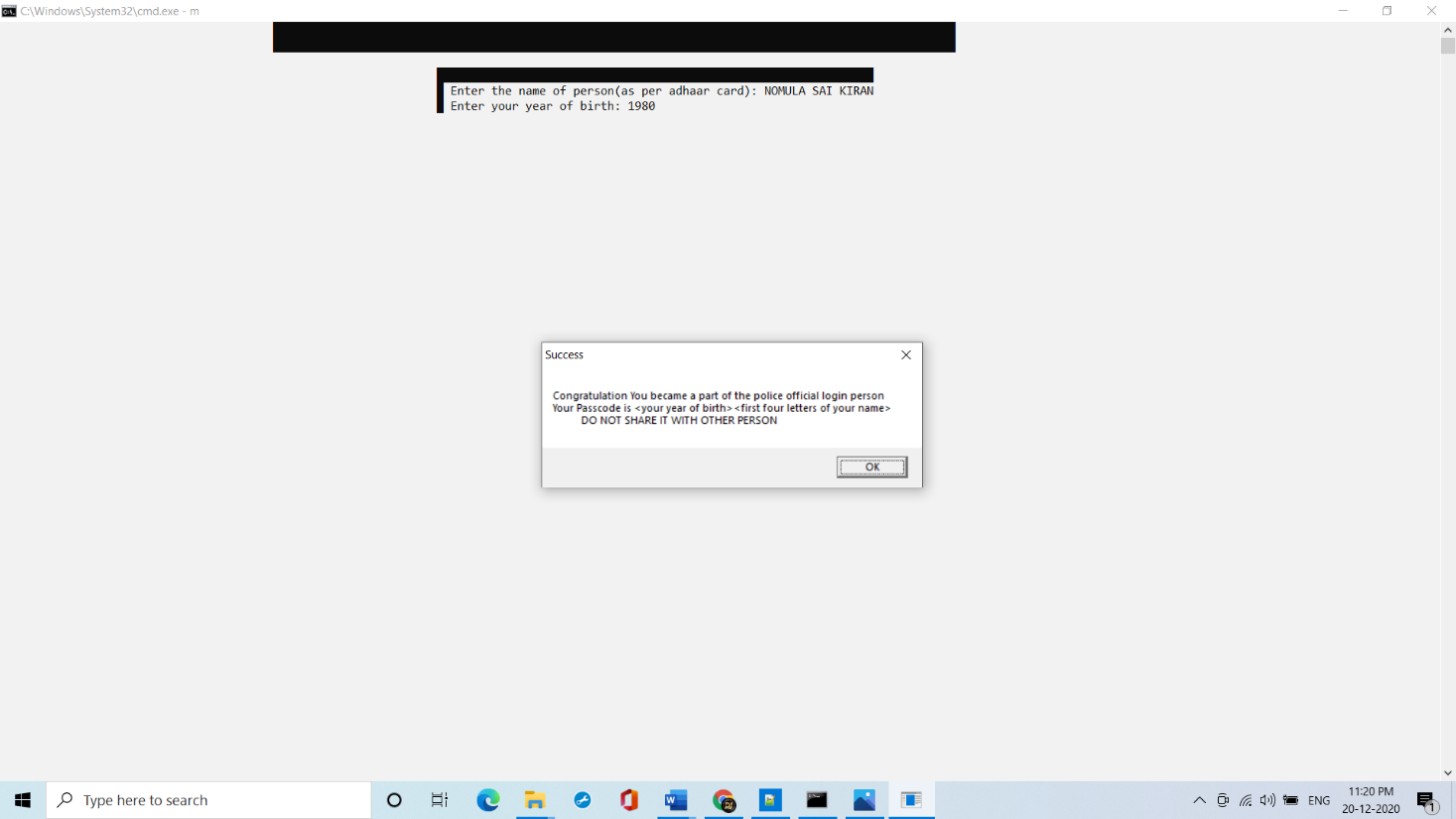
Logging out

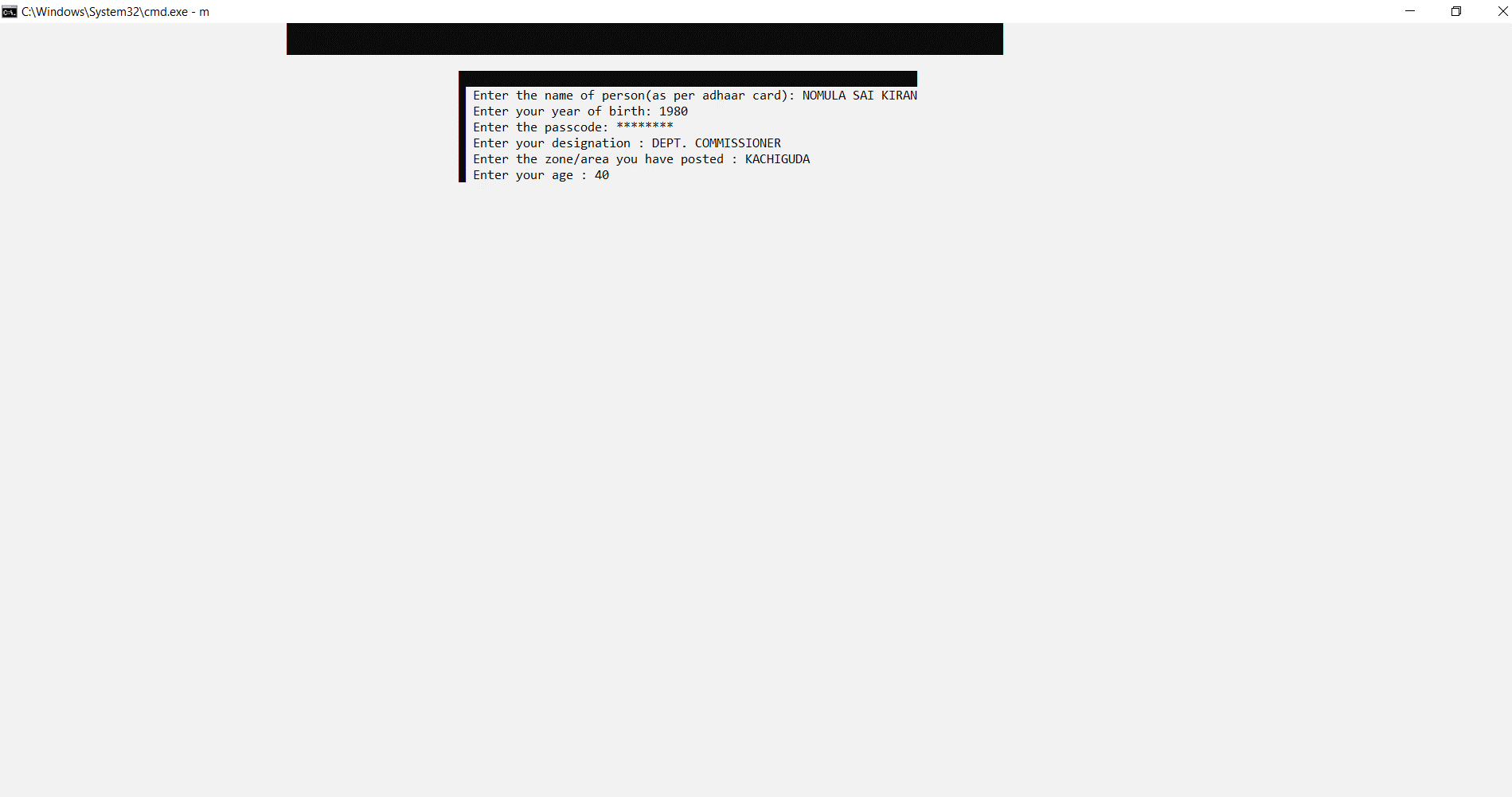


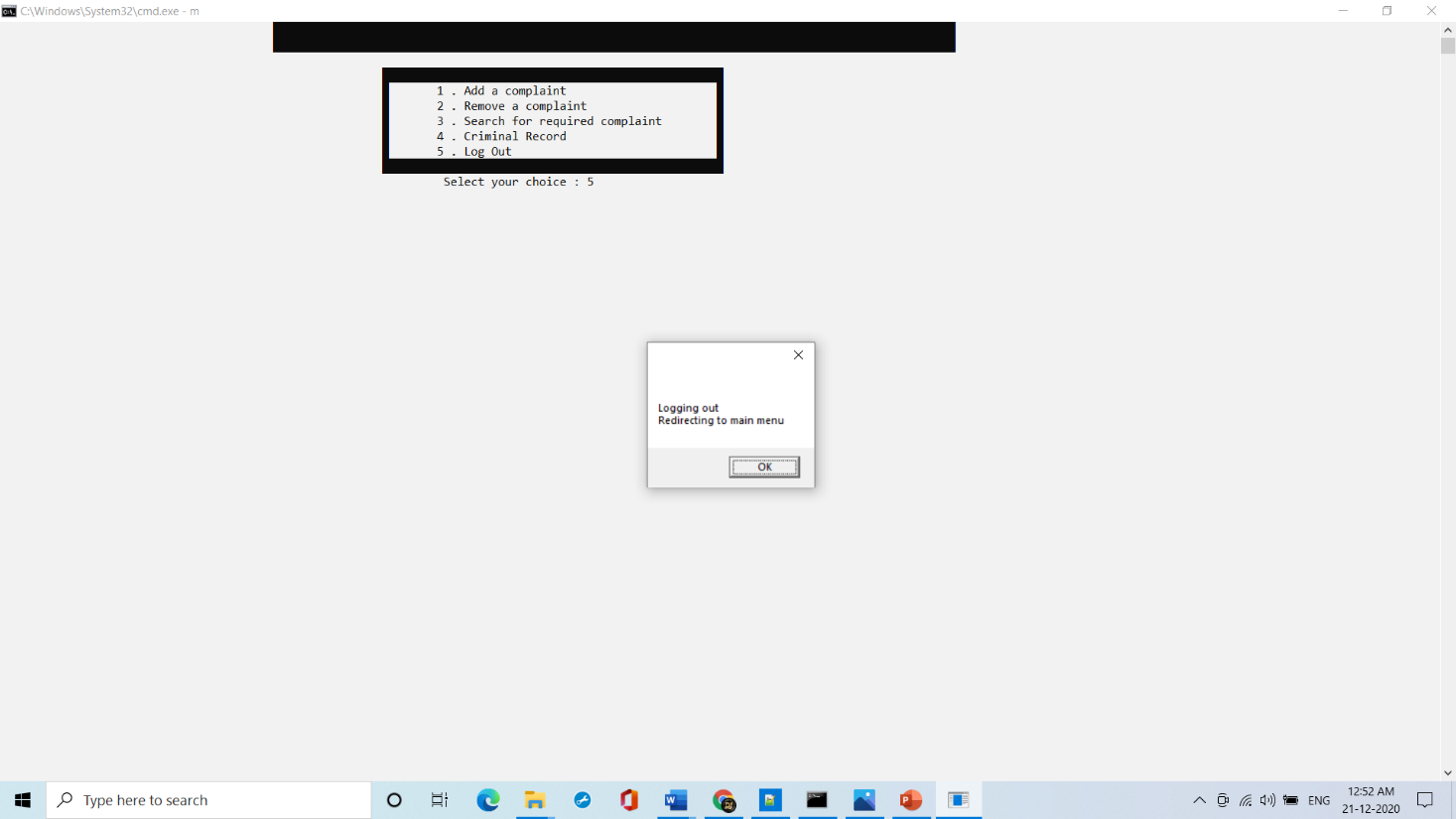
**4.2 Police Test Cases Results**

**Test Case 12:**

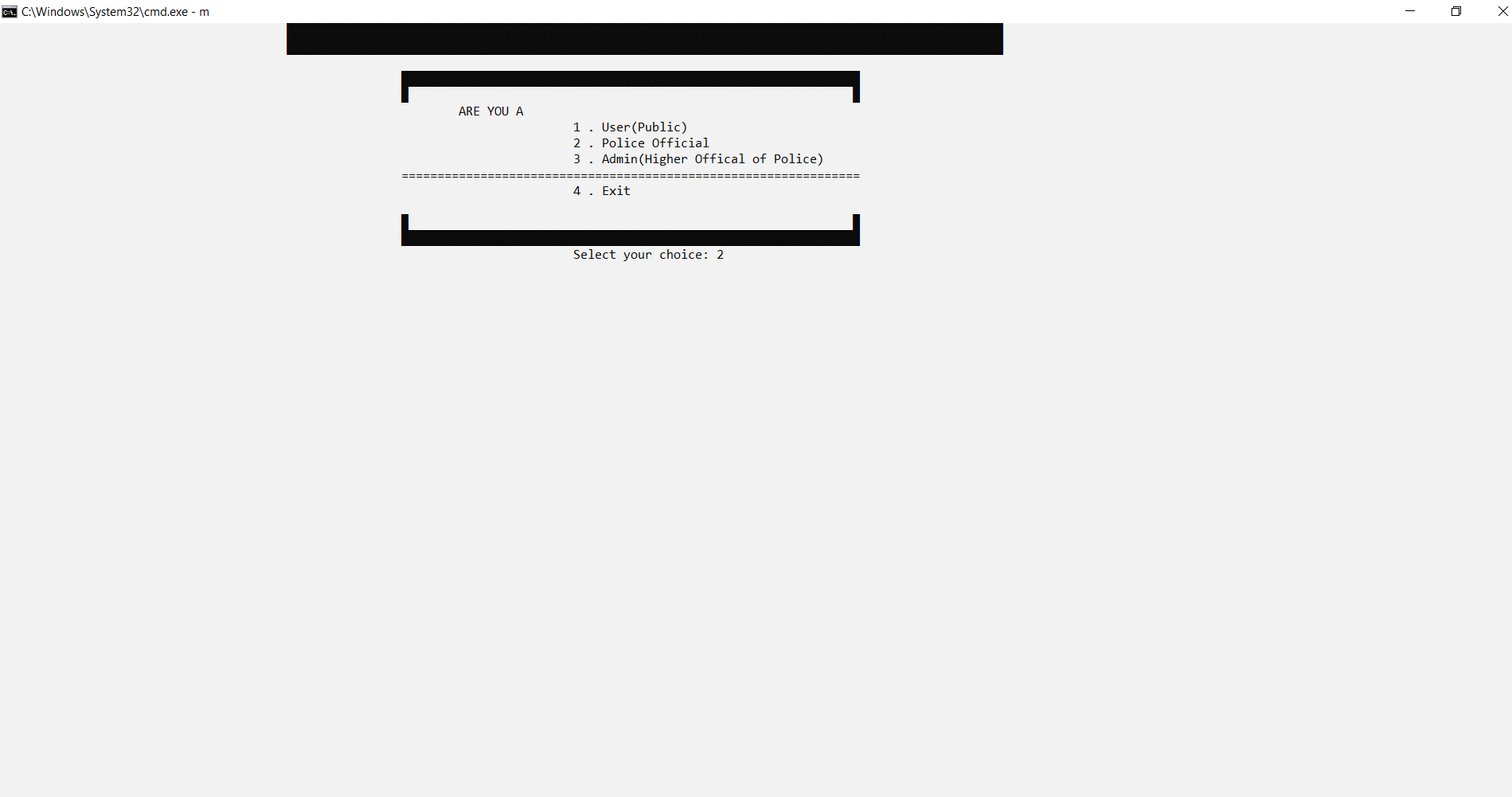


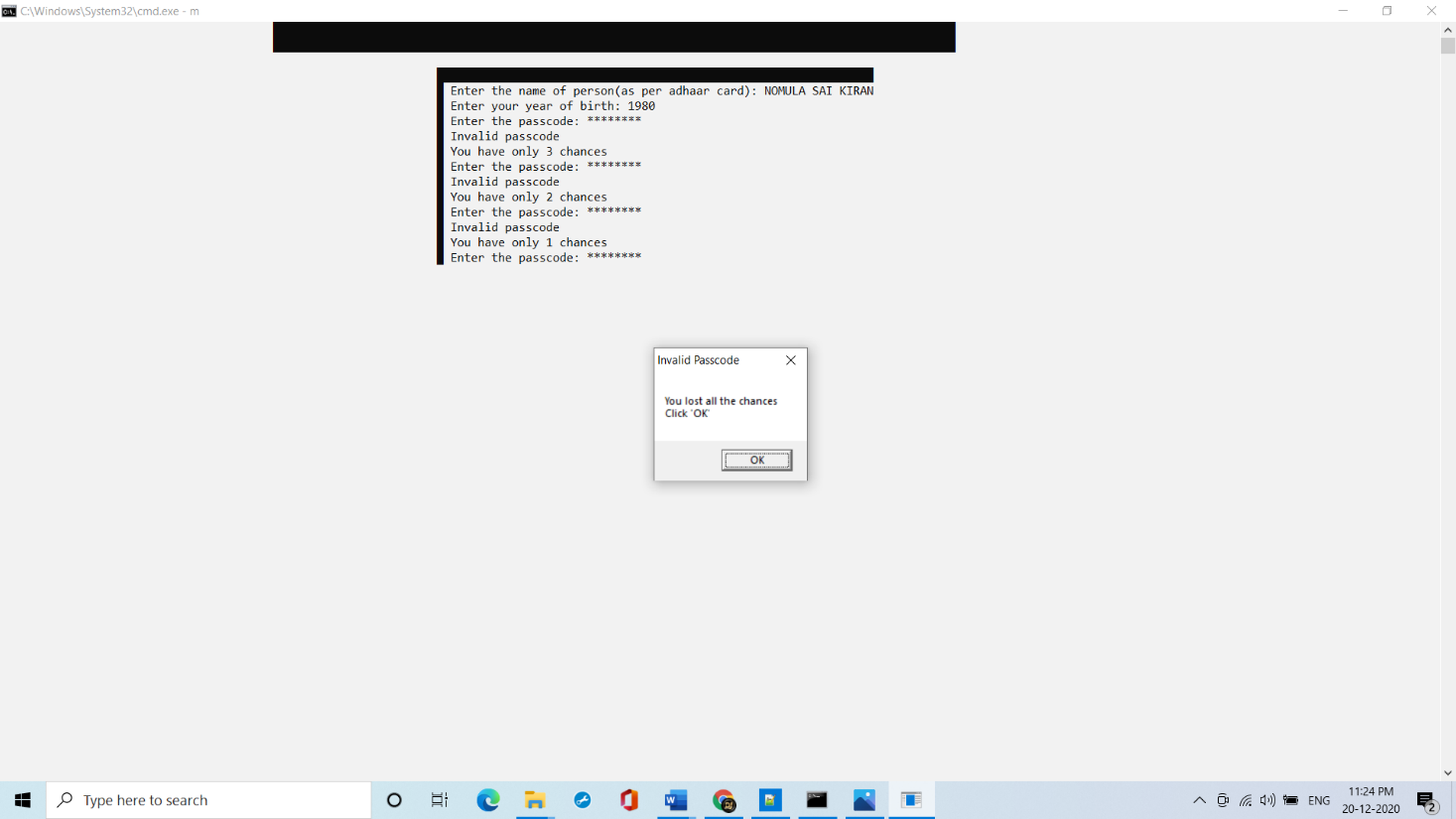




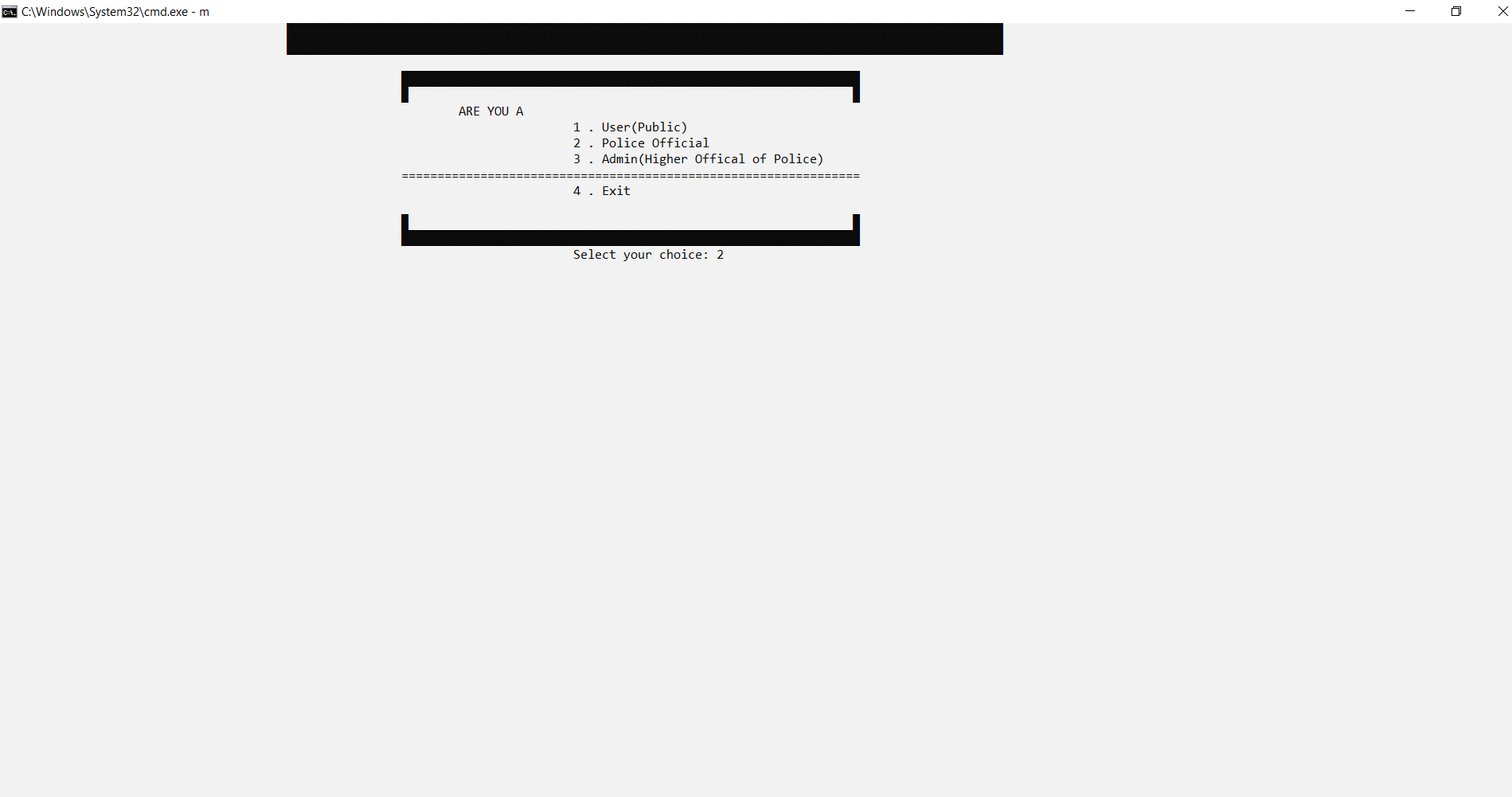


