CSCI3240 Lab 4: File Handling

Objective: To understand the concept of file handling.

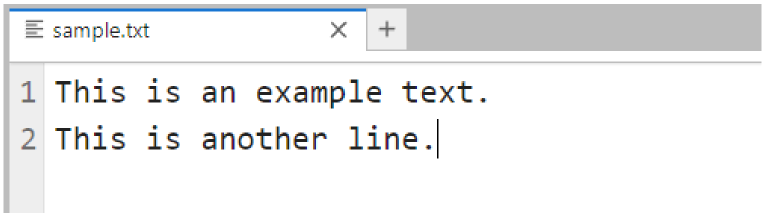
Please check the following link before you start:

- *https://www.tutorialspoint.com/basics-of-file-handling-in-c*

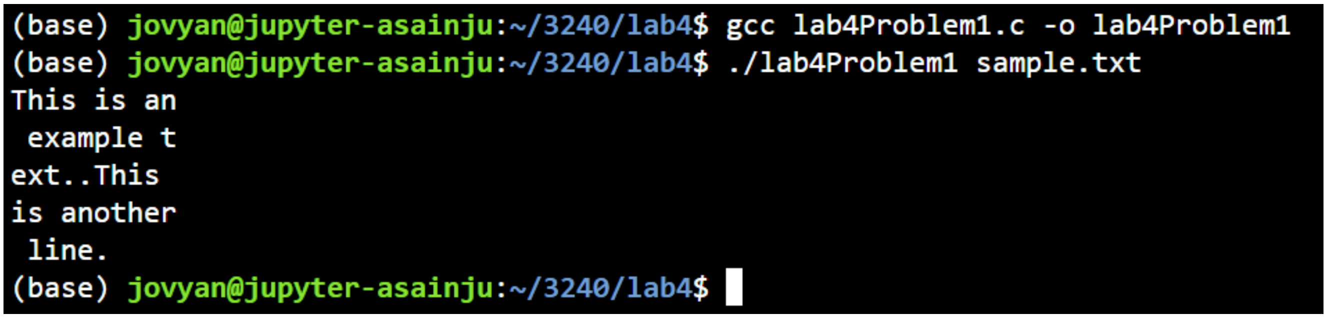
1. Write a program "***lab4Problem1.c***" to perform the following tasks:
   a. Read a file character by character and store it in an array of characters of size 11 (the last index should contain "\0").
      i. Make sure to check if the character is printable before putting it in the array.
      ii. If the character is not printable, put '.' instead. (Hint: Use isprint function)
      iii. Your array should be able to store 10 characters from the file and one null terminating character "\0". (i.e. the size of the array should be 11). The character array with '\0'at the end is considered string in c.
      iv. If the array is full, print the string with a newline character at the end. (Hint: You can use "%s" format specifier to print string in printf.)
      v. Continue to fetch next 10 set of characters from the file and repeat step i to v.
      vi. In case of end of the file (**eof**), print the string with a newline character and exit. Make sure the index after the last element on the array contains '\0' to indicate the end of string.
   b. Your program must get the filename as a command-line argument.

isprint: https://www.tutorialspoint.com/c_standard_library/c_function_isprint.htm

Sample Run



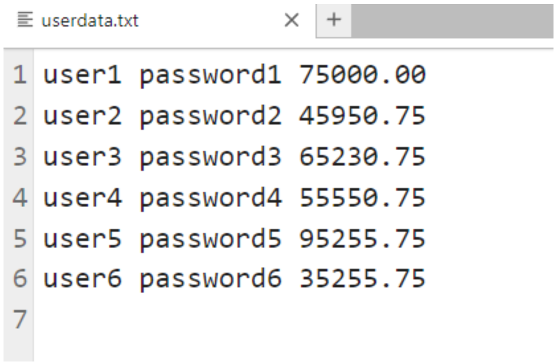
```
This is an example text.
This is another line.
```



```
(base) jovyan@jupyter-asainju:~/3240/lab4$ gcc lab4Problem1.c -o lab4Problem1
(base) jovyan@jupyter-asainju:~/3240/lab4$ ./lab4Problem1 sample.txt
This is an
 example t
ext..This
is another
 line.
(base) jovyan@jupyter-asainju:~/3240/lab4$ 
```

2. Write a program “lab4Problem2.c” that takes username and password from the user. Your program should compare the input to the pre-stored username and password in the “userdata.txt” file.

