Student Management System

You are asked to write a program that simulates a Student Management System designed for admins only. As a project, it serves two purposes: The program is designed to give you some practice writing programs that manipulate strings, data structures and files, as well as apply sorting and searching algorithms.

When the admin runs the code, the computer first asks the user to input his/her username and password. The program then checks if the username and password are correct. Let's set the username of the admin to be "**admin**" and the password to be "**admin123123**". If the user inputs an incorrect combination, the system should display "Incorrect Username and/or Password". The user is not allowed to enter more than **5 wrong combinations**. Otherwise, the menu of the system will be displayed.

The System

Your job is to write a script that handles the user interaction component of the system. To solve the problem, your program must be able to:

- Read a text file and upload all the students into the dictionary of the system
- Implement the basic control structure and manage the details

The file includes several students in the following format:

SEFST160001: Charbel, Daoud, FSW22001
SEFST160002: Manuella, Germanos, FCS22001
SEFST160003: Anis, Ismail, FCS22002
SEFST160003: …

The first word is the ID of the student. Each student should have an ID and the format of that ID is as follows: SEFSTYYXXXX (i.e: SEFST160001). The YY is the 2-digits reflecting the year in which the student APPLIED to SE Factory. The XXXX is a 4-digits number refelcting the counter on THAT YEAR. We reset that number every year. The second list contains three parts: the first part is the FIRST NAME of the student, the second is the LAST NAME, and the last part is the enrolled course. You are asked to import that file when the program starts. When uploading, pay attention to the duplicate students.

The Flow of The System:
A. The students written inside the file are imported (loaded) into the Dictionary of the program with no intervention and without letting the user know anything.
B. The program starts by greeting the user and asking for their username and password

1. **Display Statistics**
2. **Display All Students**
3. **Add New Student**
4. **Remove Student**
5. **Enroll Existing Student in a Course**
6. **Edit Student**
7. **Display Student**
8. **Exit**

- If the user chooses (1), the system should display some statistics: **Total Number of FCS Students**, **Total Number of FSW Students**, **How many students enrolled in FSW 2022**..
- If the user chooses (2), the program should ask the user if he/she wants to list **all** the students, the students enrolled in **FCS** only, or the students enrolled in **FSW** only. In any case, display the students **sorted by their Last Name**. Display the Last Name, First Name, and the Year they applied to SE Factory (extracted from their IDs). Implement the appropriate sorting algorithm.
- If the user chooses (3), the user shall be able to add a new student to the dictionary. The program prompts for ID, last name, first name. No need to ask for the enrolled course (N/A will be considered in that case). The system shall not accept an already existing student (Same ID).
- If the user chooses (4), the user shall be able to search and remove a specific student from the dictionary using its ID. A message will be displayed if the student is not found.
- If the user chooses (5), the user shall be able to search for any student by ID and then enroll him/her in the mentioned course (User has to write the course ID).
- If the user chooses (6), the user shall be able to search for any student by ID and then update its information inside the Dictionary. A message will be displayed if the student is not found.
- If the user chooses (7), the system prompts for the first name of the student or its ID and the system will display all the students with their information (ID, course, full name) if found. A message will be displayed if no student is found.
- If the user chooses (8), the program should ask him or her if they want to save the changes done in the Dictionary. If yes, the Dictionary should be saved to the same text file found inside the directory of the program. Otherwise, the program will terminate without saving.

After each option, the menu is displayed again. The user can use the system multiple times.