Course Name: Secure Programming

Assignment 11: Input Validation.

Description of code working:

Initially the real user id of the user running the program is retrieved. Then it will check if “AuditFile.txt” file which is used to maintain log entries, exist or not. If exists, then it will modify the owner of the file as root and will set a permission that only owner can read and write log entries. If does not exist, then it will create AuditFile.txt and enters a text “Entries in this file will be in the format: TimeStamp Real-UserId Command Status Additional Details” that describes the format of log entries that are going to be added. Then changes the owner as root and permissions as 600.

Then it will try to connect to sqllite mydatabase.db. Then If the table does not exist then it will create a table to store name and telephone number. Then it will check whether the arguments are less than 2 then it will display the help, listing out the command and their formats.

Else it will check whether the command is “ADD”. If so, then it will check whether exactly two more arguments with name and telephone number are passed or not. If not, then help of “ADD” command will be displayed. Then it will check the format of name and if it satisfies the rules then the telephone number format will be checked. If not, then a message that “Invalid Name Format!!” will be displayed. If telephone number format is not valid then “Invalid Contact Number” will be displayed. If it is valid then that record will be added to telephone directory using the prepared sql command. Then an entry to audit log file is added.

Else it will check whether the command is “DEL”. If so, then it will check whether exactly one more argument is passed, if not then “DEL” command help will be displayed. If exactly only one more argument along with “DEL” command is passed, then it will check whether passed argument is valid number format. If so, then this value will be passed to prepared sql delete statement, using which a record will be removed, if exists. If it does not exist, then a message will be displayed saying that it tries to delete a number that does not exist in the directory. In either case an entry to audit log file is made. If the passed argument is not a valid number format, then it will check if the argument is valid name. If so then this value will be passed, to prepared sql delete statement, using which a record will be removed, if exists. If it does not exist, then a message will be displayed saying that it tries to delete a name that does not exist in the directory. In either case an entry to audit log file is made. If the passed argument is not a valid name or number, then a message invalid format will be displayed.

Else it will check whether the command is “LIST”. If so, then it will check whether any more argument is passed along with it, if so then “LIST” command help will be displayed. If not, then it will try to display all the records of telephone directory that exist up to then. And an entry to audit log file is made.

Further description of program is available in “Additional Details, Execution steps along with screenshots:” section.

Compilation/build Instructions:

As it is a python file, no additional compilation or build instructions are needed.

To make it a set-uid privileged program, modified the owner of program by running “sudo chown root test.py” command. Then modified its privileges by running “sudo chmod 4755 test.py” command.

Installation setup and execution instructions:

No additional setup id was made. Used the existing SEED labs setup alone.

Execution Instructions:

After modifying the program as set-uid program as mentioned in “Compilation/build Instructions” section. The program can be executed by running “python test.py <arguments>” command. The command and its argument name and telephone number will be passed by enclosing them in double quotes.

Examples:

- python test.py “ADD” “User” “12345.12345”
- python test.py “DEL” “User”
- python test.py “DEL” “12345.12345”
- python test.py “LIST”

Assumptions made:

Following assumptions are made about input name:

Below Naming formats are supported:

- <FirstName MiddleName LastName>
- <FirstName LastName>
- <LastName, FirstName MiddleInitial>
- <LastName, FirstName MiddleName>
- <LastName, FirstName>
- <FirstName>

Below Naming convention rules are followed:

1. In all the supported formats FirstName is mandatory. It consists of only alphabets and no special characters are allowed. It starts with capital letter followed by small letters.
2. Few formats mentioned above include MiddleName. It consists of only alphabets and no special characters are allowed. It starts with capital letter followed by small letters.
3. Few formats mentioned above include MiddleInitial. It consists of one capital letter followed by “.”.
4. Few formats mentioned above include LastName. It allows below formats.