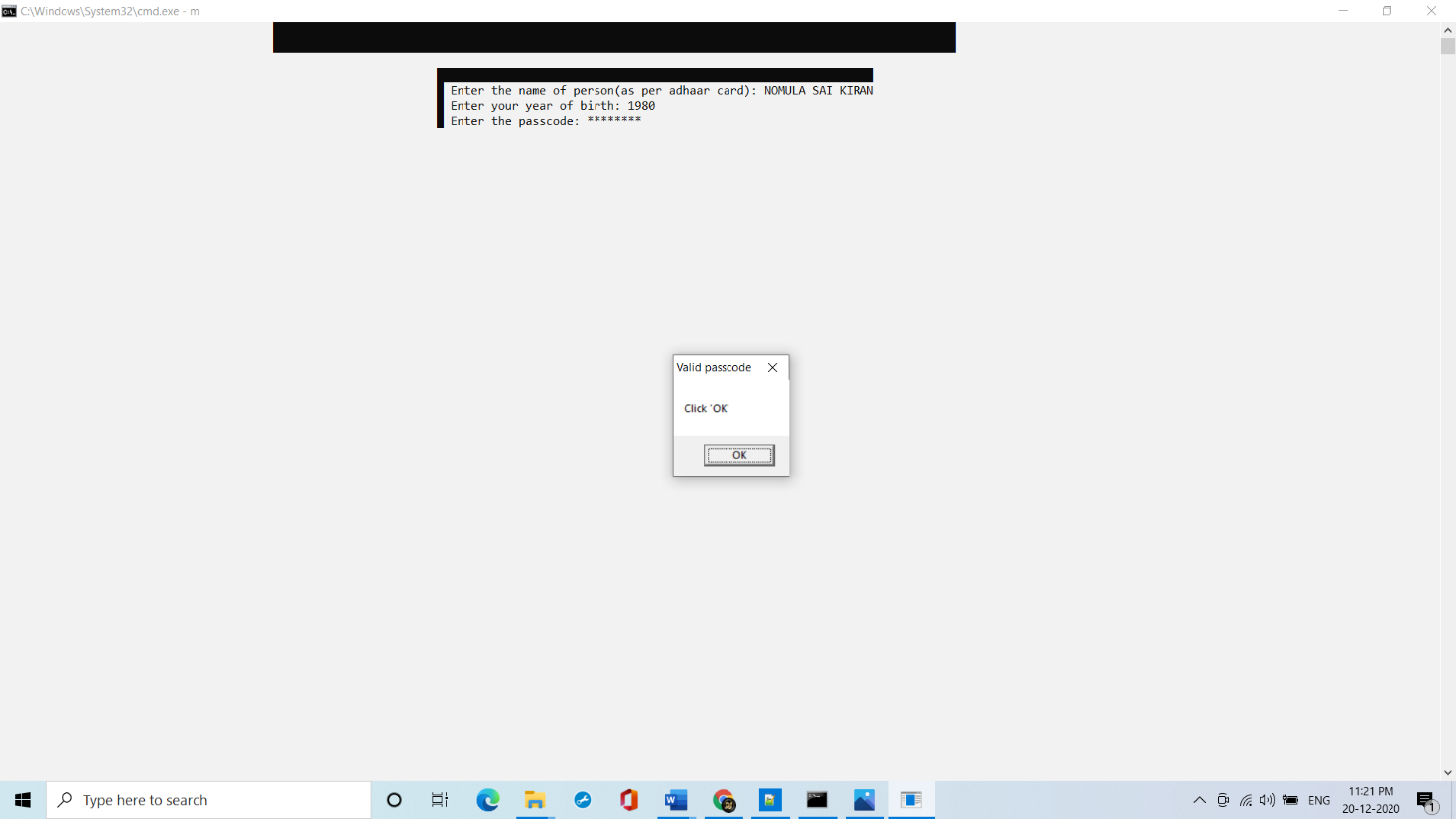
**Test Case 13:**





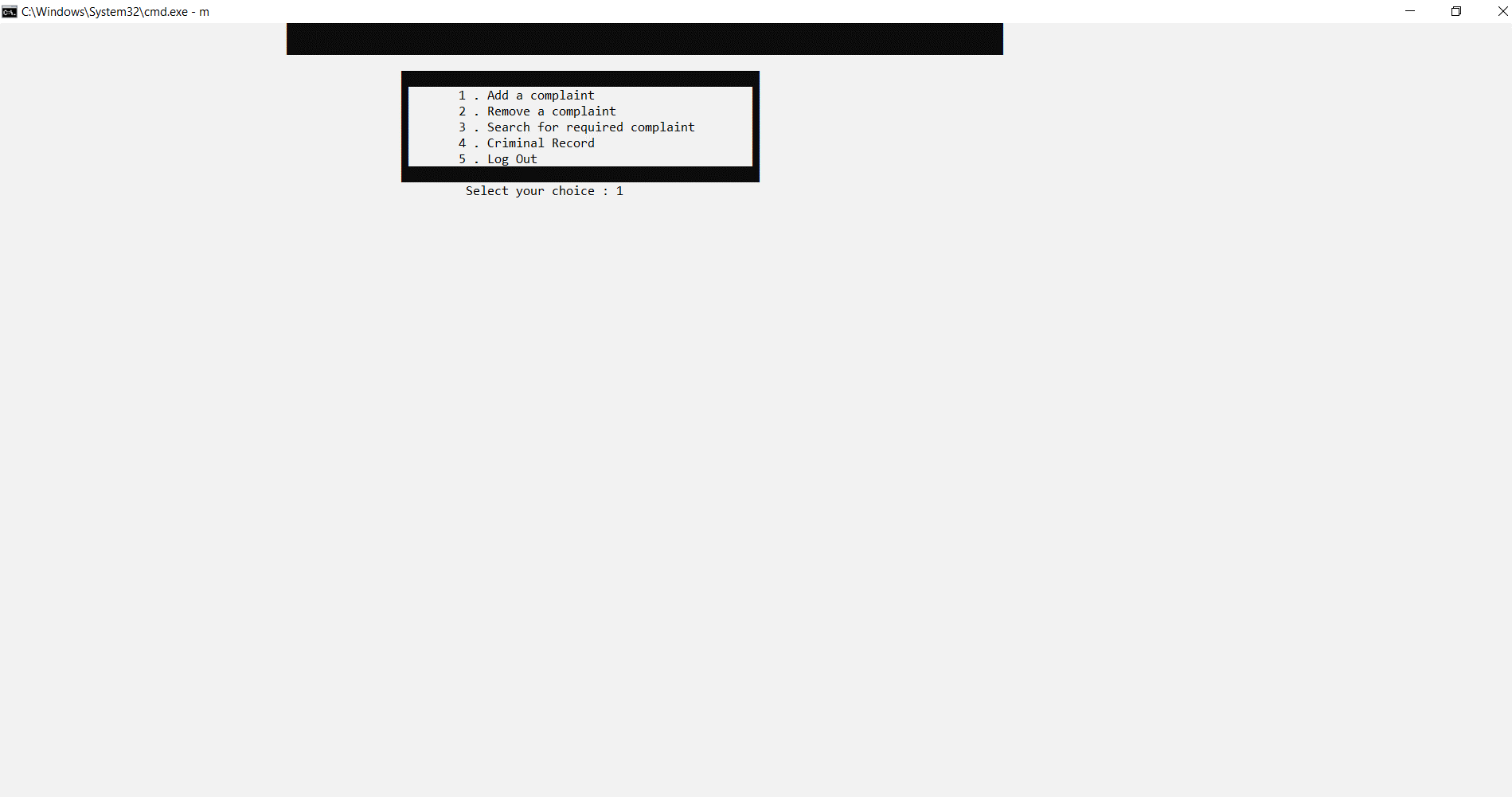
**Test Case 14:**

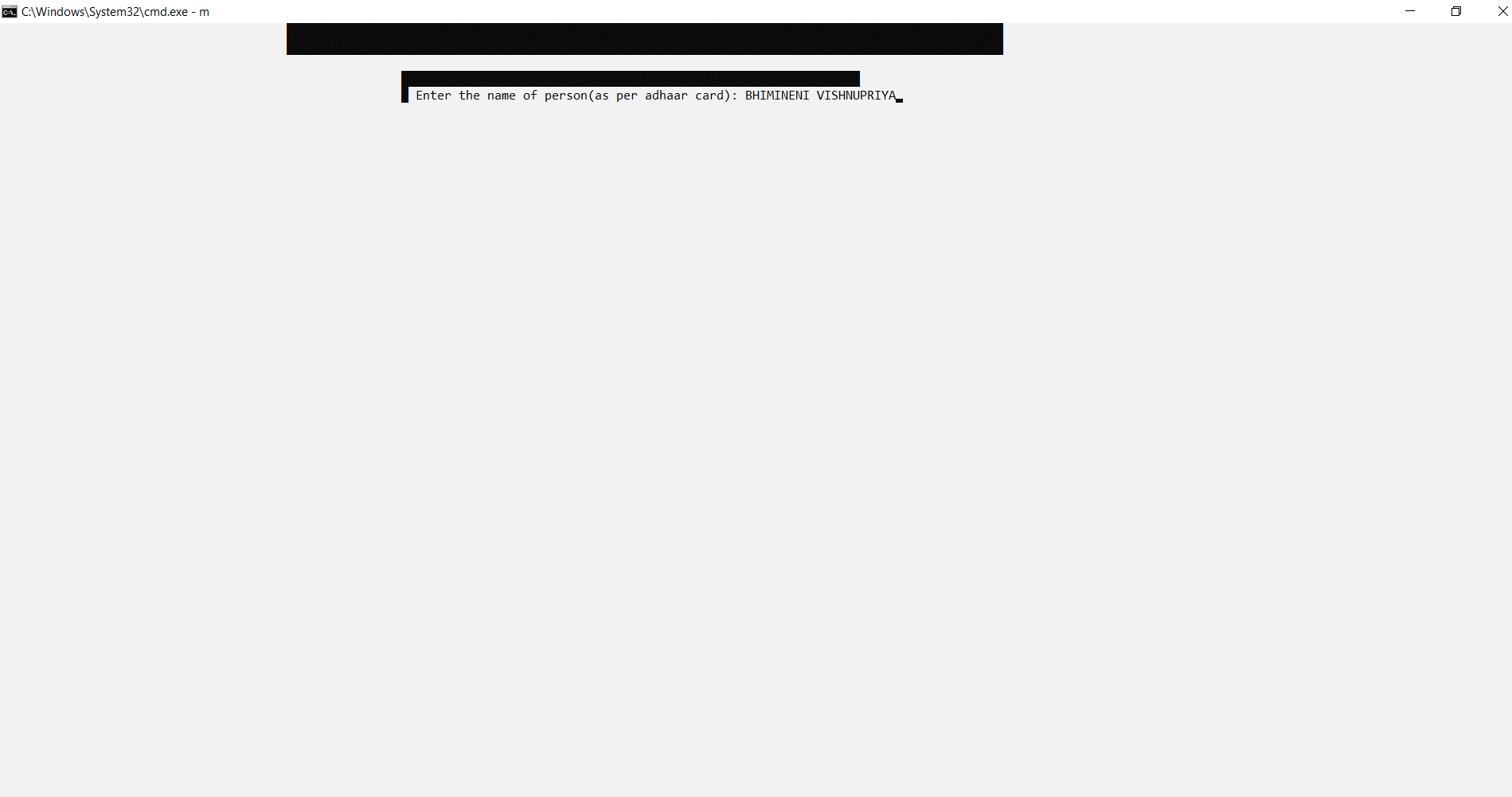


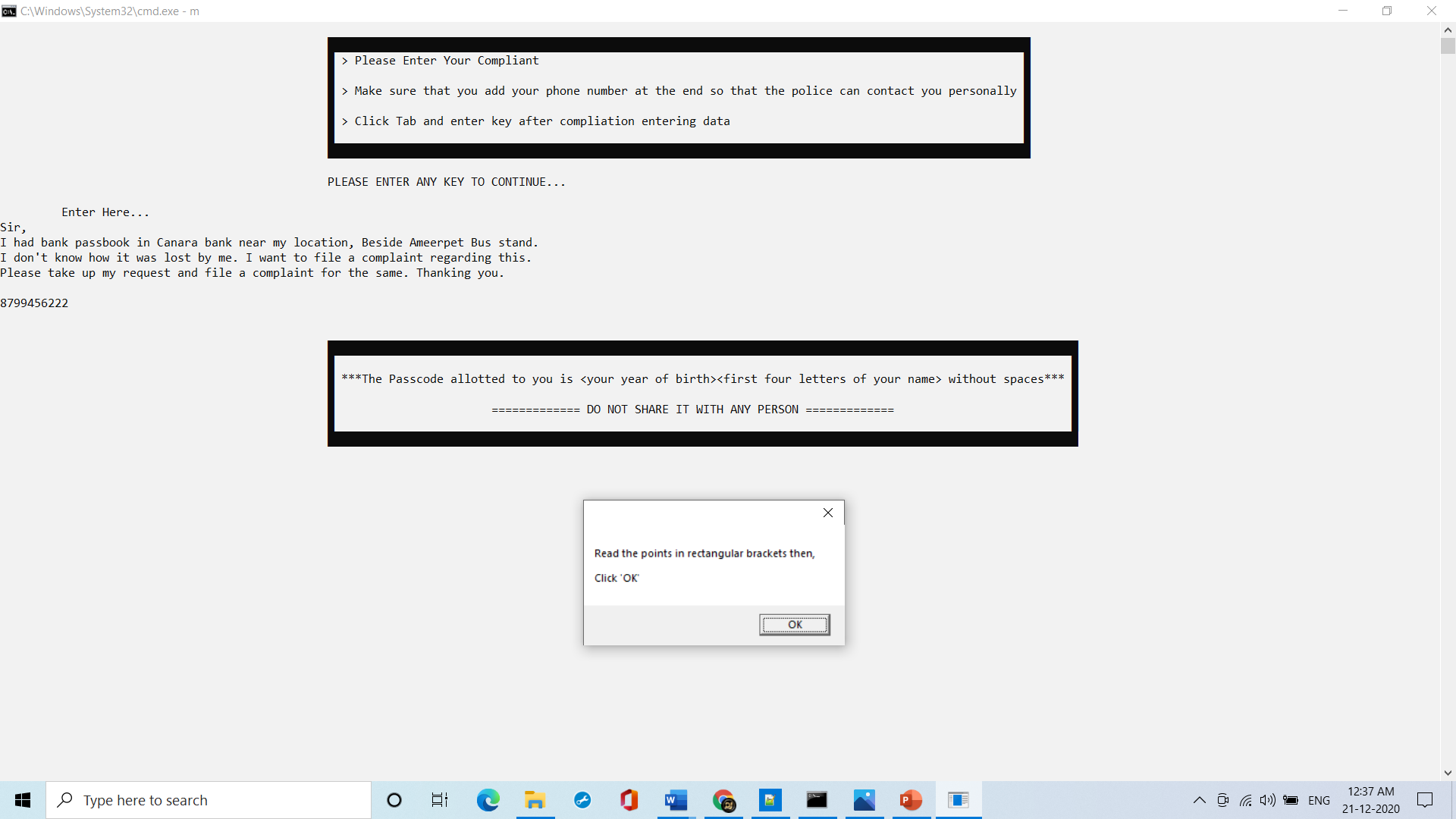


**Test Case 15:**

Adding of complaint by the police

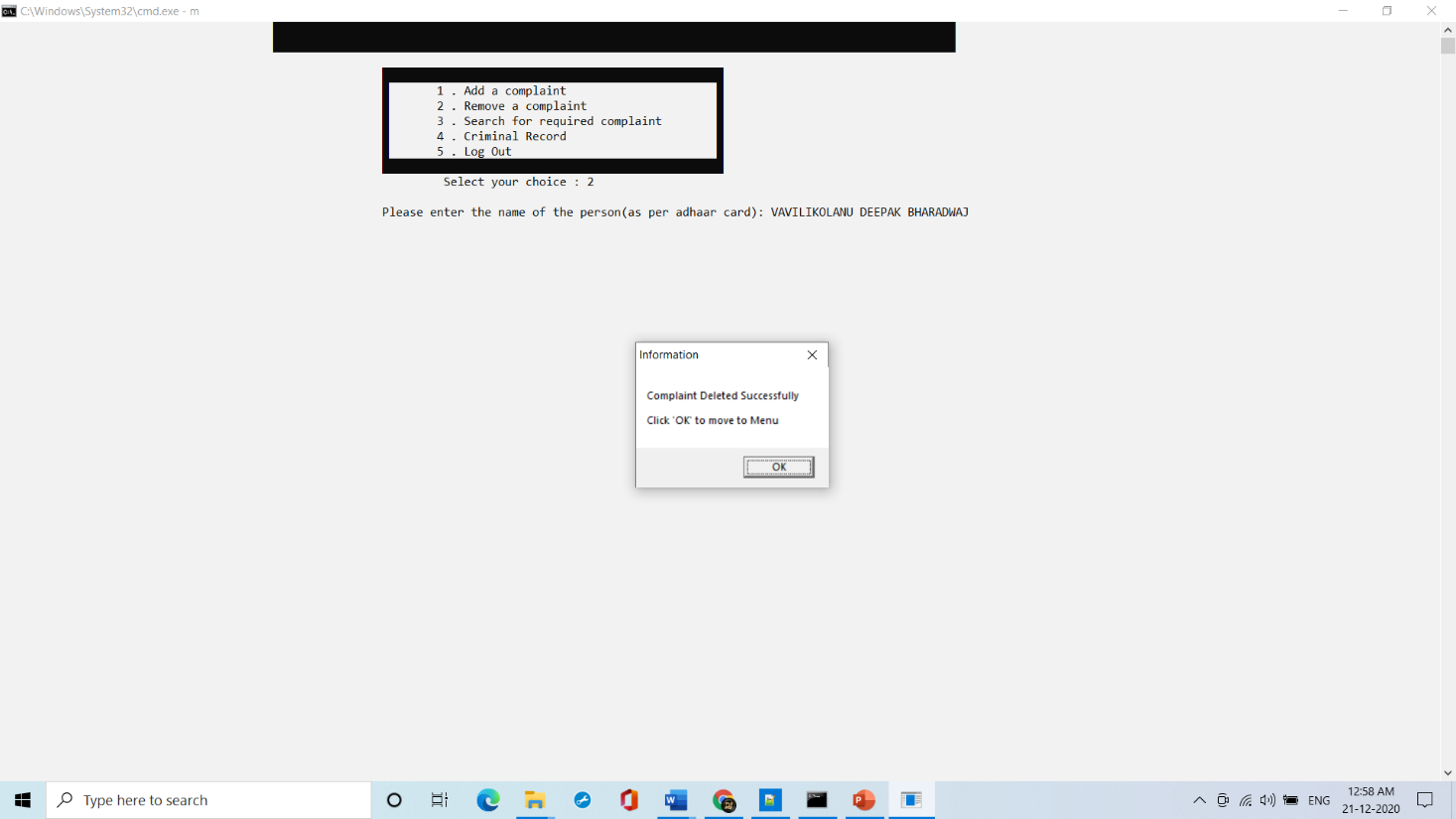






