IF100 – Spring 2020-2021
Take-Home Exam 2
Due April 21st, 2021, Wednesday, 23:59 (Sharp Deadline)

Introduction

The aim of this take-home exam is to practice decision making (conditional if statements), sequences, and methods. The use of if statements and string methods are due to the nature of the problem; that is, you cannot finish this take-home exam without using them.

Description

Authentication is the process of determining whether someone is who declares herself/himself to be. Authentication technology provides access control for systems by checking if a user's credentials match the credentials stored in the database. There are many different types of procedures to store passwords but most of the time, credentials are not stored in plain text due to security concerns. One popular method used for storing passwords is scrambling the password.

Registration is the creation of an online account using a username and a password. The online account is usually for a website or web-based service. While creating the account, credentials are written to the database with the methods mentioned above.

For this take home exam you will write a Python program that will simulate a simple version of an authentication process by taking the following four (4) information as inputs:

- Usernames, together with corresponding scrambled passwords of users, in a predetermined format
- Usernames, together with the period of time from the last modification date of the password until now, in a predetermined format
- Username
- Password

By using the username and password given by the user (the third and the fourth inputs), your program should check if the user's credentials match the credentials given before. This process will start with checking if the username exists in the database which includes usernames, together with corresponding scrambled passwords of users, in a predetermined format. If so, then your program needs to

scramble the username and the password taken in order to check if the generated scrambled password matches with the stored one. If the username does not exist, then your program needs to give an error message.

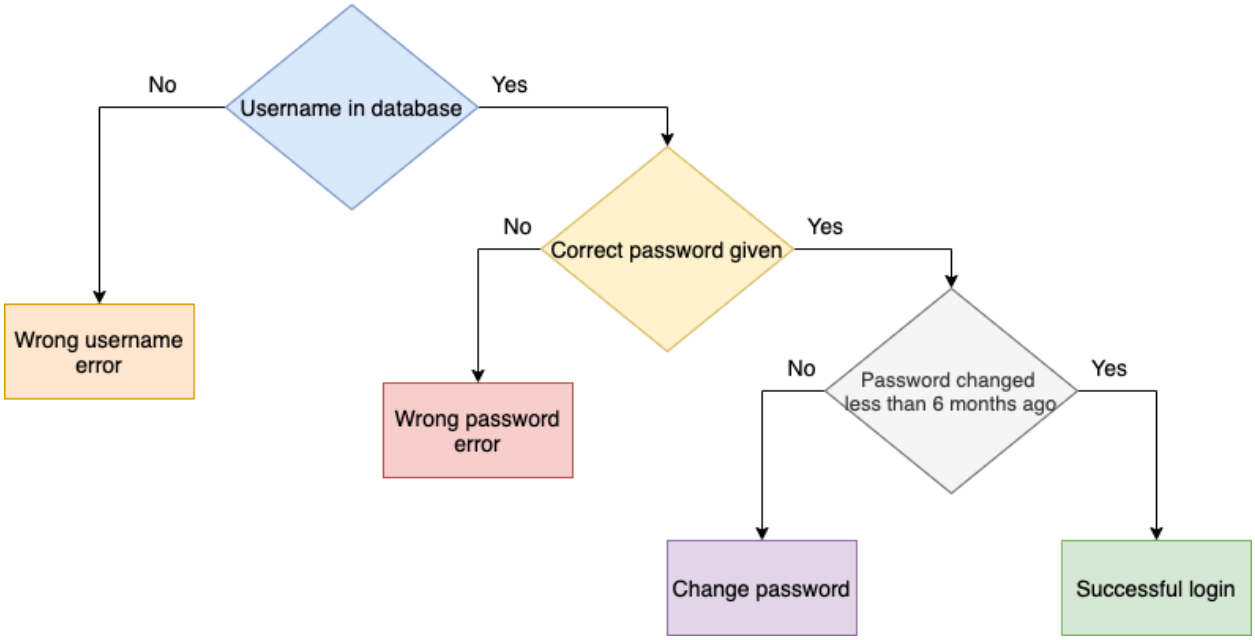
Scrambled password will consist of the concatenation of (i) the first letter of the username, (ii) the reverse version of the password, and (iii) the username without first letter. For example:

- username: **purplemonkey**
- password: **123456seven**
- scrambled password: **pneves654321urplemonkey**

After authentication, your program needs to check the last modification time of the password. The user requires to change her/his password if 6 or more months have passed. Your program also needs to update the password of the user in the database and the last modification time of the password for that particular user.

Please follow the pseudocode, and the flowchart given, for further details.