  - It can consist of only alphabets that starts with capital letter followed by small letters.
  - It can consist of only alphabets and "O'" that starts with "O'" followed by capital letter followed by small letters.
  - It can consist of only alphabets, "O'“ and "-" that starts with "O'" followed by capital letter followed by small letters followed by "-" followed by capital letter followed by small letters.

Following assumptions are made about input number:

Along with rules mentioned in the assignment sheet, below assumptions were made:

- If “+” is not used, then “011” and “00” code will be used for international numbers.
- Assumed that any subscriber number will not start with “0”.
- Assumed that any area code will not start with “0”.
- Assumed that “US” international number’s area code will not start with 0 and 1. And it will not contain 9 as middle digit in area code.

General Assumptions:

- Assumed that there is no need to display “1 record is successfully added” message after successful execution of “ADD” command.
- Assumed that there is no need to display “1 record is successfully removed” message after successful execution of “DEL” command.
- Assumed that there is no need to display “No records exist “message after successful execution of “LIST” command on empty telephone directory.
- Assumed that there is no need to make an entry to audit log file if the passed input argument format does not satisfy the rules and assumption made.

Pros/Cons of my approach:

Pros:

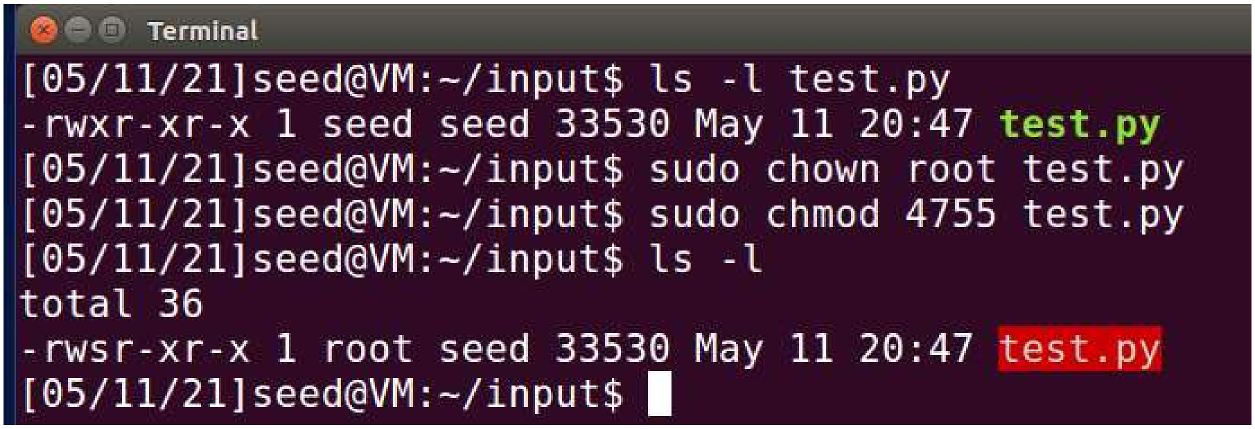
- Used sqllite database to store the telephone directory records.
- Used prepared sql queries.
- Used privileged mode audit log functionality.
- Code is simpler and more straightforward to understand and debug.
- Adapted whitelisting to check for valid inputs.

Cons:

- Implemented entire functionality using a single python program. It is not split into multiple programs each handling one responsibility.
- Used a number format checks.