   Sample Content of userdata.txt:

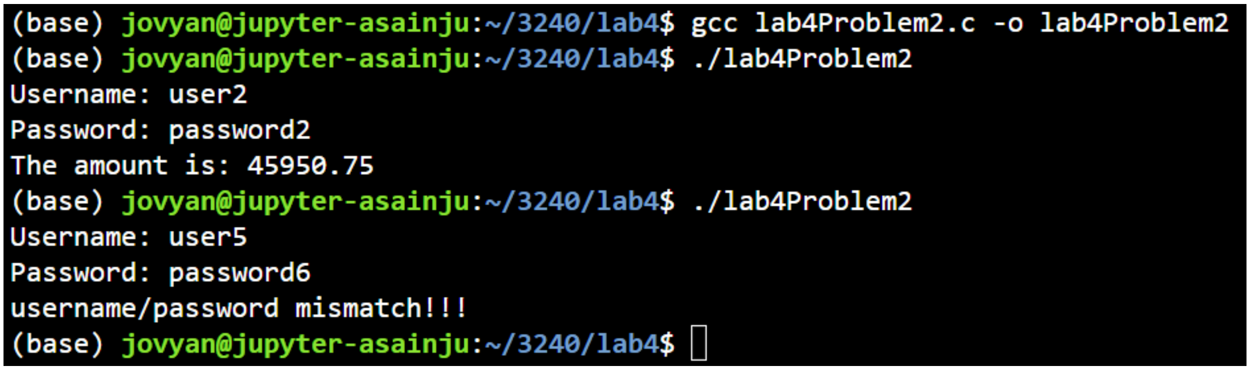


```
user1 password1 75000.00
user2 password2 45950.75
user3 password3 65230.75
user4 password4 55550.75
user5 password5 95255.75
user6 password6 35255.75

```

   a. If the username and password match with any row in the **userdata.txt** file, display the numerical data associated with that row.
   b. Else, print the **“username/password mismatch!!!”** message.

   Sample Run



```
(base) jovyan@jupyter-asainju:~/3240/lab4$ gcc lab4Problem2.c -o lab4Problem2
(base) jovyan@jupyter-asainju:~/3240/lab4$ ./lab4Problem2
Username: user2
Password: password2
The amount is: 45950.75
(base) jovyan@jupyter-asainju:~/3240/lab4$ ./lab4Problem2
Username: user5
Password: password6
username/password mismatch!!!
(base) jovyan@jupyter-asainju:~/3240/lab4$ 
```

   Hint: *Use sscanf function in c to break down the line into 3 components: username, password, and amount*
   https://www.tutorialspoint.com/c_standard_library/c_function_sscanf.htm

Rubrics (Total 100 points):

Problem1 (50 points):

Criteria	**Points**
The program tries to print non-printable characters/does not print	-5
The program does not read till the End of File (EOF)	-10
The program does not print anything when the array is full.	-10

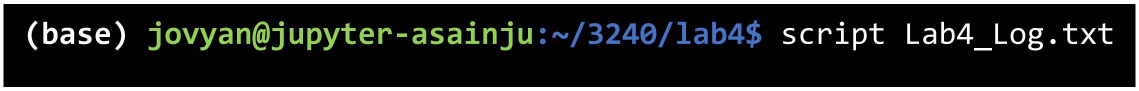
The file is not closed before the end of the program	-5
The program does not print the newline as a character (should be printed as “.”)	-5
The program does not get the filename as a command line argument	-15
The program does not print the file content in the requested format	-30
The source code is not named correctly (it should be lab4Problem1.c)	-10
No submission or source code is missing	-50

Problem2 (50 points):

Criteria	**Points**
The file is not closed before the end of the program	-5
The program incorrectly allows password match	-15
The program does not print associated numerical data after a credential match	-10
The program prints incorrect numerical data after the correct credential match	-5
The program is not able to read any valid input of maximum length	-5
The source code is not named correctly (it should be lab4Problem2.c)	-10
No submission or source code is missing	-50

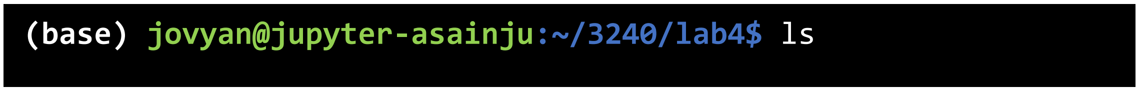
Steps to Create the Log File:

1. **Open your terminal and start the scripting process by typing:**



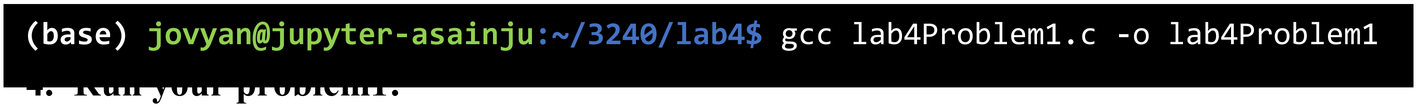
```
(base) jovyan@jupyter-asainju:~/3240/lab4$ script Lab4_Log.txt
```