- If the username does not exist in the database, then your program needs to give an error message
- If the username exists in the database
    - If the scrambled password is not matching with the one stored in the database, then your program needs to give an error message
    - If the scrambled password is matching with the one stored in the database
        - If 6 months have not passed from the last modification date of the password, your program needs to give a message for a successful login
        - If 6 months have passed from the last modification date of the password, your program needs to ask for a new password from the user and update the password of the user in the database and the last modification time of the password for that particular user. This process will finish with printing both updated databases.



Inputs

The inputs of the program and their order are explained below. It is extremely important to follow this order with the same characters since we automatically process your programs. ***Thus, your work will be graded as 0 unless the order is entirely correct***. Please see the "Sample Runs" section for some examples.

The prompts of the input statements to be used have to be exactly the same as the prompts of the "Sample Runs".

Here is the detailed information on the inputs and the input checks:

- Usernames and scrambled passwords - *usernames and passwords list*
  Format: `username1:password1;...;usernameN:passwordN`
  i.e. **`"purplemonkey:pneves654321urplemonkey;john:j321dsaohn"`**
  - Colon (":") is used between each one of the usernames and passwords. First comes the username, and after the colon, there comes the scrambled password. The length of the username and password can be anything larger than 3. Therefore, you cannot make any further assumptions on the username or password length. For example, the username can be "j8hn" or "j@neDoe" or "purplemonkey", whose lengths are 4, 7, and 12, respectively. The password can be "EasyPa$$" or "pZ.3*v0" or "1234", whose lengths are 8, 7, and 4, respectively.

- Semicolons (";") are used between username and password pairs.
- The number of username and password pairs is not fixed, and there is no predefined or pre-assumed order in this input, either with respect to the usernames or passwords.
- You may assume that there won't be any duplicated usernames within this input. You may also assume that the username and the passwords will not contain ":" or ";" characters.
- You may check *Sample Runs,* for more examples. However, please ***keep in mind*** that sample runs may not cover all possible cases mentioned in this document.

- Usernames and last modification periods - *usernames and period list*
  Format: `username1:xyear(s) ymonth(s);...;usernameN:xyear(s) ymonth(s)`
  i.e. **"purplemonkey:1year(s) 2month(s);john:0year(s) 5month(s)"**
  - Colon (":") is used between each one of the usernames and time periods. First comes the username, and after the colon, there comes the time period. You may assume that all the usernames that exist in the first input will also be in this input. Rules and details about usernames are the same with the first input.
  - Semicolons (";") are used between username and time period pairs.
  - The number of username and time period pairs is not fixed, and there is no predefined or pre-assumed order in this input, either with respect to the usernames or time periods. Additionally, the order of the usernames in this input might be different from the order of the usernames given in the first input. However, you may assume that each user occurring in the first input also occurs in the second input as well.
  - Time period information is consisting of 2 different parts, year and month. Example format for period information is `Yyear(s) Mmonth(s)` where `Y` and `M` values will always be entered as non-negative integers. Both of these values could be 0. You can calculate 1 year as 12 months.
  - You may assume that there won't be any duplicated usernames within this input. Again, you may assume that usernames will not contain ":" or ";" characters.
  - You may check *Sample Runs,* for more examples. However, please ***keep in mind*** that sample runs may not cover all possible cases mentioned in this document.

- Username - *username*
  Format: `username`
    - You may assume all usernames will be longer than 3 characters.
    - Usernames can consist of any printable characters except for colon (":") and semicolon (";").
- Password - *password*
  Format: `password`
    - You may assume all passwords will be longer than 3 characters.
    - Passwords can consist of any printable characters except for colon (":") and semicolon (";").
- New password - *new password*
  Format: `password`
    - Rules and details are the same as *password*.
    - New password will be taken from the user if 6 months passed from the last modification date of the password.

Output

If the `username` does not exist in the *usernames and passwords list,* then your program should display an error message saying "Wrong username".

If the `username` exists in the *usernames and passwords list,* but the scrambled password created from *username* and *password* is not matching with the database, then your program should display an error message saying "Wrong password".

In any erroneous case, your program should terminate after displaying the message and *without taking any further inputs* or *without displaying any other results*.

If the user enters the correct *username* and *password*, but 6 months did not pass from the last modification date of the password, then your program should display a message saying "Successful login".

If the user enters the correct *username* and *password*, but 6 months passed from the last modification date of the password, your program needs to ask for a new password from the user and update the password of the user in the *usernames and passwords list* with new scrambled password gendered by using *username* and *new password*. Your program also needs to update the last modification time of the password for that particular user in *usernames and period list* with "0year(s)

0month(s)". Then your program should display the updated *usernames and passwords list* and *usernames and period list.*

You may check the "Sample Runs" section given below for some examples.