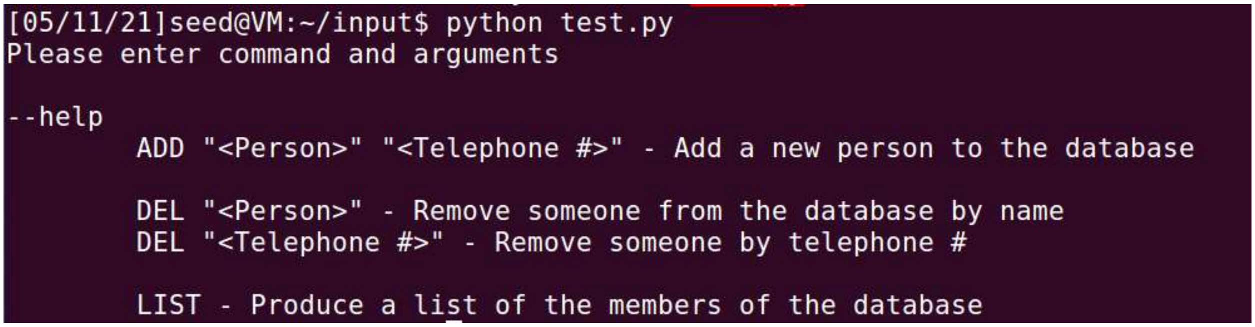
Additional Details, Execution steps along with screenshots:

Changed the owner of program as “root” and changed the privileges to “4755”. Checked the access using “ls -l input.py”



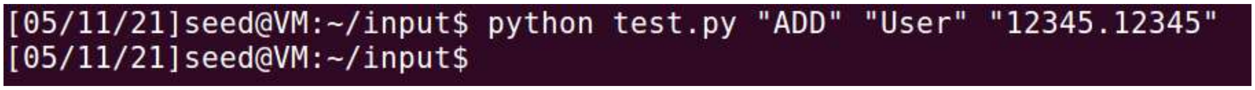
```
[05/11/21]seed@VM:~/input$ ls -l test.py
-rwxr-xr-x 1 seed seed 33530 May 11 20:47 test.py
[05/11/21]seed@VM:~/input$ sudo chown root test.py
[05/11/21]seed@VM:~/input$ sudo chmod 4755 test.py
[05/11/21]seed@VM:~/input$ ls -l
total 36
-rwsr-xr-x 1 root seed 33530 May 11 20:47 test.py
[05/11/21]seed@VM:~/input$ 
```

When no arguments are passed then it will display help along with a message “Please enter command and arguments”



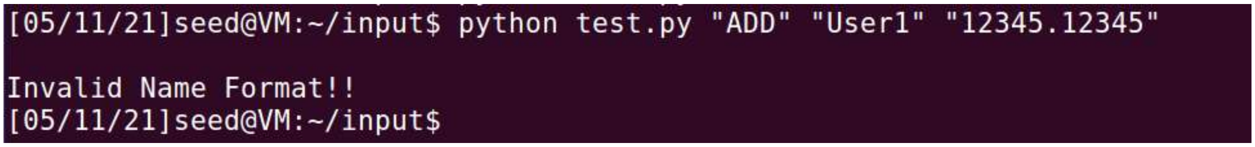
```
[05/11/21]seed@VM:~/input$ python test.py
Please enter command and arguments

--help
        ADD "<Person>" "<Telephone #>" - Add a new person to the database

        DEL "<Person>" - Remove someone from the database by name
        DEL "<Telephone #>" - Remove someone by telephone #

        LIST - Produce a list of the members of the database
```

When “ADD” command with proper name and telephone number are passed then it will be added to the database.



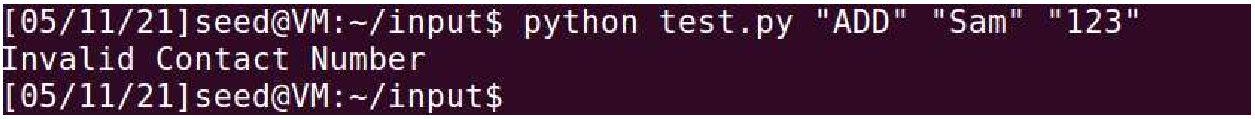
```
[05/11/21]seed@VM:~/input$ python test.py "ADD" "User" "12345.12345"
[05/11/21]seed@VM:~/input$
```

When “ADD” command with improper name format is passed then it will display a message that “Invalid Name Format!!”.



