Question:1

```
def register():
    username = input("please input the first 2 letters of
                your first name & your birth day")
    password = input("please input your desired password")
    file = open("loginfile.txt", "a")
    file.write(username)
    file.write(" ")
    file.write(password)
    file.write("\n")
    file.close()
    if login():
        print("you are now logged in...")
    else:
        print("you aren't logged in!")

def login():
    username = input("please enter your username:")
    password = input("please enter your password:")
    for line in open("loginfile.txt", "r").readlines():
        if username == login_info[0] and password ==
                                        login_info[1]:
            print("correct credentials!")
            return True
    print("Incorrect credentials.")
    return False.

register()
login()
```

Problem Solving Using Python and R Lab

Lab6. Python File Processing

Question1. Write a program for *Password Management System*

- File creation: Ask user to enter N user names and their passwords. Store usernames and passwords into a file named **"loginfile.txt".** Store each user and password in one line.
- File Processing: Write a program that opens your "security.txt" file and reads usernames and passwords from it. Store user names in one list and passwords in another lists.
- Querying: ask user to enter user name and password for verification. If they match the values stored in the lists, print a message "Login Successful". Otherwise print a message "Login Failed, try again".

Question 2:

```
marks = [99, 0, 100.0, 95.0, 96.0, 97.0]
with open('marks1.txt', 'a') as file:
    for mark in marks:
        file.write("%.1f\n" % mark)
    number_list = []
with open('marks1.txt', 'r') as fp:
    number_list = [float(item) for item in fp.readlines()]
print(max(number_list))

def Nmaxelements(list1, N):
    final_list = []
    for i in range(0, N):
        max1 = 0
        for j in range(len(list1)):
            if list1[j] > max1:
                max1 = list1[j];
        list1.remove(max1);
        final_list.append(max1)
    print(final_list)
Nmaxelements(number_list, 3)

def Nminelements(list1, N)
    final_list = [];
    for i in range(0, N):
        min1 = 9999999;
        for j in range(len(list1)):
            if list1[j] < min1:
                min1 = list1[i];
        list1.remove(min1);
        final_list.append(min1)
    print(final_list)
Nminelements(number_list, 3)
```

Question2. Write a program for *Student Performance Analysis*

- Create a text file, 'marks.txt', with **N** marks as floating point numbers. Open the file, read marks from it and compute and print the highest mark.
- If the user runs the program more than once you should not overwrite the previous text file – simply append the marks to the end of the file.
- Modify the above program so that it also prints **Top-3** highest marks (Note: you may need to use list concept)
- Modify the above program so that it also prints the **Lowest-3** marks.

Question: 3

*

```
While True:
        St_name = str(input("Enter the name: "))
        file = open ("stock_prices.txt", "a")
        file.write (st_name)
        file.write (" ")
        for i in range (5):
            p = input()
            file.write(p)
            file.write (" ")
        file.write ("\n")
        con = str(input("want to continue: "))
        if con == 'n':
            break
        file.close()
```

*

```
for st in open("stock_prices.txt", "r").readline():
    p_min = []
    calc = st.split()
    print (calc[0])
    for i in range (1, 6):
        p_min.append (int (calc[i]))
    print (min(p_min))
    print (max(p_min))
    av = sum(p_min)
    avg = av/5
```