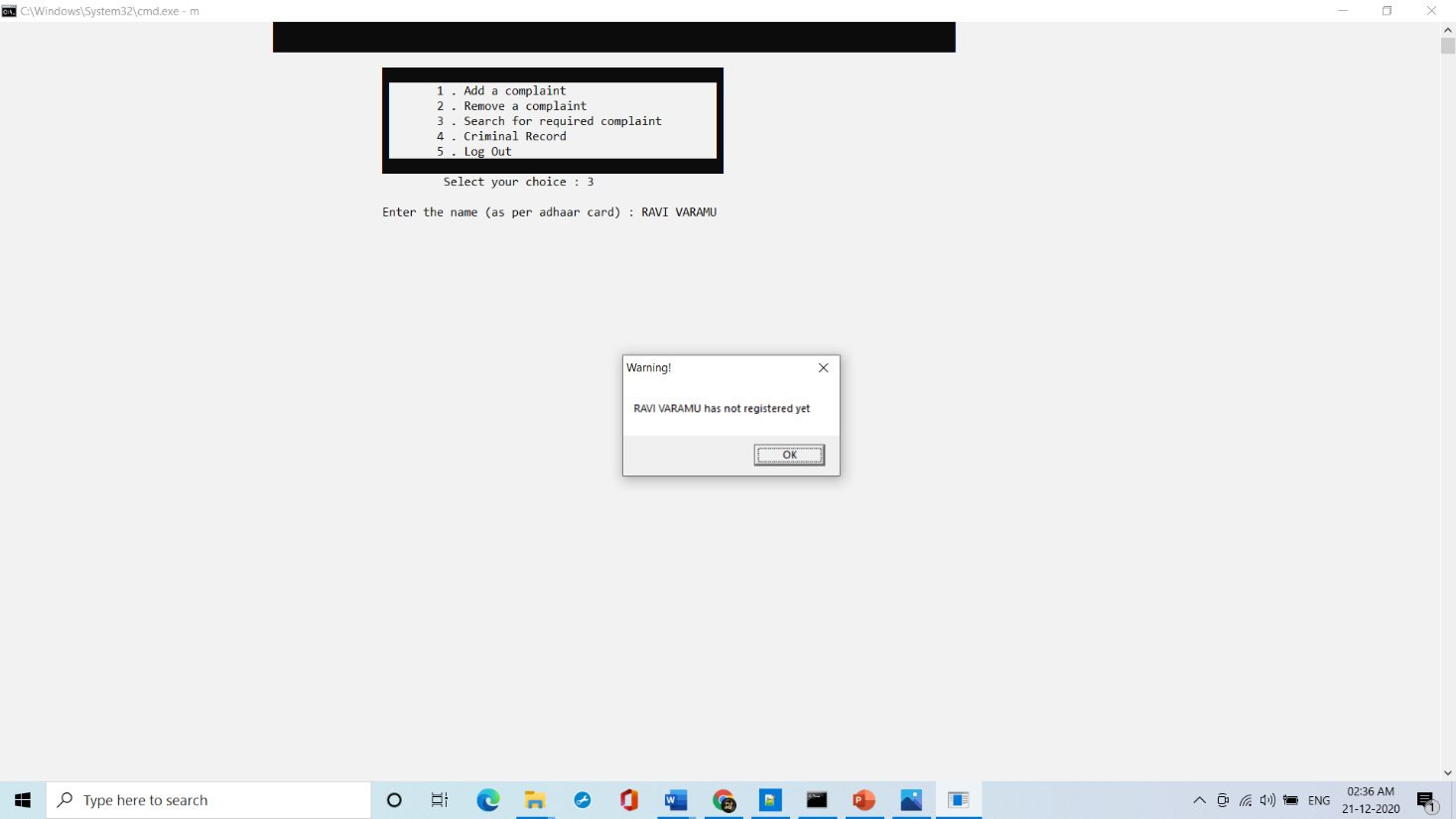
**Test Case 16:**

Removing of a complaint by the police

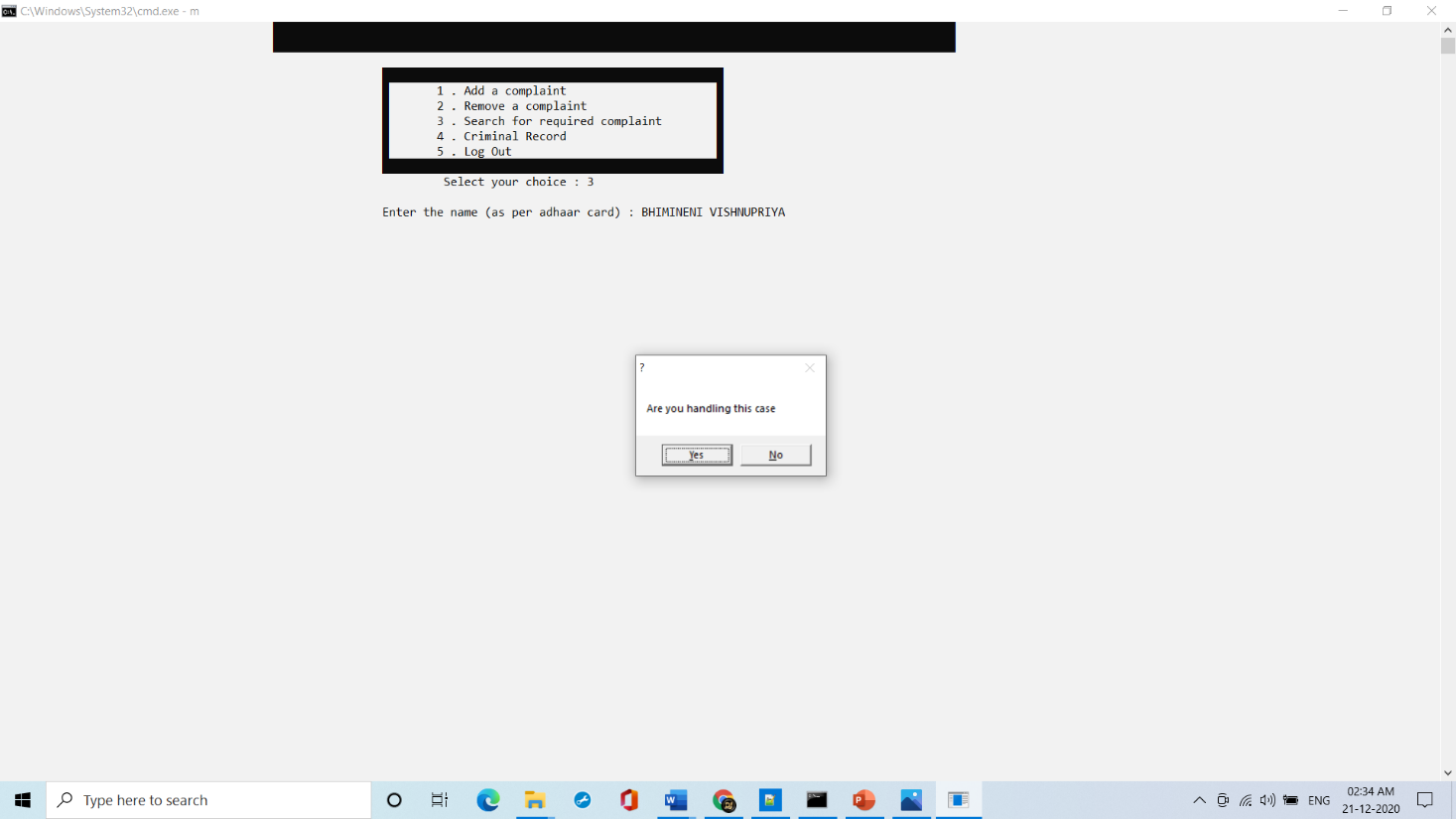


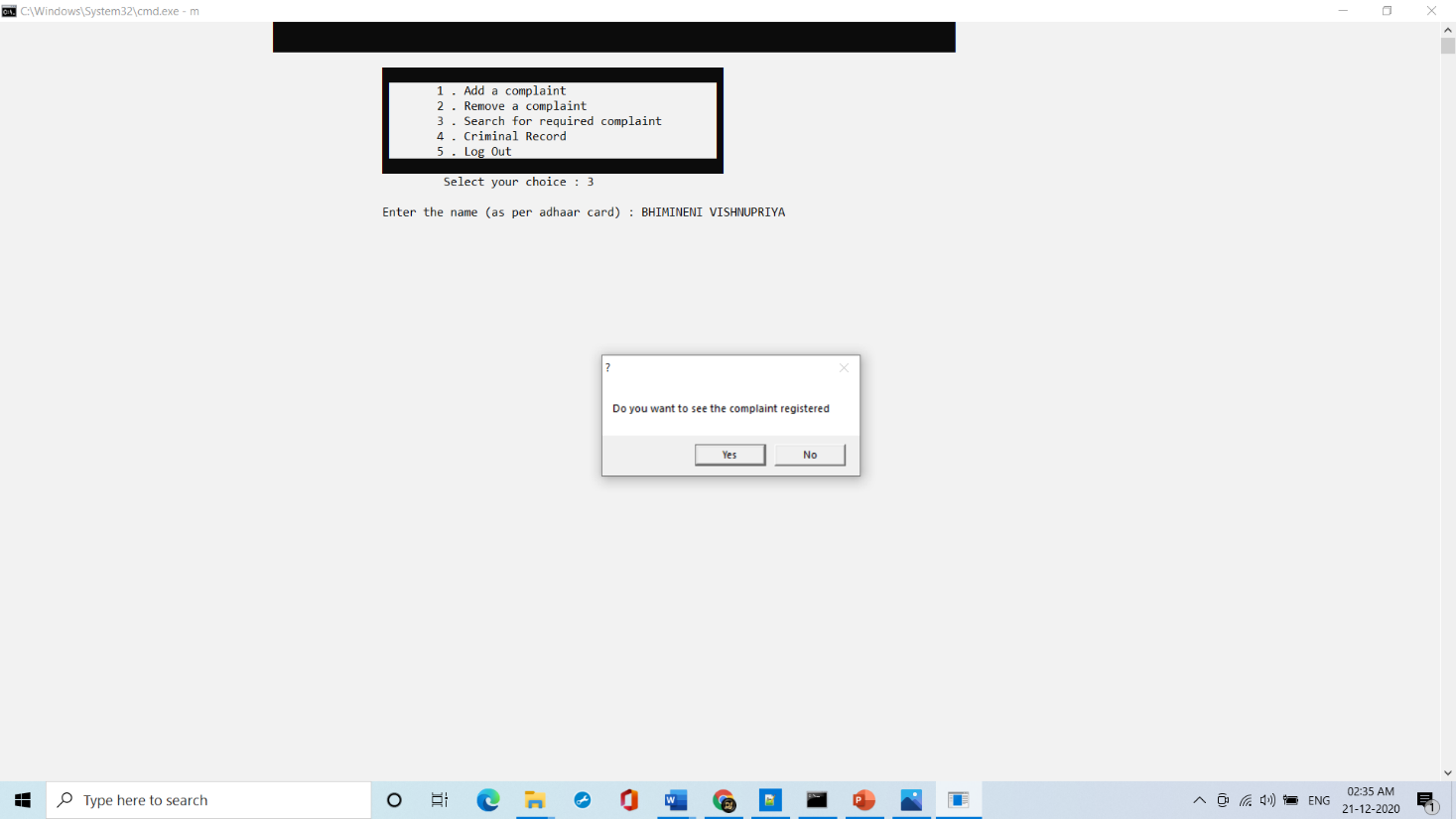
**Test Case 17:**

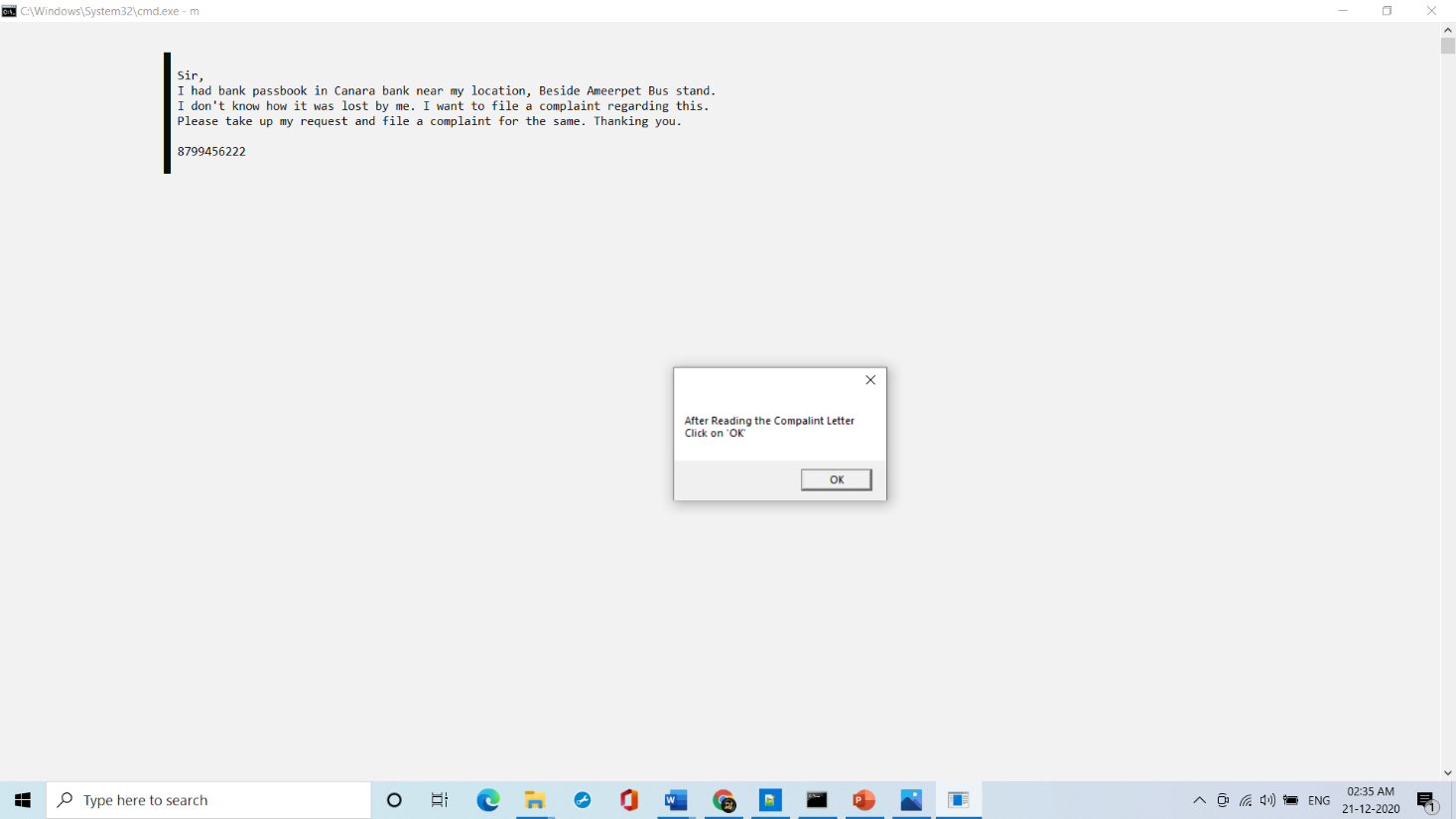
Searching for the complaint details which is not registered