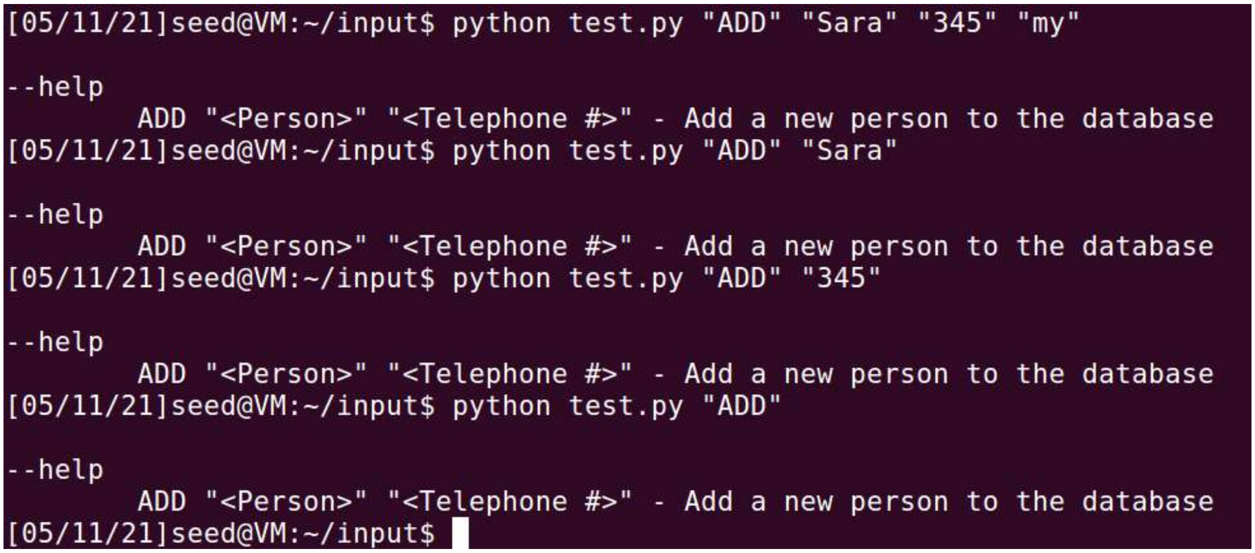
```
[05/11/21]seed@VM:~/input$ python test.py "ADD" "User1" "12345.12345"

Invalid Name Format!!
[05/11/21]seed@VM:~/input$
```

When “ADD” command with improper telephone number format is passed then a message “Invalid Contact Number” will be displayed.



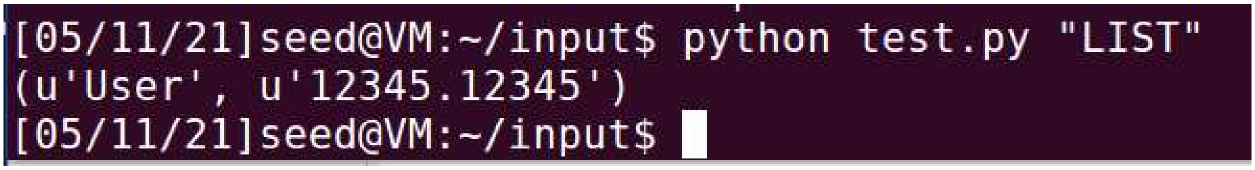
```
[05/11/21]seed@VM:~/input$ python test.py "ADD" "Sam" "123"
Invalid Contact Number
[05/11/21]seed@VM:~/input$
```

When “ADD” command with no arguments or with one or more than two arguments are passed then “ADD” command help will be displayed as shown in below screenshot.