Question3. Write a program for *Stock Price Analysis*

- File Creation: Continually prompt a user for stock name, followed by price values for 5 days. Each row indicates stock name and daily prices of one stock. Store these values in a text file called **"stock-prices.txt".** Open the file in Append Mode. Prompt message "Do you want to continue? " and stop reading values accordingly. Then, you can close your file.
- File Processing: Now, open your file for processing. Print stock name, minimum price, maximum price and average price values.
- You can also print which day stock price was lowest in the week and which day stock price was highest. So, modify your print statement to print stock name, minimum price & day of minimum price, maximum price & day of maximum price and average price values. (Hint: Use enumerate to get index values)

```
print (avg)
print ("\n")
```

*

```
for st in open ("stock-prices.txt", "r").readlines():
    p_min = []
    print ("--------")
    calc = st.split()
    print (calc[0])
    for i in range (1,6):
        p_min.append (int(calc[i]))
    mip = min(p_min)
    mxp = max (p_min)
    im = p_min.index (mip)
    ix = p_min.index (mxp)
    print ("min price", mip, "on day", im+1)
    print ("max price", mxp, "on day", ix+1)
```

```
print ("1. Display the contents of file:")
print ("------")
f = open ("samplemv.txt", 'r')
display = f.read()
print (display)
f.close()
print ("")
print ("2. Count the number of lines in a text file:")
print ("------")
f = open ("samplemv.txt", 'r')
counter = 0
content
counter = file.read()
colist = content.split ("\n")
for i in colist:
    if i:
        counter += 1
print("Number of lines in the text file:", counter)
print ("\n")
```

Question4. Write a program for File Explorer

- Display the contents of file
- Count the number of lines in a text file. (Use splitlines())
- Count the number of unique words in a file.
- Find frequency of words in a given file. (Hint: Use Counter object)
- Show a random line in a file. (Use Random object)

```
print("3. Count the number of unique words in a file:")
print("------")
num_words = 0
c = open("Sample mv.txt", 'r')
for line in c:
    words = line.split()
    num_words += len(words)
print("Number of words:", num_words)
c.close()
print("\n")
print("4. Find Frequency of words in a given file:")
print("-----")
fname = input('Enter the file name:')
print("----")
try:
    fhand = open(fname)
    counts = dict()
    for line in fhand:
        words = line.split()
        for word in words:
            if word in counts:
                counts[word] += 1
            else:
                counts[word] = 1
```

Question 5:

~~from (~~

```
fhand = open ('mbox-short.txt')
for line in fhand:
    line = line.rstrip()
    if line.startswith ('From '):
        print (line)

fhand = open ('mbox-short.txt')
count = 0
for line in fhand:
    line = line.rstrip()
    if line == "": continue
    words = line.split()
    if words[0] != "From": continue
    print (words[1])
    count = count + 1
print ("There were", count, "lines in the file with
from as the first word")
```

Question5. [File Searcher]. Develop an application in Python to read through the email data (**"mbox-short.txt"**) and when you find line that starts with "From", you will split the line into words using the split function. We are interested in who sent the message, which is the second word on the From line: From stephen.marquard@uct.ac.za Sat Jan 5 09:14:16 2008. You will parse the From line and print out the second word for each From line, then you will also count the number of From (not From:) lines and print out a count at the end

Question 6:

```
from csv import writer
def append_list_as_row (file_name, list_of_elem):
    with open('student_marks.csv', 'a+', newline='')
                                        as write_obj:
        csv_writer = writer(write_obj)
        csv_writer.writerow (list_of_elem)
row_contents = ['Suresh', 68, 78, 89, 87, 90]
row_contents1 = ['ganesh', 68, 78, 89, 87, 90]
row_contents2 = ['Harish', 68, 78, 89, 87, 90]
row_contents3 = ['Rajesh', 68, 78, 89, 87, 90]
append_list_as_row ('student_marks.csv', row_contents)
append_list_as_row ('student_marks.csv', row_contents1)
append_list_as_row('student_marks.csv', row_contents2)
append_list_as_row ('student_marks.csv', row_
                                        contents3)
```

Question6. Write a program to read and write CSV files

- File Creation: Create MS Excel file **("student_marks.csv")** with 5 rows of student name, mark1, mark2, mark3, mark4. Use comma to separate each value in a row.
- File Display: Now, open your CSV file and display the file contents row by row (More information at: https://docs.python.org/3/library/csv.html).
- File Writing: Now, open **("student_marks.csv")** for writing. Ask user to enter name followed by 4 marks for one new student and write them onto the file.

```
import csv
with open('student_marks.csv', newline = '') as csvfile:
    reader = csv.reader(csvfile, delimiter = ',',
                        quotechar = '|')
    for row in reader:
        print(', '.join(row))
```