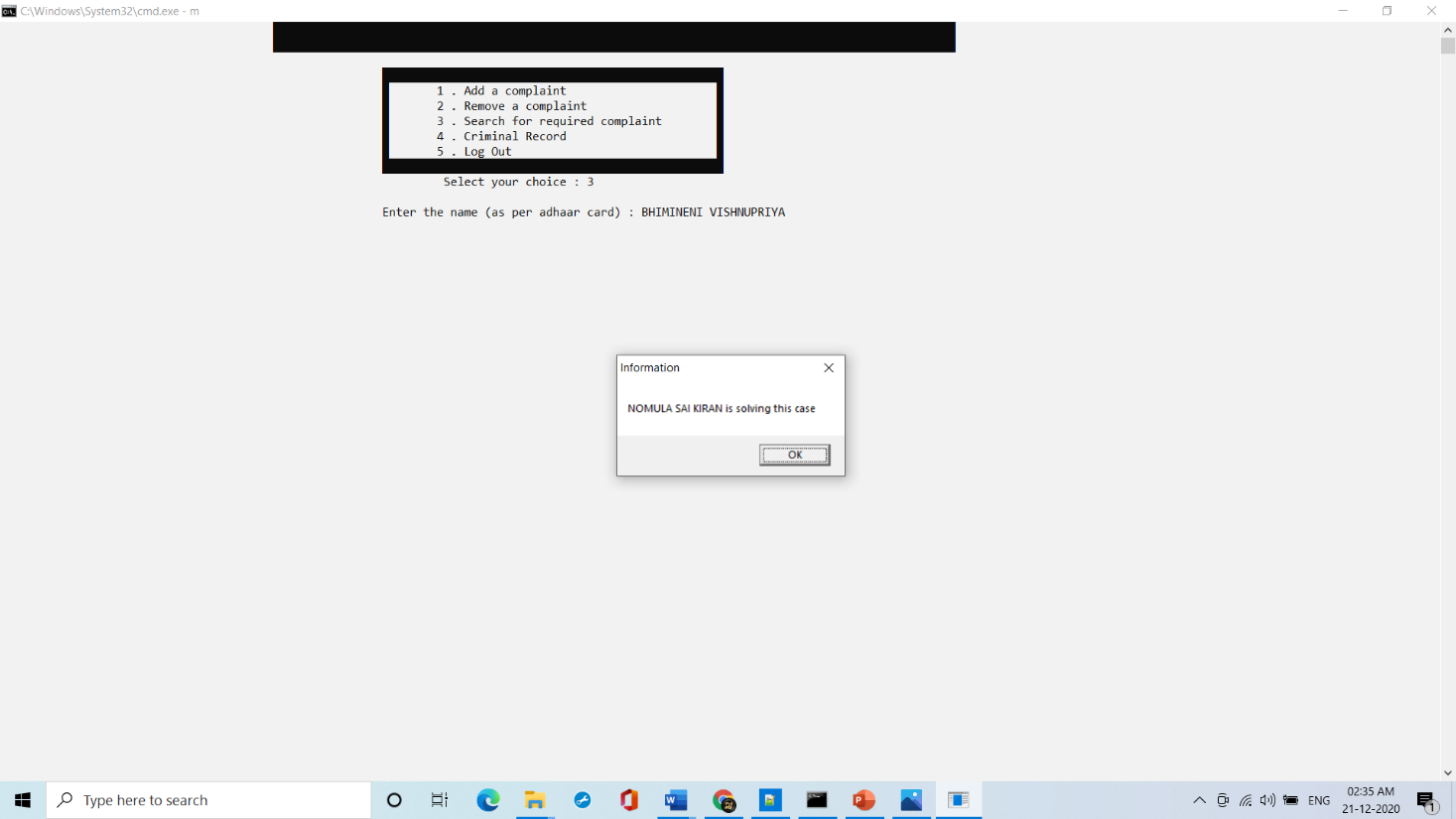


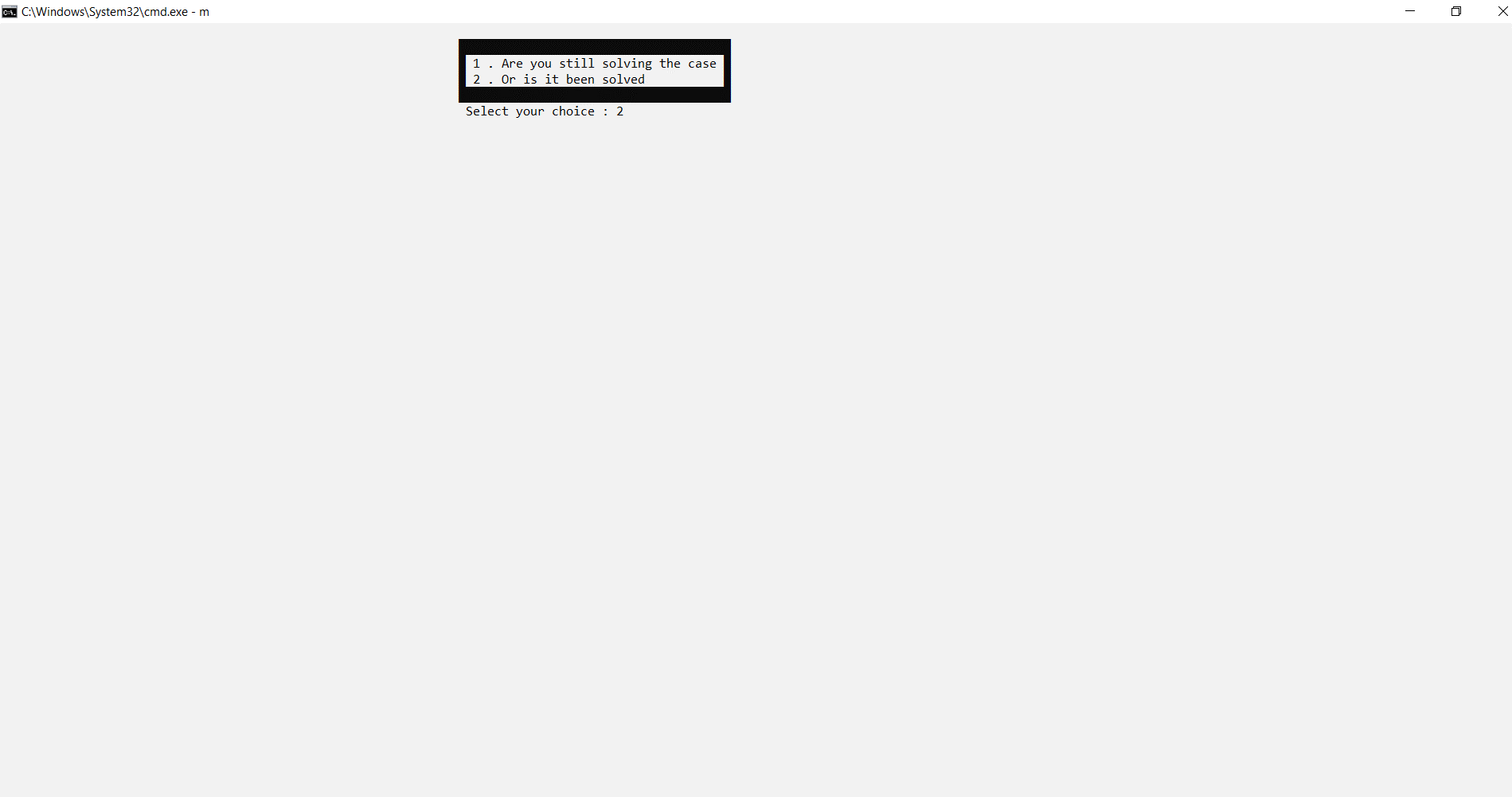
**Test Case 18:**

Searching for the complaint details which is registered

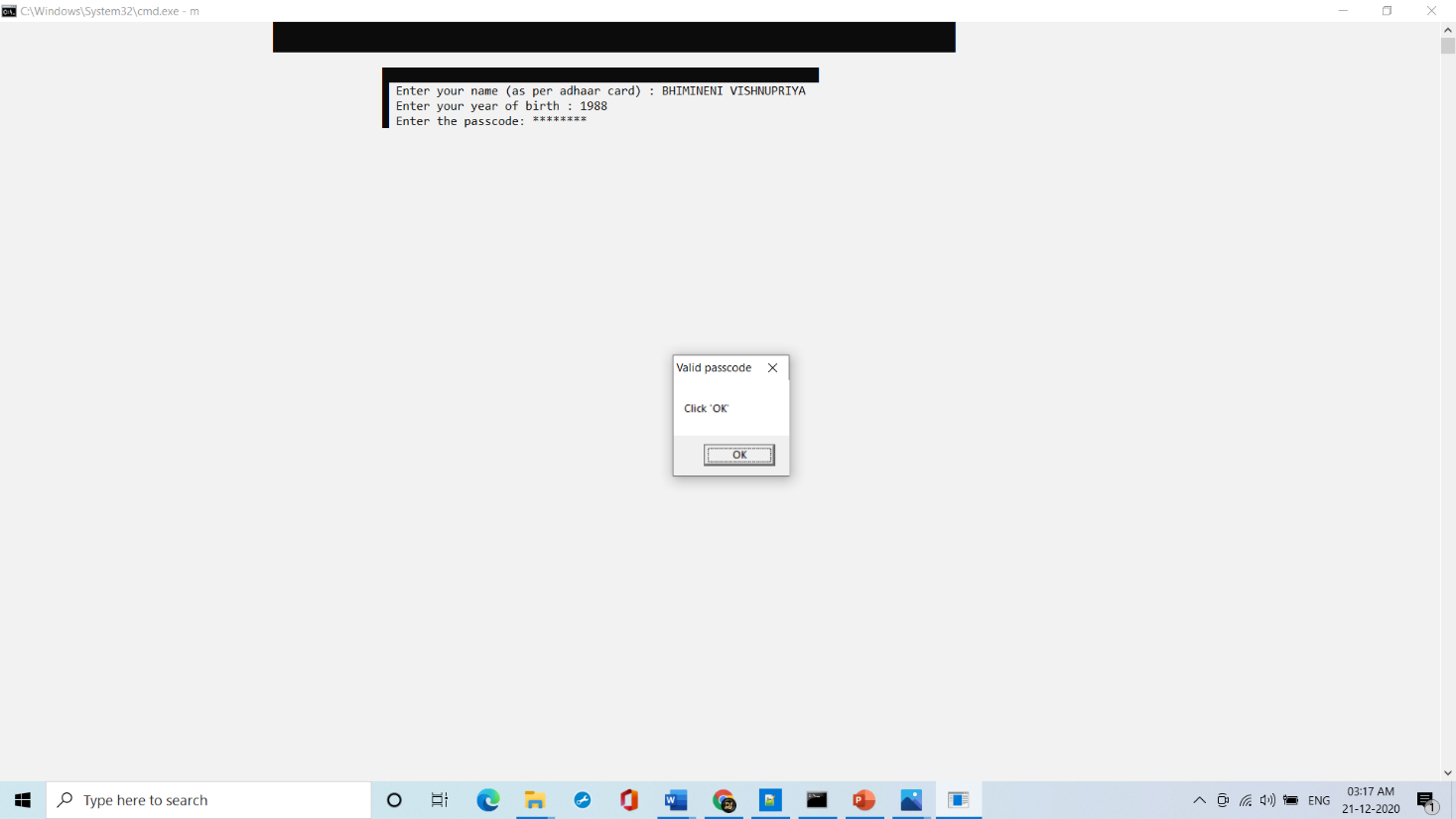


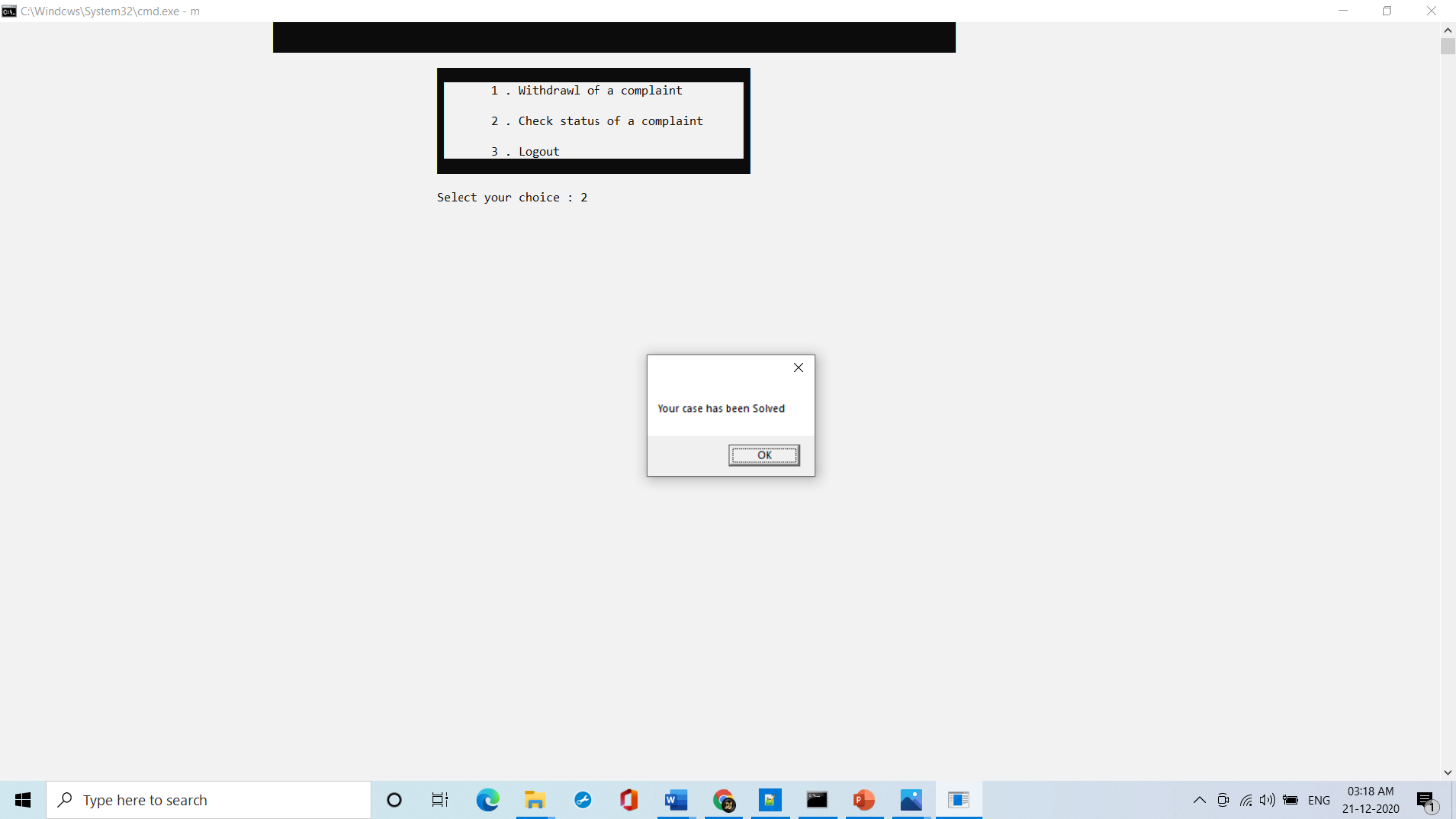






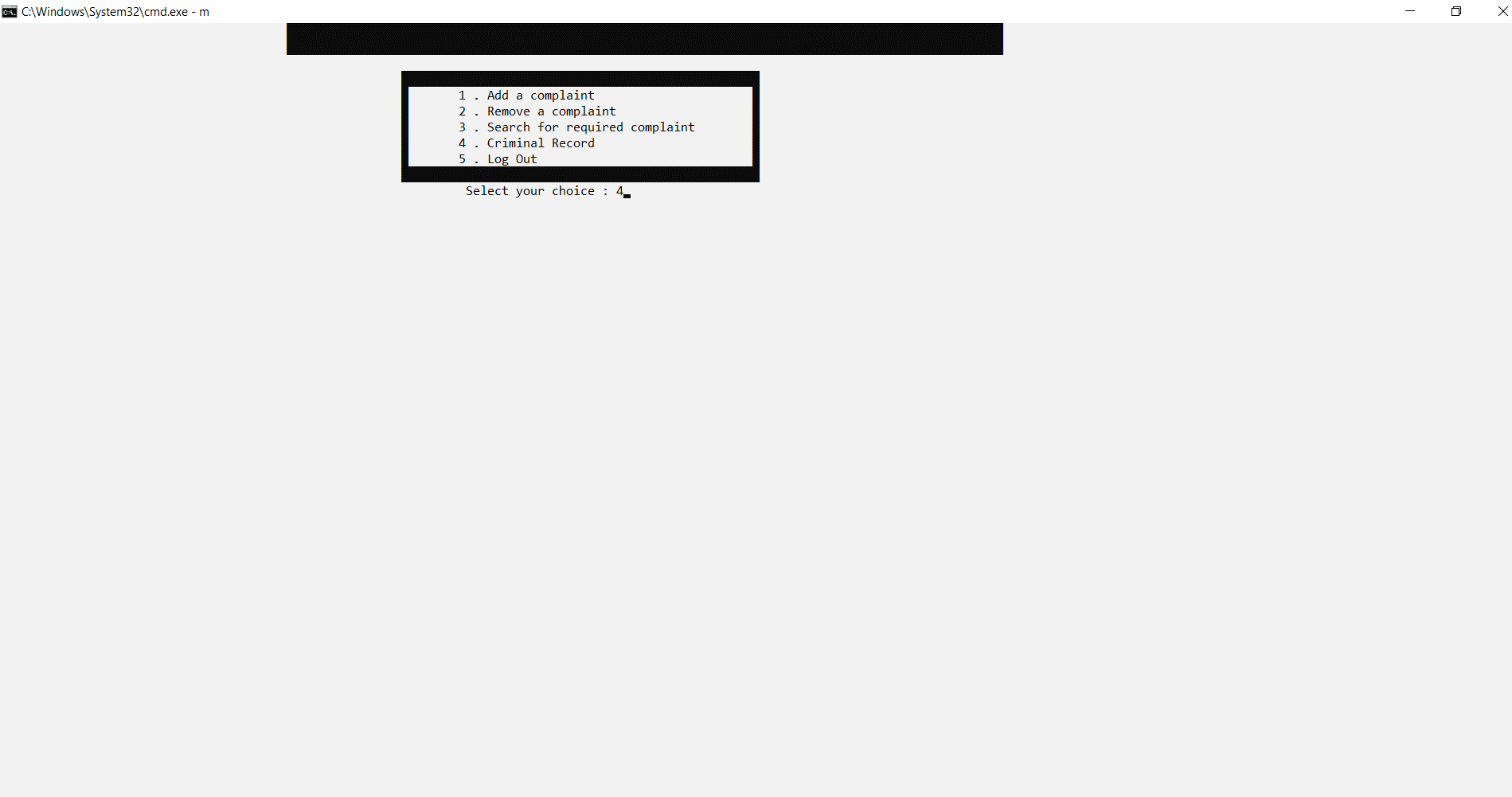
Checking of the same case from User’s side