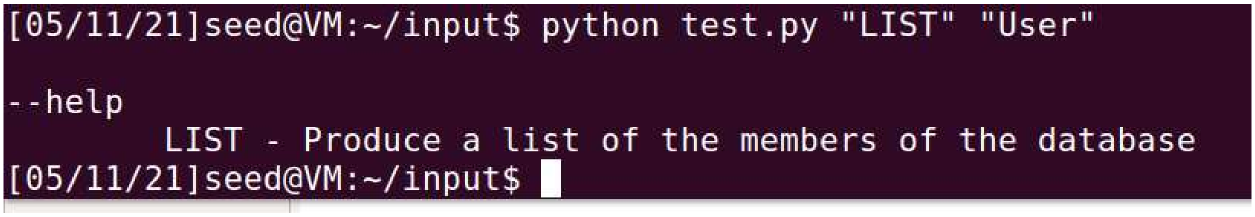
```
[05/11/21]seed@VM:~/input$ python test.py "ADD" "Sara" "345" "my"

--help
        ADD "<Person>" "<Telephone #>" - Add a new person to the database
[05/11/21]seed@VM:~/input$ python test.py "ADD" "Sara"

--help
        ADD "<Person>" "<Telephone #>" - Add a new person to the database
[05/11/21]seed@VM:~/input$ python test.py "ADD" "345"

--help
        ADD "<Person>" "<Telephone #>" - Add a new person to the database
[05/11/21]seed@VM:~/input$ python test.py "ADD"

--help
        ADD "<Person>" "<Telephone #>" - Add a new person to the database
[05/11/21]seed@VM:~/input$
```

When “LIST” command is passed then the list of records that were added up to then will be displayed.



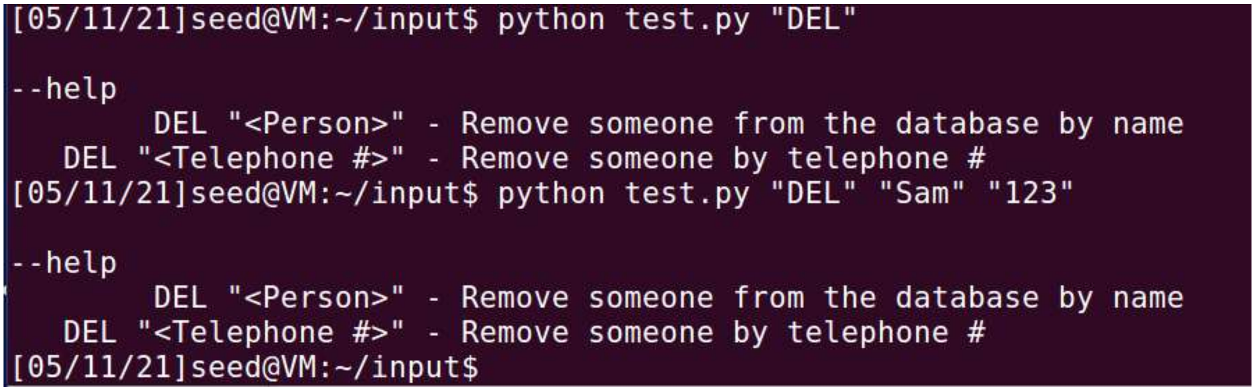
```
[05/11/21]seed@VM:~/input$ python test.py "LIST"
(u'User', u'12345.12345')
[05/11/21]seed@VM:~/input$
```

When “LIST” command is passed with any arguments as shown in below screenshot is passed then it will display the “LIST” command help.