Sample Runs

Below, we provide some sample runs of the program that you will develop. The *italic* and **bold** phrases are inputs taken from the user. You have to display the required information in the same order and with the same words and characters as below.

Sample Run 1

```
Please enter usernames and passwords:
eagle_man:e5t4r3e2w1qagle_man;red.rose01:rgn4egnAROed.rose01;rhinowar:r123456.neveshinowar
Please enter last modification information: eagle_man:0year(s) 2month(s);red.rose01:1year(s) 6month(s);rhinowar:10year(s) 6month(s)
Please enter the username: wolf
Wrong username
```

Sample Run 2

```
Please enter usernames and passwords:
coff33c4ke:cr0t4id@1goff33c4ke;j0mesb0nd7:jredips?etaraK0mesb0nd7;hubblebubble:h.sneil@_etucubblebubble
Please enter last modification information: coff33c4ke:2year(s) 5month(s);j0mesb0nd7:0year(s) 5month(s);hubblebubble:0year(s) 10month(s)
Please enter the username: j0mesb0nd7
Please enter the password: Karate?spide
Wrong password
```

Sample Run 3

```
Please enter usernames and passwords:
coff33c4ke:cr0t4id@1goff33c4ke;j0mesb0nd7:jredips?etaraK0mesb0nd7;hubblebubble:h.sneil@_etucubblebubble
Please enter last modification information: hubblebubble:0year(s) 10month(s);coff33c4ke:2year(s) 5month(s);j0mesb0nd7:0year(s) 5month(s)
Please enter the username: j0mesb0nd7
Please enter the password: Karate?spider
Successful login
```

Sample Run 4

Please enter usernames and passwords:
coff33c4ke:cr0t4id@lgoff33c4ke;j0mesb0nd7:jredips?etaraK0mesb0nd7;hubblebubble:h.sneil@_etucubblebubble
Please enter last modification information: ***hubblebubble:0year(s) 6month(s);coff33c4ke:2year(s) 5month(s);j0mesb0nd7:0year(s) 5month(s)***
Please enter the username: ***hubblebubble***
Please enter the password: ***cute_@liens.***
Please enter your new password: ***iRonMan4ever***
coff33c4ke:cr0t4id@lgoff33c4ke;j0mesb0nd7:jredips?etaraK0mesb0nd7;hubblebubble:hreve4naMnoRiubblebubble
hubblebubble:0year(s) 0month(s);coff33c4ke:2year(s) 5month(s);j0mesb0nd7:0year(s) 5month(s)

Sample Run 5

Please enter usernames and passwords:
beagle_man:belgaeBrepaPkcoReagle_man;red.rose01:rgn4egnAROed.rose01;rhinowar:r123456.neveshinowar;eagle_man:e5t4r3e2w1qagle_man
Please enter last modification information: ***beagle_man:0year(s) 5month(s);red.rose01:0year(s) 3month(s);eagle_man:1year(s) 5month(s);rhinowar:4year(s) 2month(s)***
Please enter the username: ***eagle_man***
Please enter the password: ***q1w2e3r4t5***
Please enter your new password: ***gl@di4t0r***
beagle_man:belgaeBrepaPkcoReagle_man;red.rose01:rgn4egnAROed.rose01;rhinowar:r123456.neveshinowar;eagle_man:er0t4id@lgagle_man
beagle_man:0year(s) 5month(s);red.rose01:0year(s) 3month(s);eagle_man:0year(s) 0month(s);rhinowar:4year(s) 2month(s)

Sample Run 6

Please enter usernames and passwords:
eagle_man:e5t4r3e2w1qagle_man;red.rose01:rgn4egnAROed.rose01;rhinowar:r123456.neveshinowar
Please enter last modification information: ***eagle_man:0year(s) 2month(s);red.rose01:1year(s) 6month(s);rhinowar:10year(s) 6month(s)***
Please enter the username: ***_man***
Wrong username

Sample Run 7

```
Please enter usernames and passwords:
inancarin:i654321nancarin;arin:a54321rin
Please enter last modification information: inancarin:0year(s)
5month(s);arin:0year(s) 6month(s)
Please enter the username: arin
Please enter the password: 12345
Please enter your new password: 12345
inancarin:i654321nancarin;arin:a54321rin
inancarin:0year(s) 5month(s);arin:0year(s) 0month(s)
```