2. **List all the files in the current directory:**



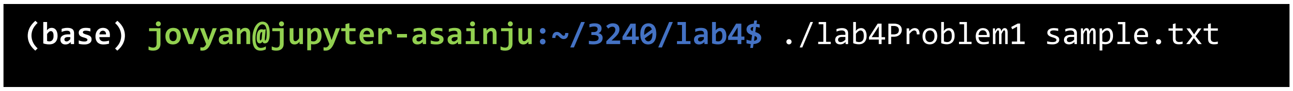
```
(base) jovyan@jupyter-asainju:~/3240/lab4$ ls
```

3. **Compile your problem 1:**



```
(base) jovyan@jupyter-asainju:~/3240/lab4$ gcc lab4Problem1.c -o lab4Problem1
```

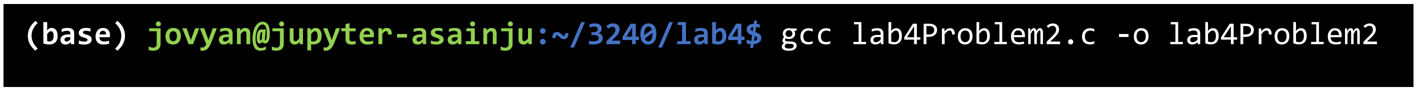
4. **Run your problem1:**



```
(base) jovyan@jupyter-asainju:~/3240/lab4$ ./lab4Problem1 sample.txt
```

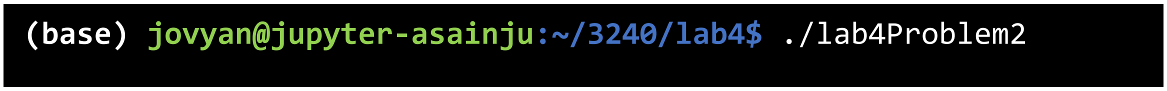
Note: You can test with multiple other input values here.

5. **Compile problem 2:**

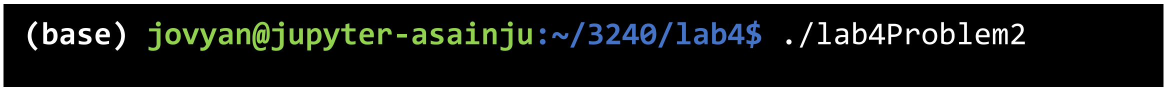


```
(base) jovyan@jupyter-asainju:~/3240/lab4$ gcc lab4Problem2.c -o lab4Problem2
```

6. **Run your problem 2:**



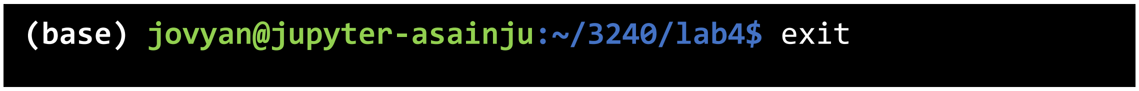
```
(base) jovyan@jupyter-asainju:~/3240/lab4$ ./lab4Problem2
```



```
(base) jovyan@jupyter-asainju:~/3240/lab4$ ./lab4Problem2
```

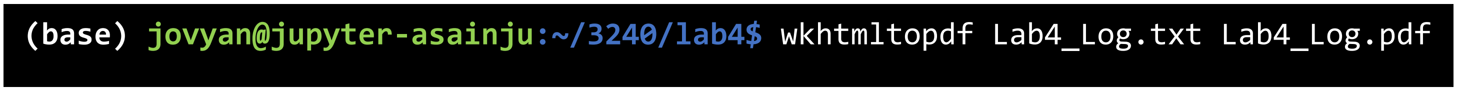
Note: You can test with multiple other input values here.

7. **Exit the scripting process to finish and save the log file:**



```
(base) jovyan@jupyter-asainju:~/3240/lab4$ exit
```

8. **Convert the log file from txt to pdf using the txt created from using script**



```
(base) jovyan@jupyter-asainju:~/3240/lab4$ wkhtmltopdf Lab4_Log.txt Lab4_Log.pdf
```

The ‘`Lab4_Log.pdf` ’ file will be generated in your current directory. Make sure this file is included in your submission.

Submission Instructions:

Please upload the following files:

- Lab4Problem1.c
- Lab4Problem2.c
- Lab4_Log.pdf
- AI_Disclaimer.pdf

- **Submission Due:** Check Lab 4 Dropbox