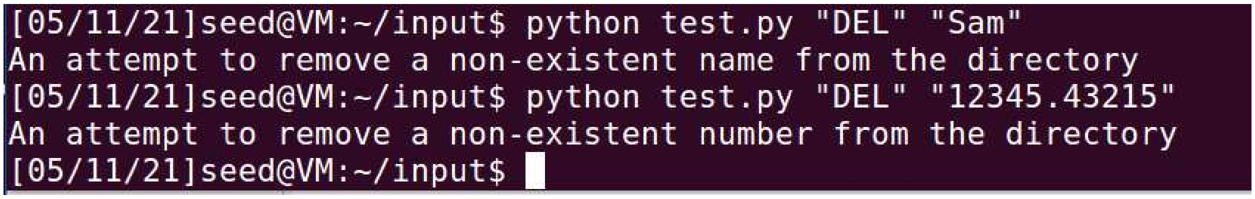
```
[05/11/21]seed@VM:~/input$ python test.py "LIST" "User"

--help
        LIST - Produce a list of the members of the database
[05/11/21]seed@VM:~/input$
```

When “DEL” command with no arguments or more than one argument is passed then “DEL” command help will be displayed.



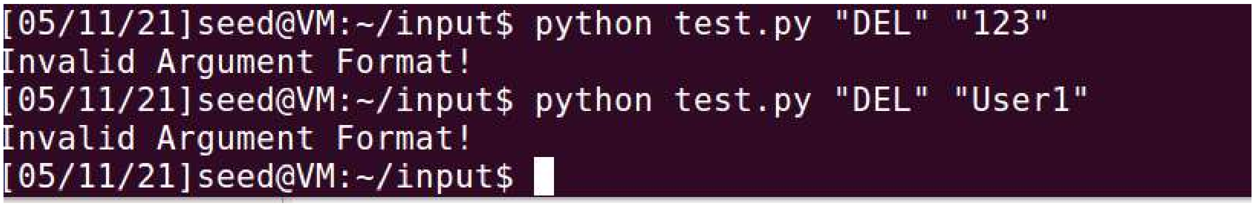
```
[05/11/21]seed@VM:~/input$ python test.py "DEL"

--help
        DEL "<Person>" - Remove someone from the database by name
   DEL "<Telephone #>" - Remove someone by telephone #
[05/11/21]seed@VM:~/input$ python test.py "DEL" "Sam" "123"

--help
        DEL "<Person>" - Remove someone from the database by name
   DEL "<Telephone #>" - Remove someone by telephone #
[05/11/21]seed@VM:~/input$
```

When “DEL” command along with either name or number that does not exist is passed then it displays message that either “An attempt to remove a non-existent name from the directory” or “An attempt to remove a non-existent number from the directory” will be displayed as shown in below screen-sot.



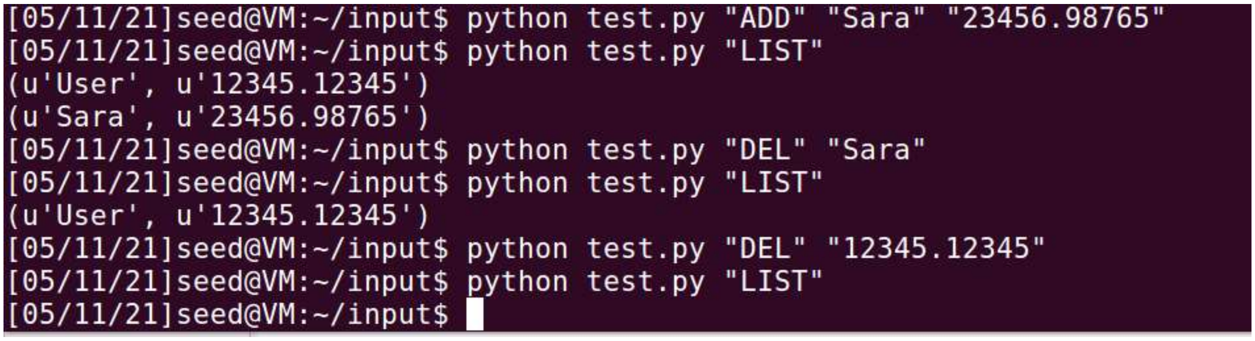
```
[05/11/21]seed@VM:~/input$ python test.py "DEL" "Sam"
An attempt to remove a non-existent name from the directory
[05/11/21]seed@VM:~/input$ python test.py "DEL" "12345.43215"
An attempt to remove a non-existent number from the directory
[05/11/21]seed@VM:~/input$ 
```

When “DEL” command along with invalid input format is passed as argument then “Invalid Argument Format!” message will be displayed as shown in below screenshot.