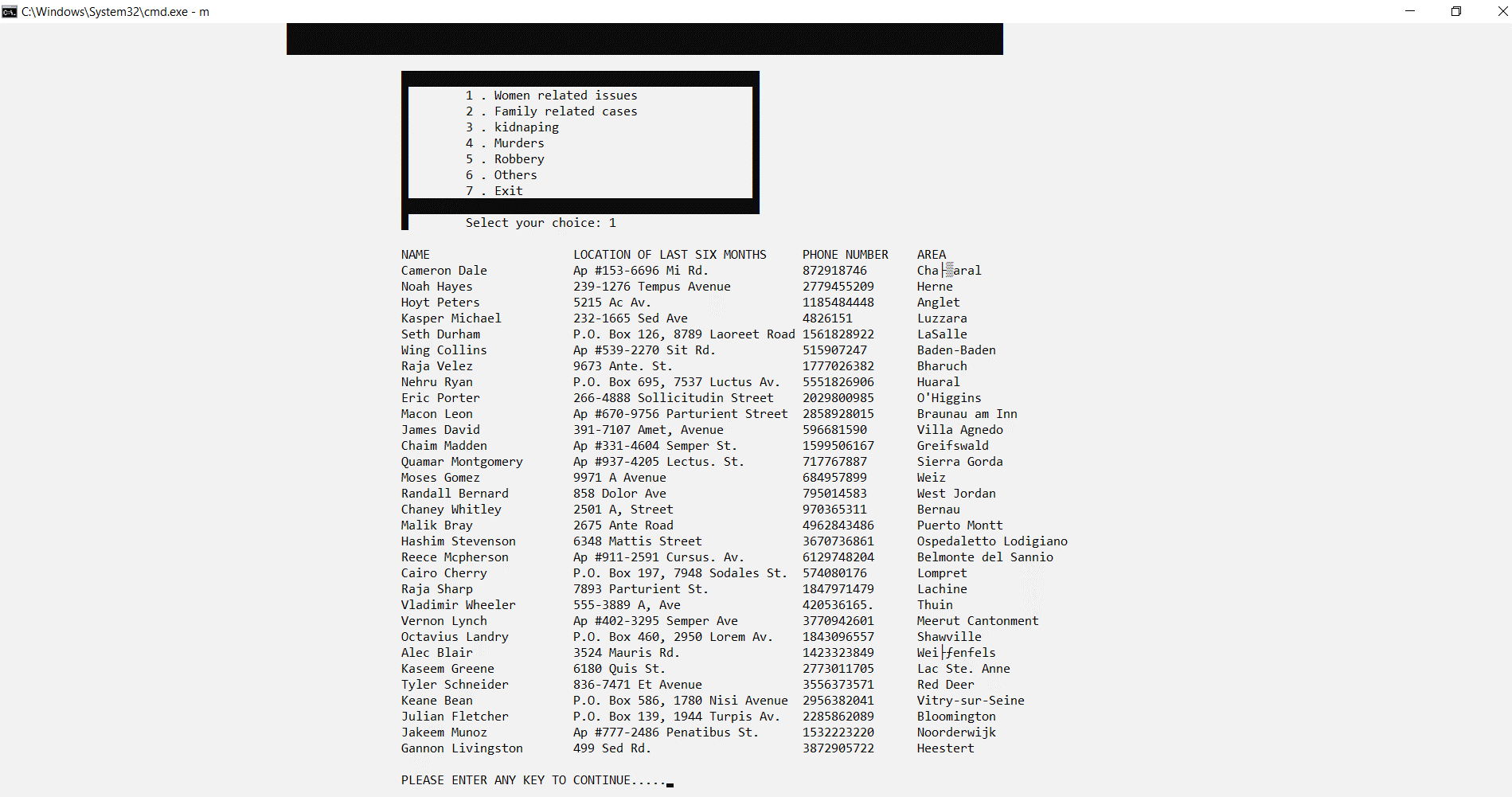


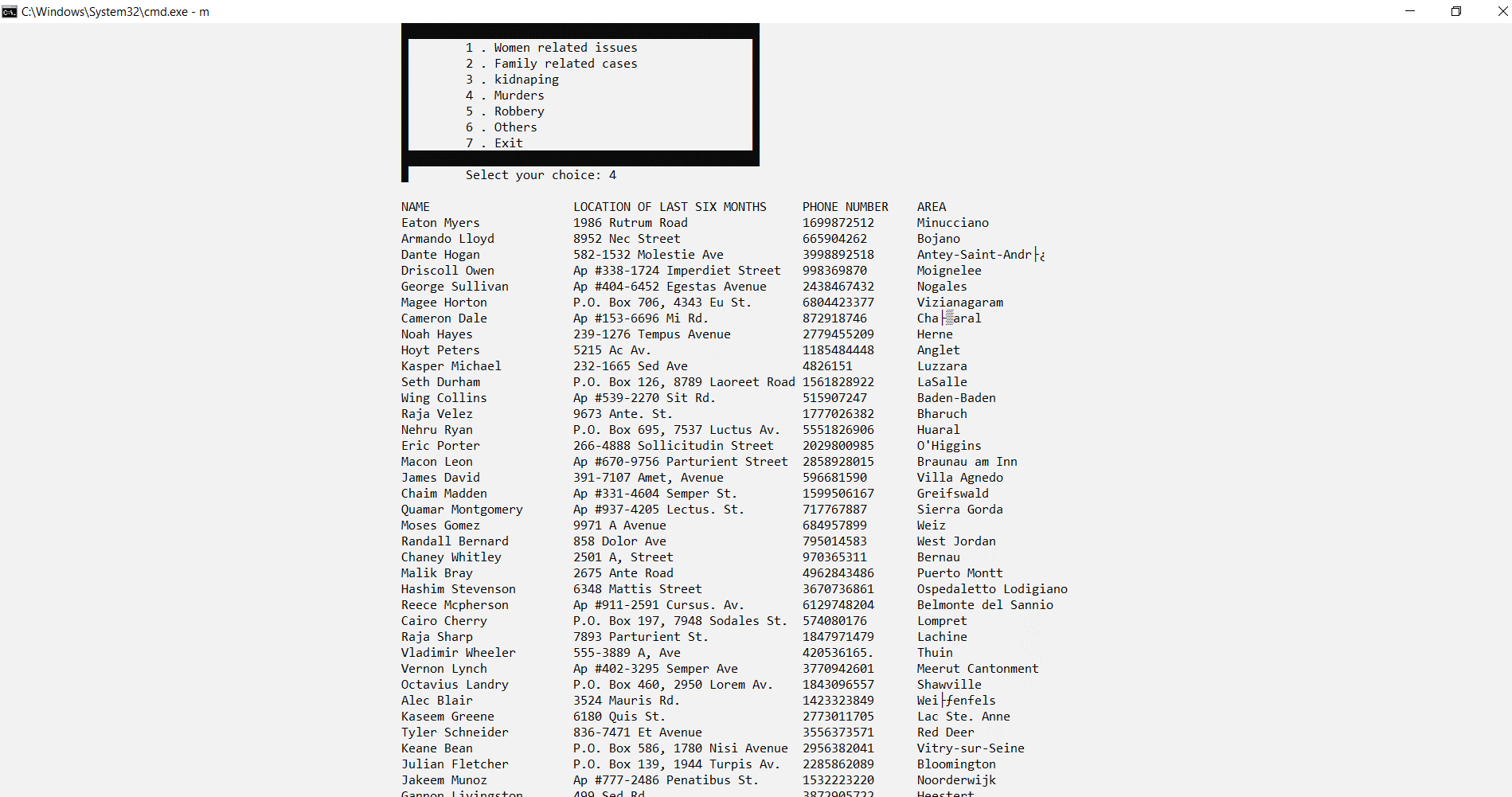


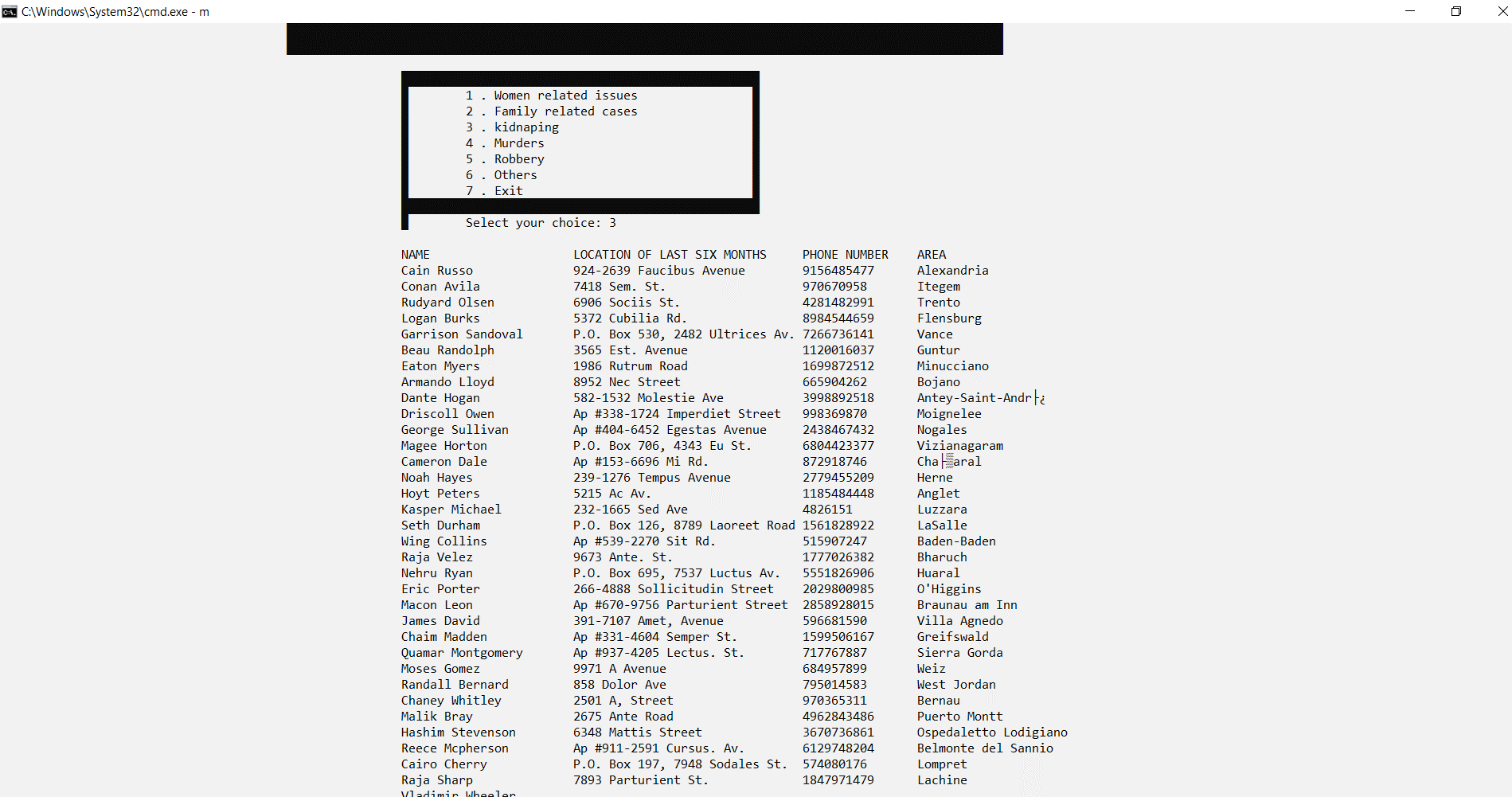
**Test Case 19:**

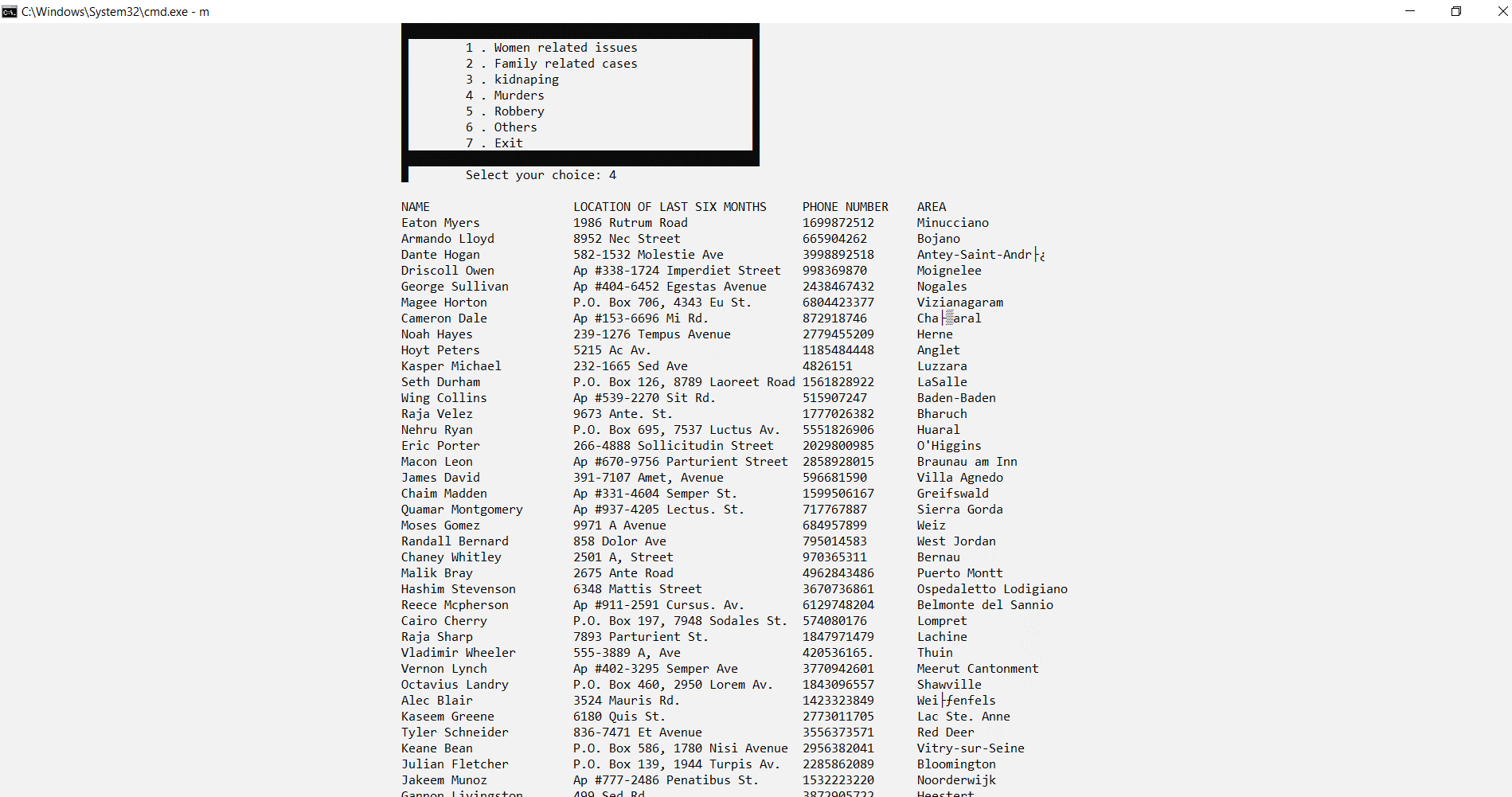
Searching of Crime Record of a person and from a place