Sample Run 8

```
Please enter usernames and passwords:
inancarin:i654321nancarin;arin:a54321rin
Please enter last modification information: inancarin:0year(s)
5month(s);arin:0year(s) 5month(s)
Please enter the username: Arin
Wrong username
```

Sample Run 9

```
Please enter usernames and passwords:
eagle_man:e5t4r3e2w1qagle_man;red.rose01:rgn4egnAROed.rose01;rhinowar:r123
456.neveshinowar
Please enter last modification information: rhinowar:10year(s)
6month(s);eagle_man:0year(s) 2month(s);red.rose01:1year(s) 6month(s)
Please enter the username: eagle_man
Please enter the password: ORAnge4ng
Wrong password
```

How to get help?

You can use GradeChecker (https://learnt.sabanciuniv.edu/GradeChecker/) to check your expected grade. Just a reminder, you will see a character ¶ which refers to a newline in your expected output.

What and where to submit?

You should prepare (or at least test) your program using Python 3.x.x. We will use Python 3.x.x while testing your take-home exam.

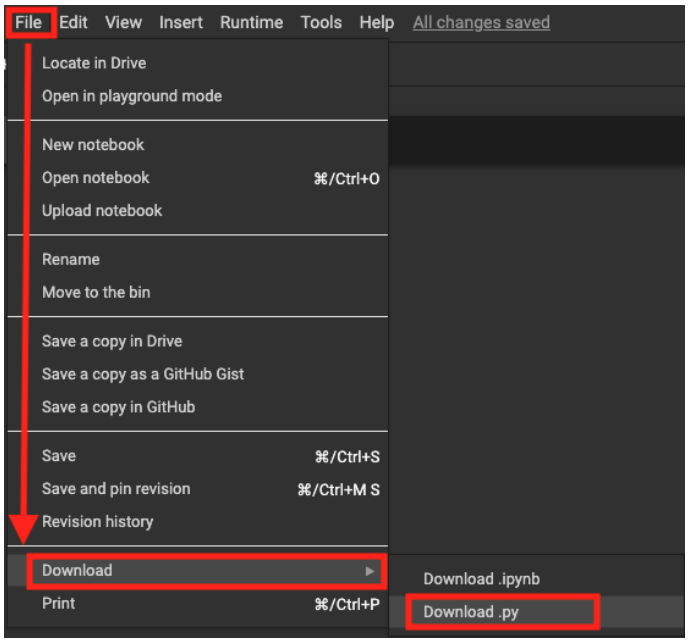
It'd be a good idea to write your name and lastname in the program (as a comment line of course). Do not use any Turkish characters anywhere in your code (not even

in comment parts). If your name and last name is "İnanç Arın", and if you want to write it as comment; then you must type it as follows:

 # Inanc Arin

Submission guidelines are below. Since the grading process will be automatic, students are expected to strictly follow these guidelines. If you do not follow these guidelines, your grade will be 0.

- Download your code as *py* file with "File" ->"*Download*" -> "*Download .py*" as below:



- Name your *py* file that contains your program as follows:

  "username_the2.py"

  For example: if your SUCourse+ username is **"duygukaltop"**, then the name of the *py* file should be: **duygukaltop_the2*.py*** (please only use lowercase letters).

- Please make sure that this file is the latest version of your take-home exam program.

- Submit your work **through SUCourse+ only**! You can use the GradeChecker only to see if your program can produce the correct outputs

both in the correct order and in the correct format. It will not be considered as the official submission. You must submit your work to SUCourse+.

- If you would like to resubmit your work, you should first remove the existing file(s). This step is very important. If you do not delete the old file(s), we will receive both files and the old one may be graded.

General Take-Home Exam Rules

- Successful submission is one of the requirements of the take-home exam. If, for some reason, you cannot successfully submit your take-home exam and we cannot grade it, your grade will be 0.
- There is NO late submission. You need to submit your take-home exam before the deadline. Please be careful that SUCourse+ time and your computer time may have 1-2 minutes differences. You need to take this time difference into consideration.
- Do NOT submit your take-home exam via email or in hardcopy! SUCourse+ is the only way that you can submit your take-home exam.
- If your code does not work because of a syntax error, then we cannot grade it; and thus, your grade will be 0.
- Please do submit your **own** work only. It is really easy to find "similar" programs!
- Plagiarism will not be tolerated. Please check our plagiarism policy given in the syllabus of the course.
- ***You must use Google Colab while developing your solutions. Otherwise, you might be asked to have an interview with the instructors or we might not help when you object.***

Good luck!
Ethem Tunal Hamzaoğlu
& IF100 Instructors