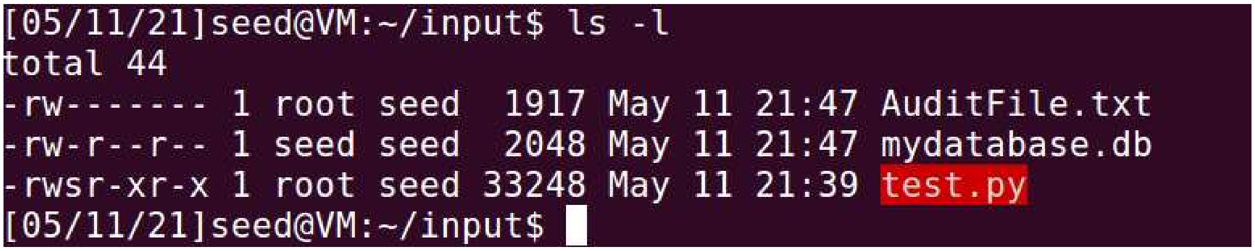
```
[05/11/21]seed@VM:~/input$ python test.py "DEL" "123"
Invalid Argument Format!
[05/11/21]seed@VM:~/input$ python test.py "DEL" "User1"
Invalid Argument Format!
[05/11/21]seed@VM:~/input$ 
```

Added another record to demonstrate “DEL” command by passing name and telephone number. Checked the list of existing records using “LIST”. Demonstrated “DEL” command by passing name “Sara” as argument. Then to check if the record with name Sara is removed or not used “LIST” command to check the existing records. Demonstrated “DEL” command by passing telephone number “12345.12345”. Then to check if the record is removed or not used “LIST” command and noticed that no records are returned confirming that the record is removed successfully.



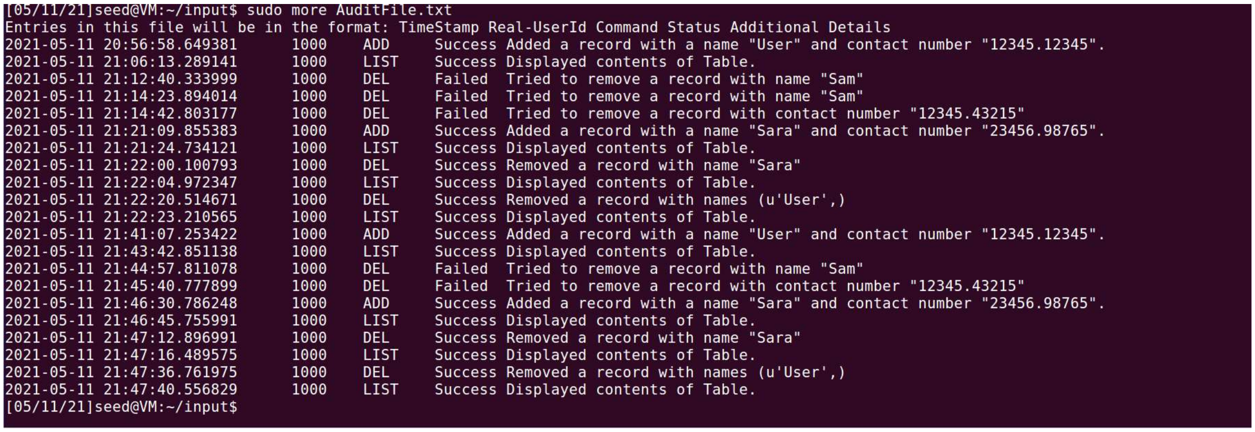
```
[05/11/21]seed@VM:~/input$ python test.py "ADD" "Sara" "23456.98765"
[05/11/21]seed@VM:~/input$ python test.py "LIST"
(u'User', u'12345.12345')
(u'Sara', u'23456.98765')
[05/11/21]seed@VM:~/input$ python test.py "DEL" "Sara"
[05/11/21]seed@VM:~/input$ python test.py "LIST"
(u'User', u'12345.12345')
[05/11/21]seed@VM:~/input$ python test.py "DEL" "12345.12345"
[05/11/21]seed@VM:~/input$ python test.py "LIST"
[05/11/21]seed@VM:~/input$ 
```