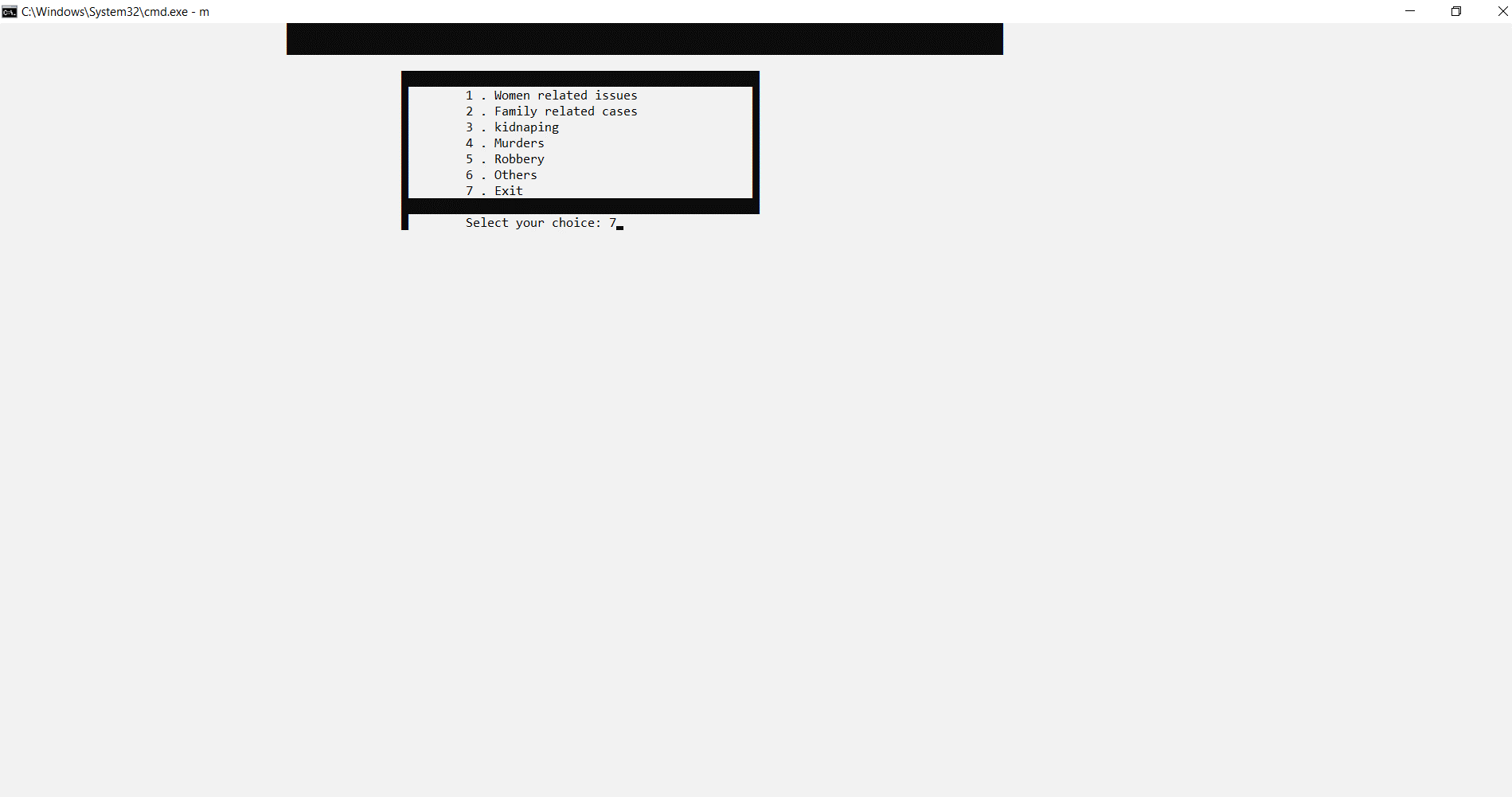


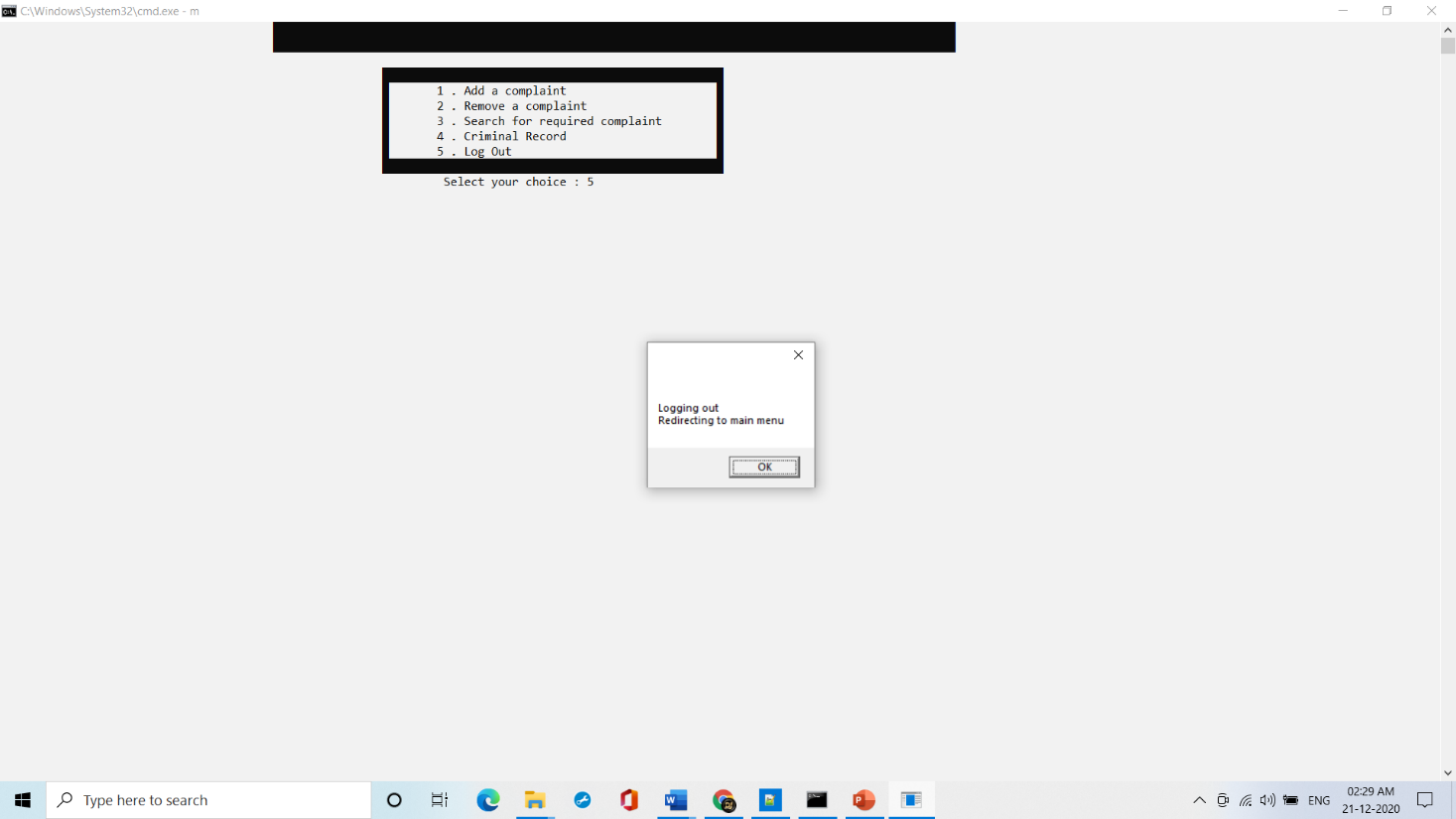






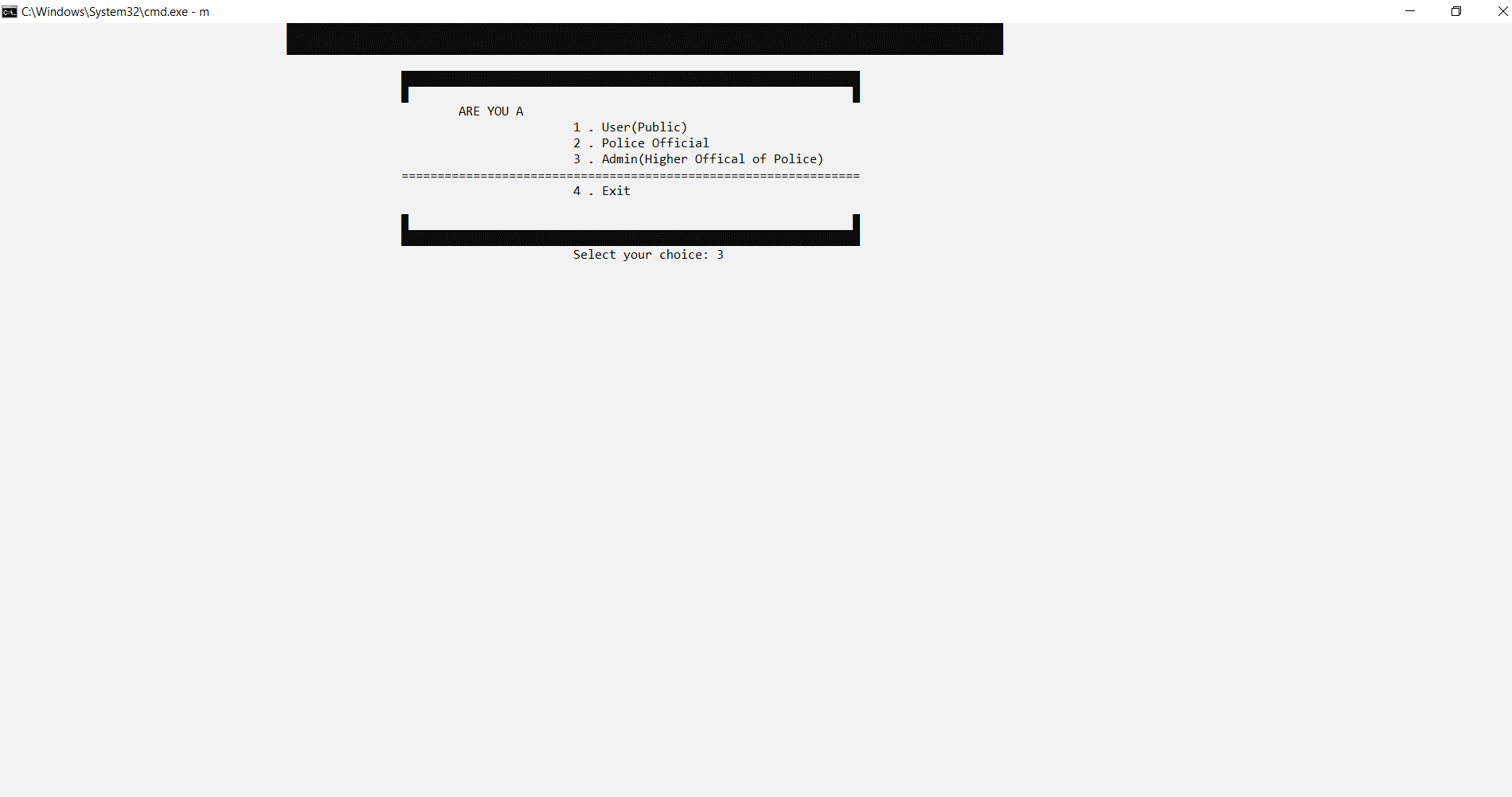


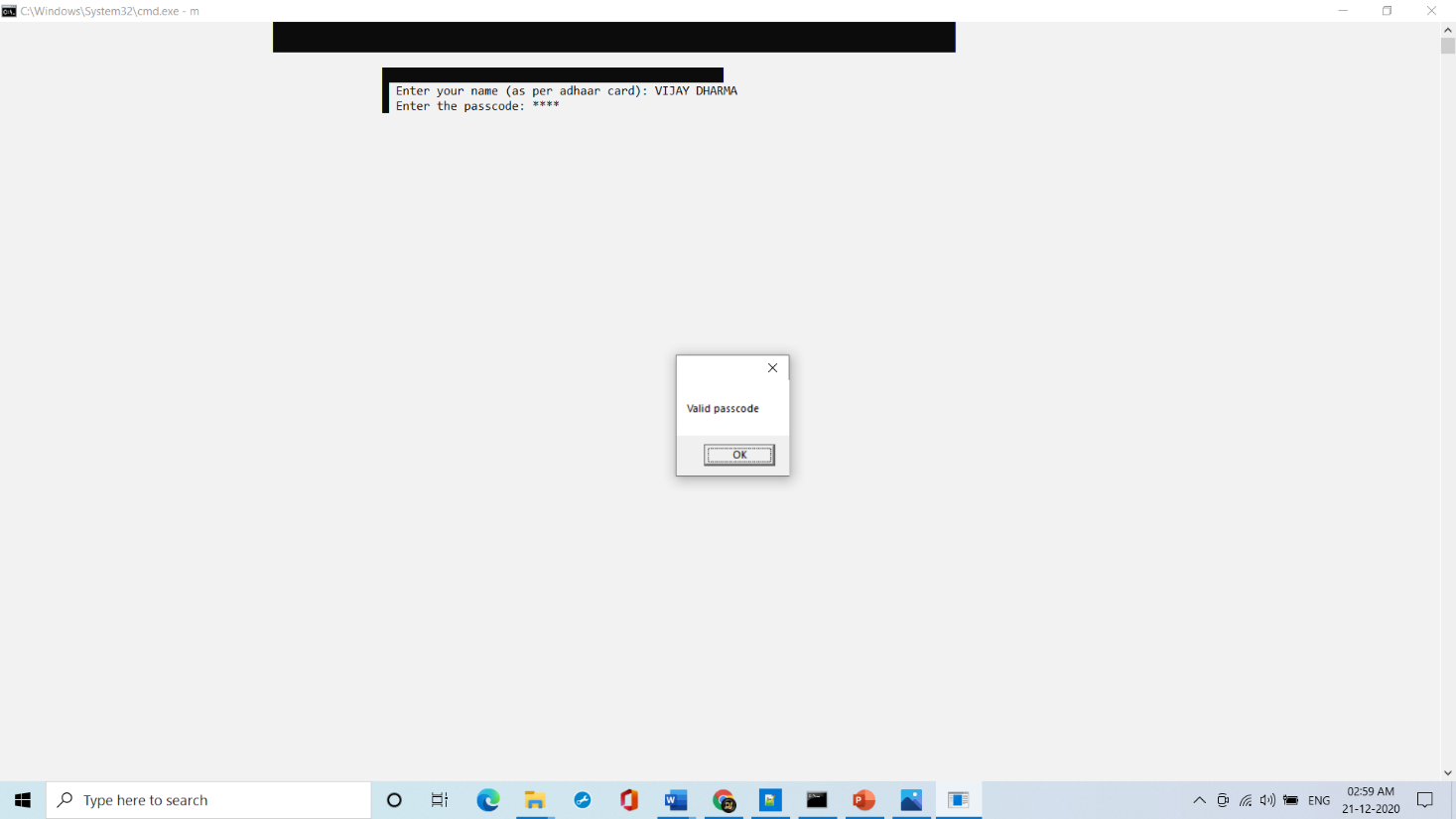




**4.3 Admin Test Cases Results**

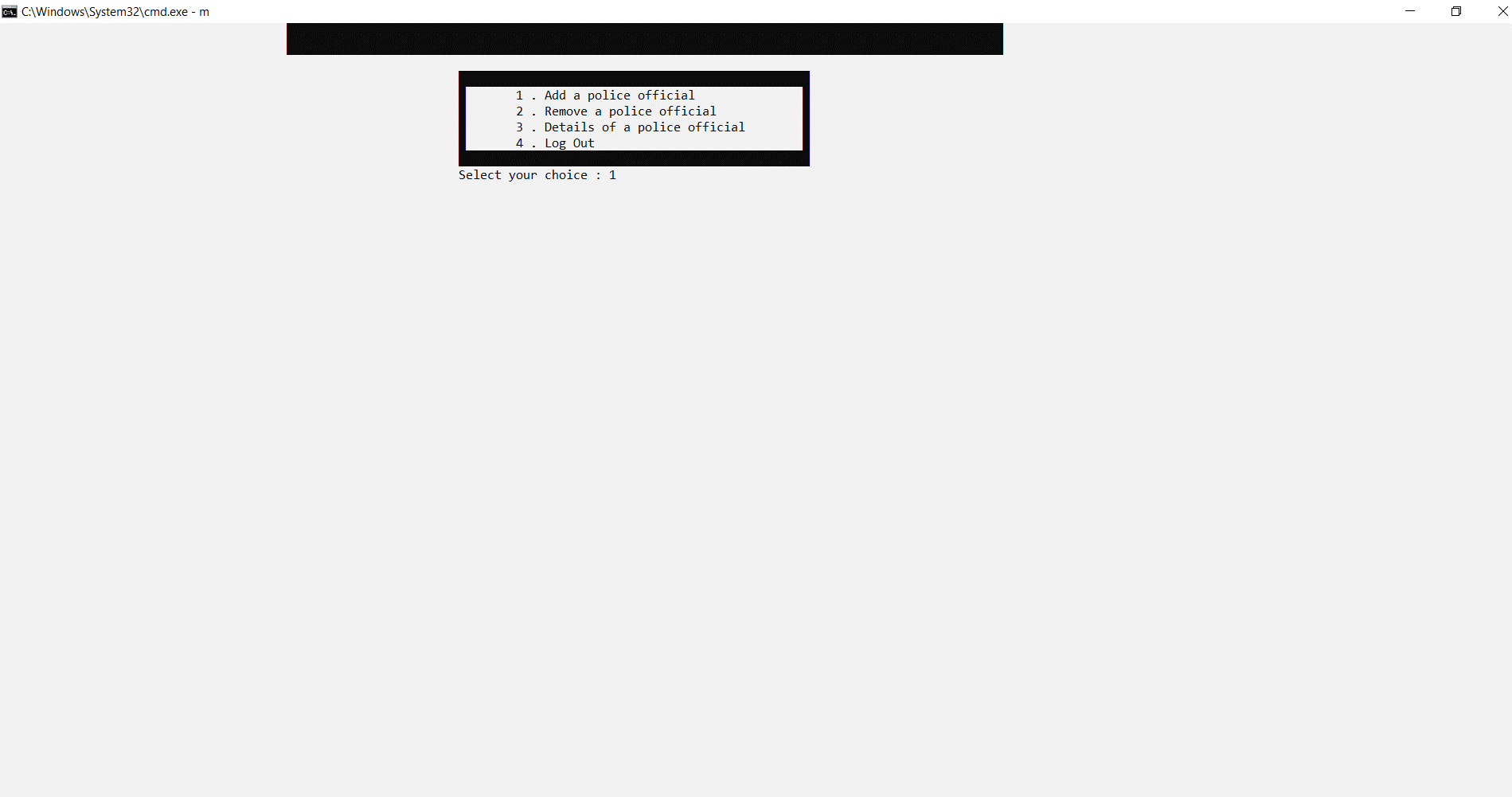
**Test Case 20:**

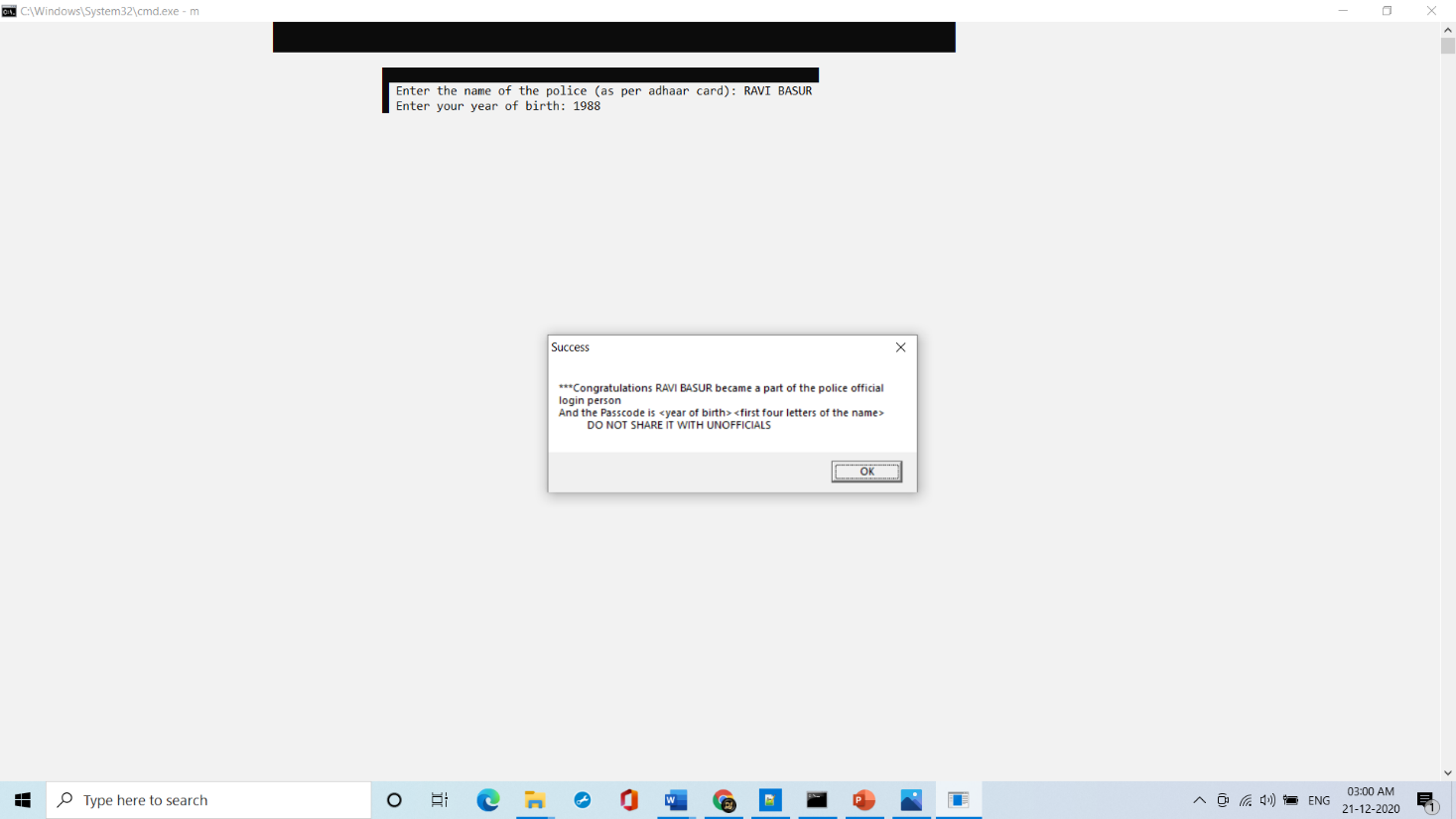


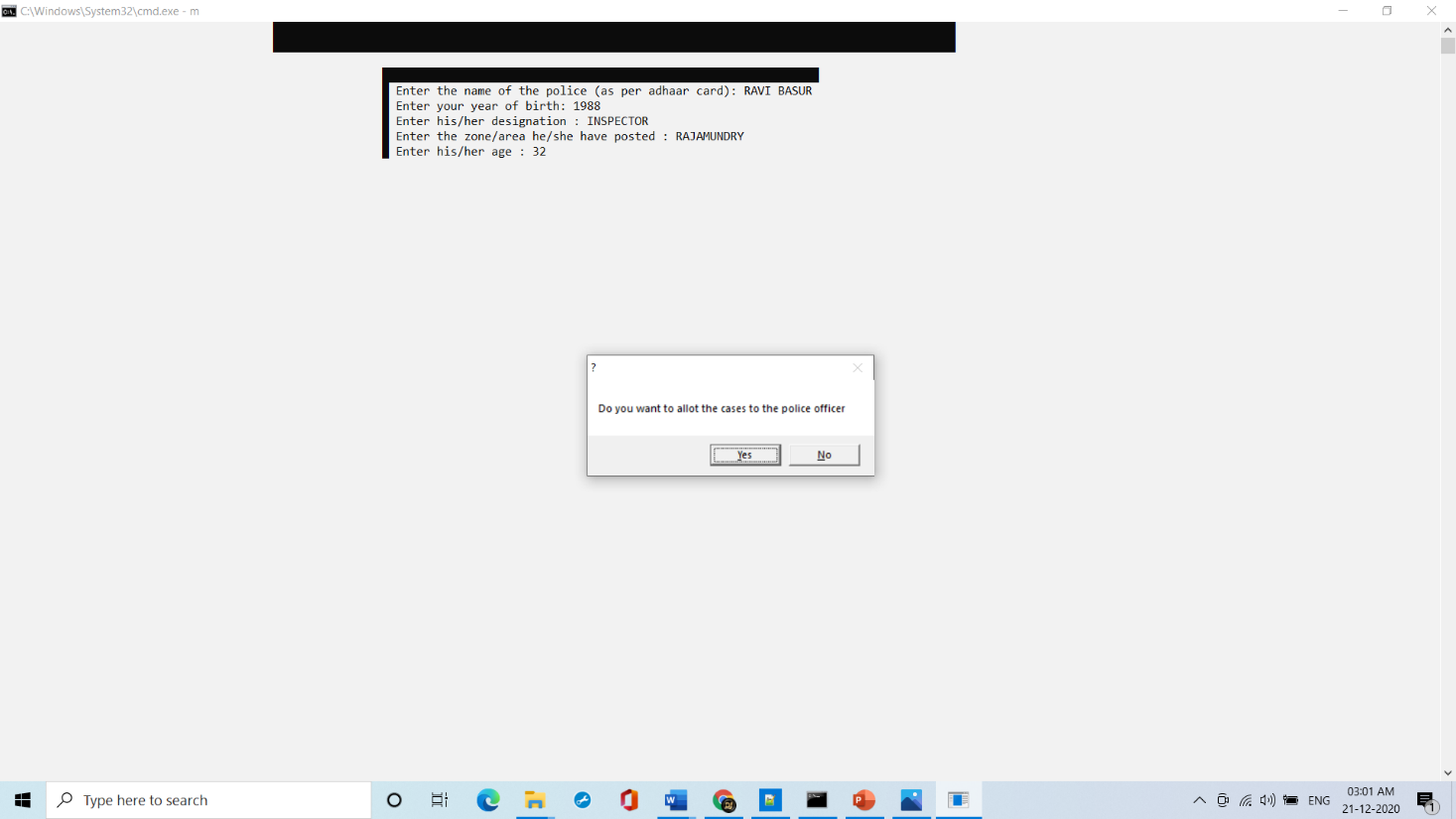


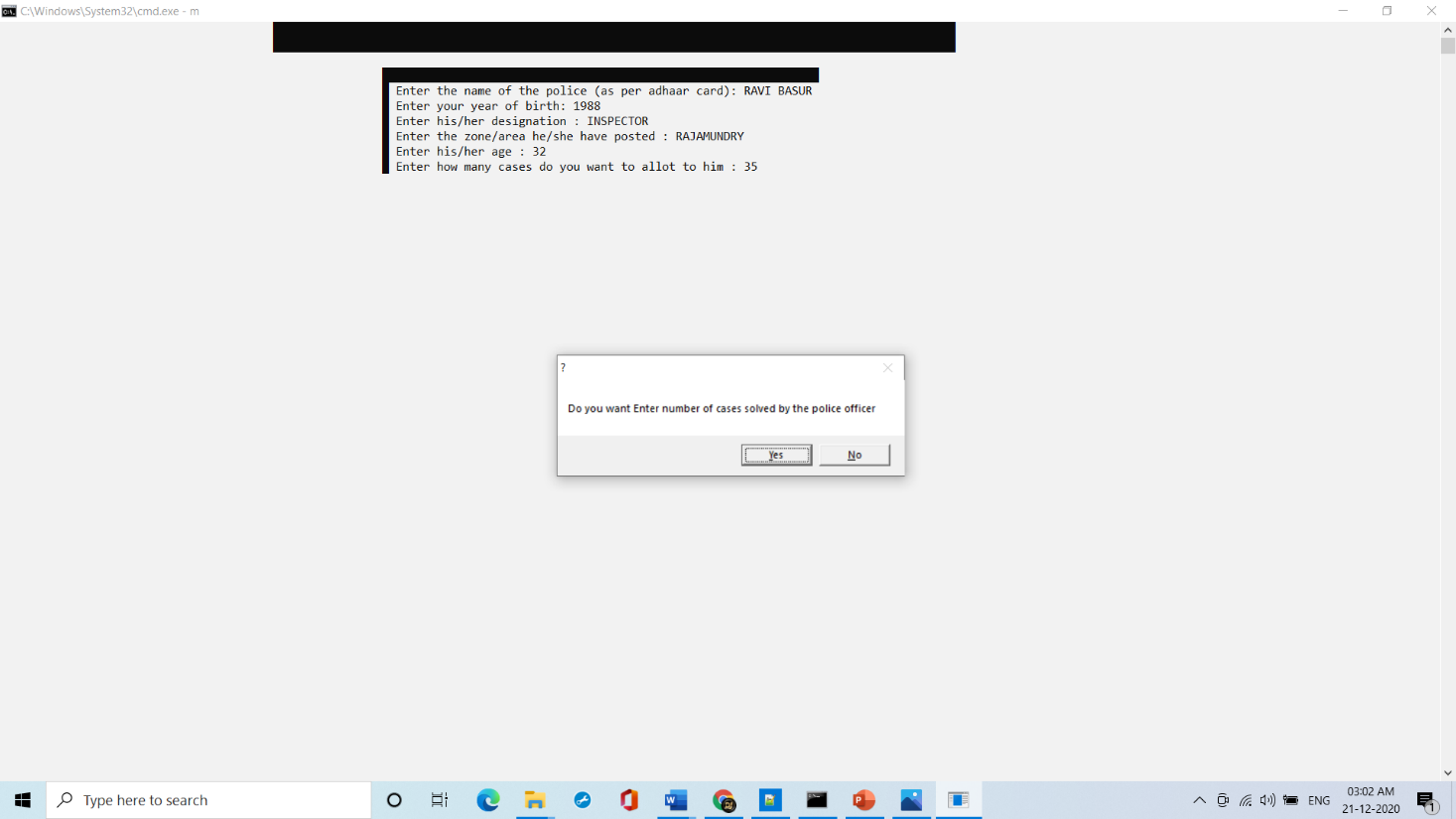
**Test Case 21:**

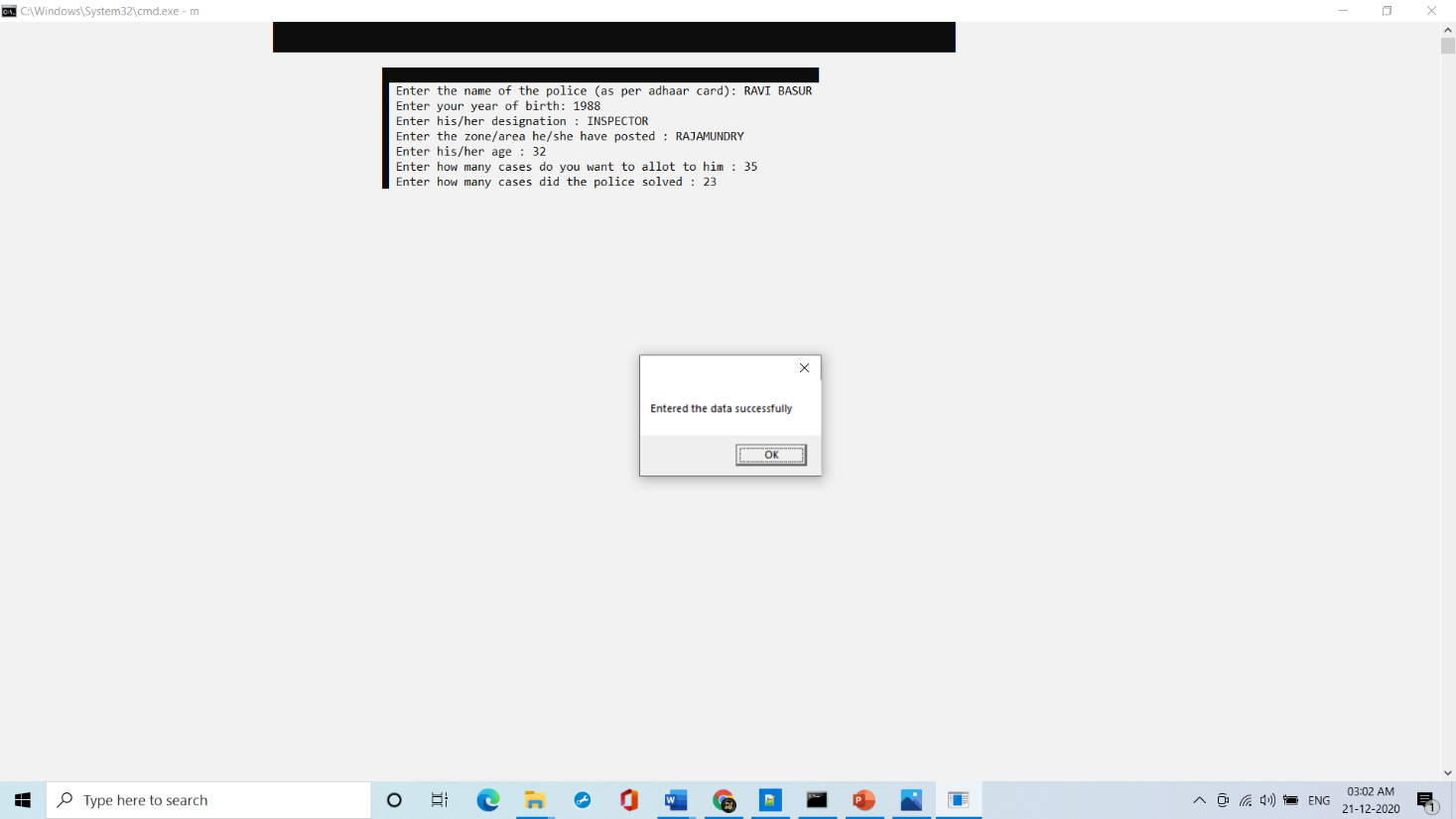
Adding a police official





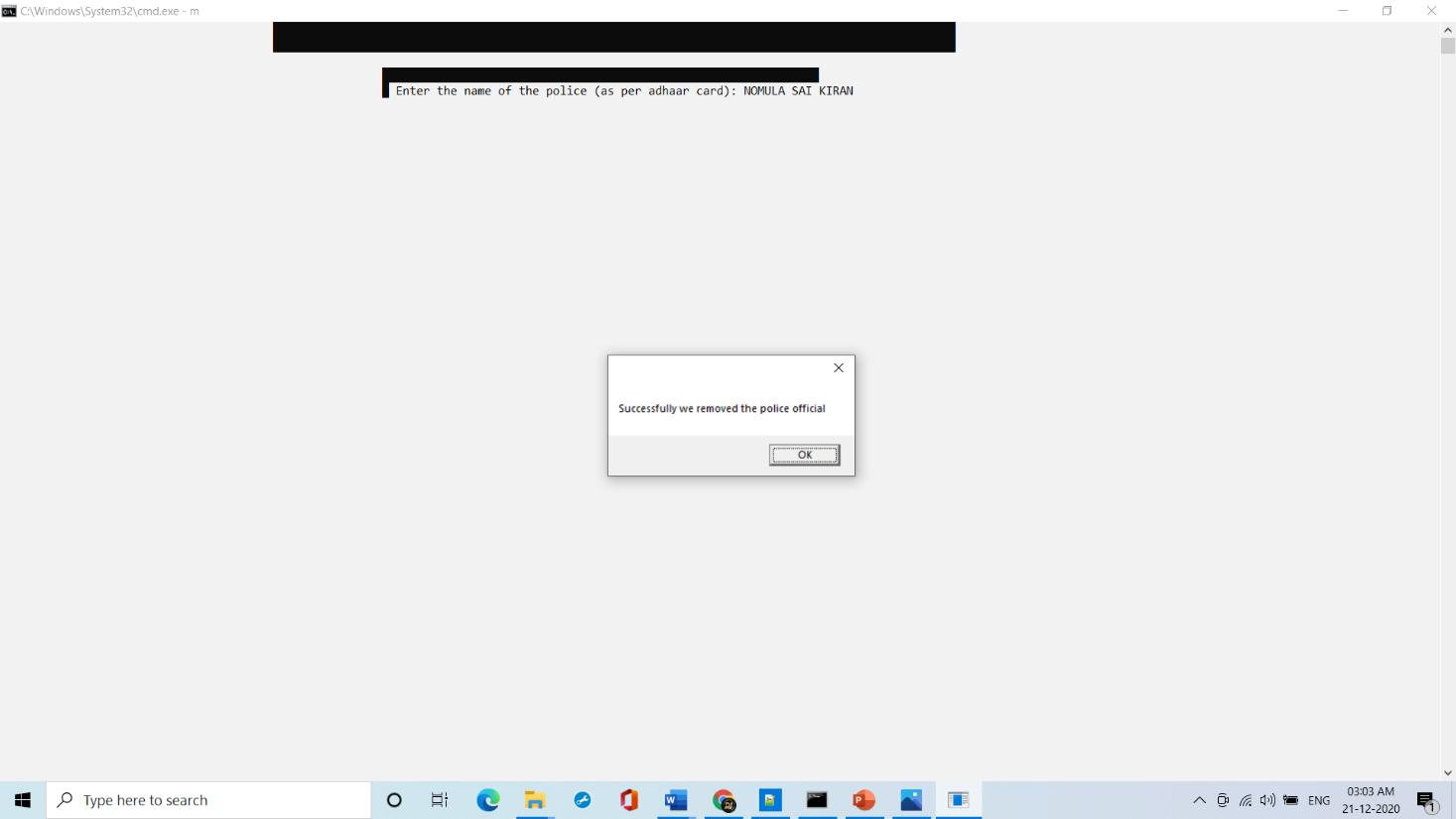




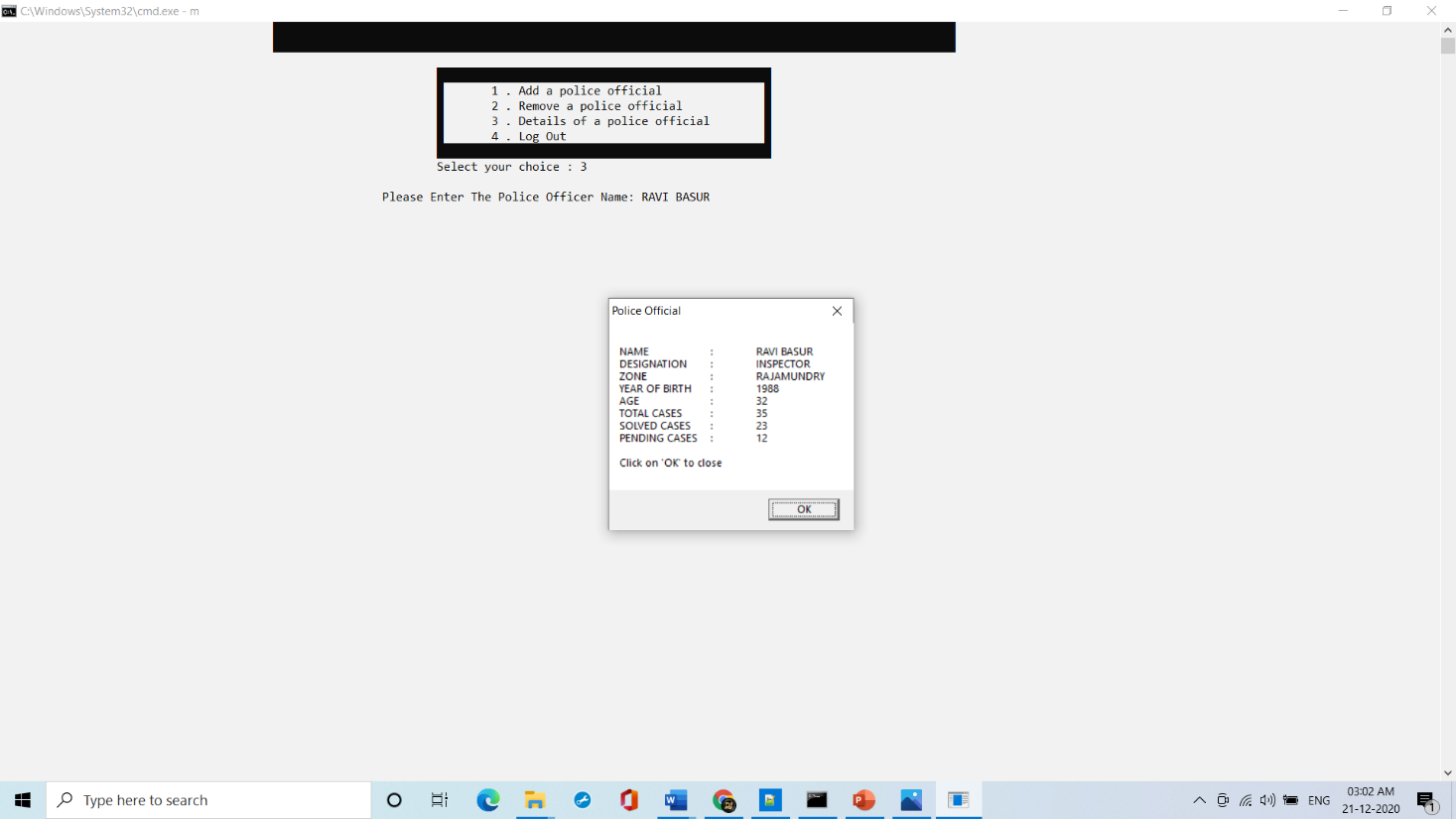


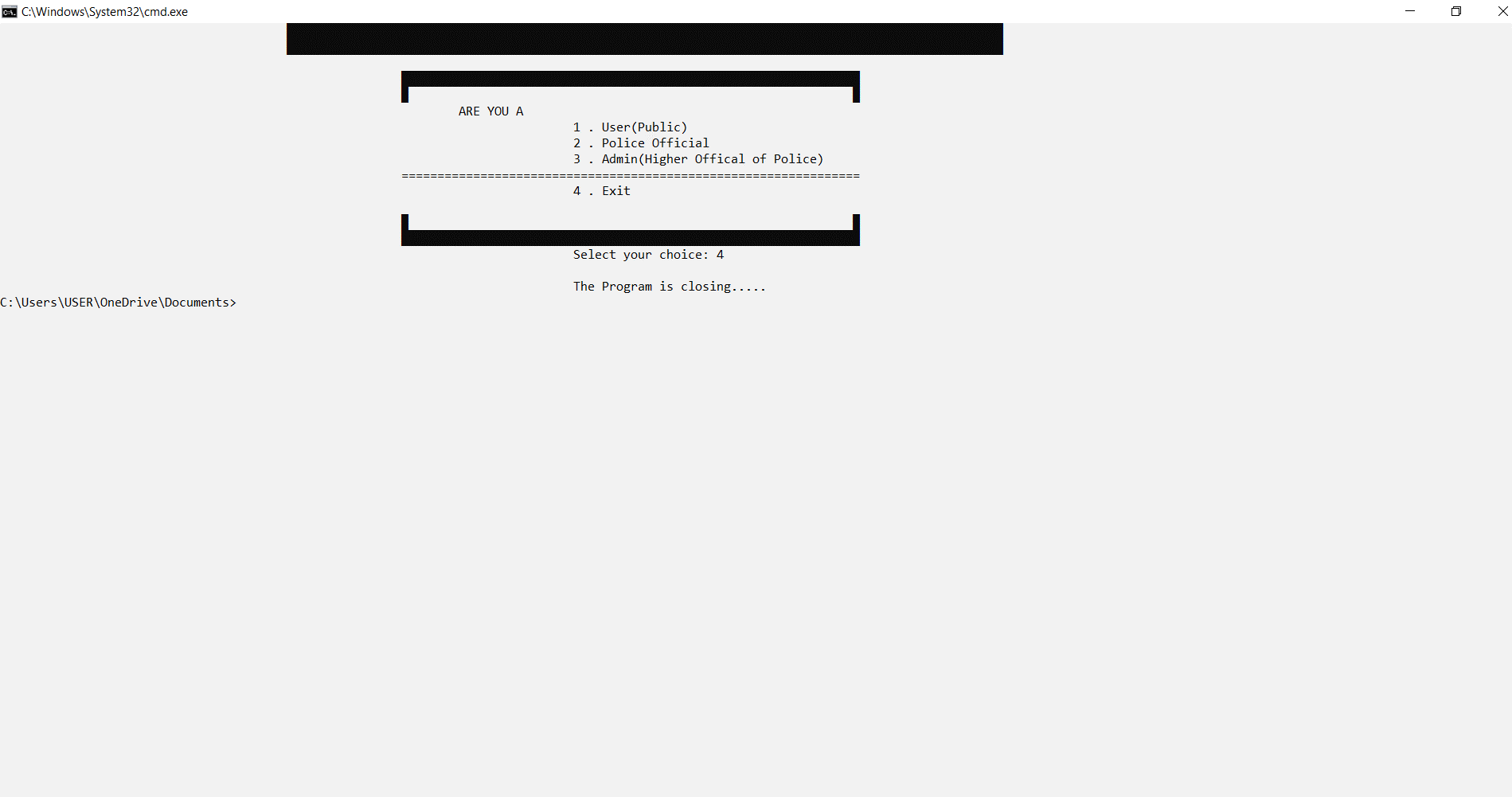
**Test Case 22:**





**Test Case 23:**





# ADDITIONAL KNOWLEDGE ACQUIRED

Implementing this project in C Language has introduced us to different libraries such as: ‘conio.h’, ‘ctpye.h’,’unistd.h’,’sys/stat.h’ and ‘windows.h’. We were able to use the knowledge we have on the Linked List Data