In the below screen-sot we can notice that AuditFile.txt which is used to maintain logs is owned by root and only root has access to read or write this file. Thus, securing it from allowing anyone to access or modify it.



```
[05/11/21]seed@VM:~/input$ ls -l
total 44
-rw------- 1 root seed  1917 May 11 21:47 AuditFile.txt
-rw-r--r-- 1 seed seed  2048 May 11 21:47 mydatabase.db
-rwsr-xr-x 1 root seed 33248 May 11 21:39 test.py
[05/11/21]seed@VM:~/input$ 
```

In the below screenshot, we can notice the log entries in “AuditFile.txt”. Each log entry will consist of Timestamp, Real User Id, Command, Status, Additional Information. For “ADD” command, the name and telephone number of records that is added to directory is mentioned. For successful “DEL” command the name of the record that was removed, is mentioned. For failed “DEL” command, in additional information, the name or telephone number that was tried to remove is mentioned.



```
[05/11/21]seed@VM:~/input$ sudo more AuditFile.txt
Entries in this file will be in the format: TimeStamp Real-UserId Command Status Additional Details
2021-05-11 20:56:58.649381      1000    ADD     Success Added a record with a name "User" and contact number "12345.12345".
2021-05-11 21:06:13.289141      1000    LIST    Success Displayed contents of Table.
2021-05-11 21:12:40.333999      1000    DEL     Failed  Tried to remove a record with name "Sam"
2021-05-11 21:14:23.894014      1000    DEL     Failed  Tried to remove a record with name "Sam"
2021-05-11 21:14:42.803177      1000    DEL     Failed  Tried to remove a record with contact number "12345.43215"
2021-05-11 21:21:09.855383      1000    ADD     Success Added a record with a name "Sara" and contact number "23456.98765".
2021-05-11 21:21:24.734121      1000    LIST    Success Displayed contents of Table.
2021-05-11 21:22:00.100793      1000    DEL     Success Removed a record with name "Sara"
2021-05-11 21:22:04.972347      1000    LIST    Success Displayed contents of Table.
2021-05-11 21:22:20.514671      1000    DEL     Success Removed a record with names (u'User',)
2021-05-11 21:22:23.210565      1000    LIST    Success Displayed contents of Table.
2021-05-11 21:41:07.253422      1000    ADD     Success Added a record with a name "User" and contact number "12345.12345".
2021-05-11 21:43:42.851138      1000    LIST    Success Displayed contents of Table.
2021-05-11 21:44:57.811078      1000    DEL     Failed  Tried to remove a record with name "Sam"
2021-05-11 21:45:40.777899      1000    DEL     Failed  Tried to remove a record with contact number "12345.43215"
2021-05-11 21:46:30.786248      1000    ADD     Success Added a record with a name "Sara" and contact number "23456.98765".
2021-05-11 21:46:45.755991      1000    LIST    Success Displayed contents of Table.
2021-05-11 21:47:12.896991      1000    DEL     Success Removed a record with name "Sara"
2021-05-11 21:47:16.489575      1000    LIST    Success Displayed contents of Table.
2021-05-11 21:47:36.761975      1000    DEL     Success Removed a record with names (u'User',)
2021-05-11 21:47:40.556829      1000    LIST    Success Displayed contents of Table.
[05/11/21]seed@VM:~/input$
```