Structure and execute it as a real-time application. We used the ‘windows.h’ library for controlling the display colours in a controlled manner. We explored the ‘conio.h’ libraries for achieving a look-and-feel of an actual window application by constructing our own time delay function.

Also, we have further improved in our knowledge in file-handling because of the vast amount of data manipulation we have done using text files.

Other than this, we have learnt the value of team spirit and have understood the intention behind working in teams. We have learnt to be team players.

# CONCLUSION AND FUTURE WORK

To conclude, we have built a platform in which we stored data of police official ,admin and user . This project helps in easy way of storing and accessing data.

Our future work includes incorporating more concepts such as Data Structures in our Learning and Quiz modules. Also, we would like to include a feature which allow the User to send requests to the Admin for any changes they require. Other than this, we are deliberating on increasing the authority of the Admin by shifting some features from read-only to editable access as well.

This project can be further improved by converting it into a Web Application using Python and the Django Framework or a Mobile Application using Flutter or React Native.

# REFERENCES

1. C Language Documentation: [https://docs.microsoft.com/en-us/cpp/c- language/?view=msvc-160](https://docs.microsoft.com/en-us/cpp/c-language/?view=msvc-160)
2. Stack Overflow (for debugging errors): <https://stackoverflow.com/>

3. All other websites of C programs like: <https://www.windows-commandline.com/delete-directory-from-command-line/>

1. Tutorials Point: https://www.tutorialspoint.com/cprogramming/c\_file\_io.htm
2. GeeksforGeeks : https://www.geeksforgeeks.org/c-programming-language/