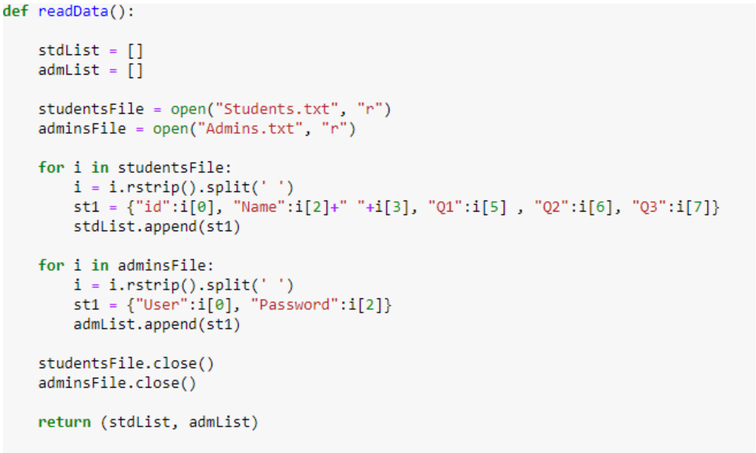
King Fahd University of Petroleum and Minerals
College of Computer Science and Engineering
Information and Computer Science Department

ICS 104-79: Introduction to Programming Using Python and C
Fall Semester 2022-2023
Lab Project 2022

Name:	Mohammed Busaleh	Mohammed Hejazi
ID:	202158210	202182750

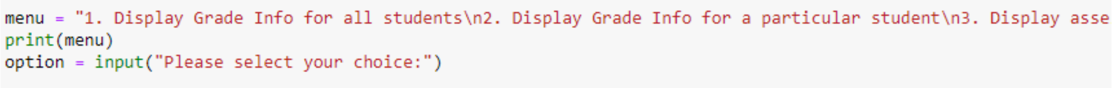
We stored students and admins information as two lists of dictionaries, then used a combination of different python function and operators to apply the requirements.

- Firstly, we read the data using files functions. This code will work in the background



```
def readData():

    stdList = []
    admList = []

    studentsFile = open("Students.txt", "r")
    adminsFile = open("Admins.txt", "r")

    for i in studentsFile:
        i = i.rstrip().split(' ')
        st1 = {"id":i[0], "Name":i[2]+" "+i[3], "Q1":i[5] , "Q2":i[6], "Q3":i[7]}
        stdList.append(st1)

    for i in adminsFile:
        i = i.rstrip().split(' ')
        st1 = {"User":i[0], "Password":i[2]}
        admList.append(st1)

    studentsFile.close()
    adminsFile.close()

    return (stdList, admList)
```

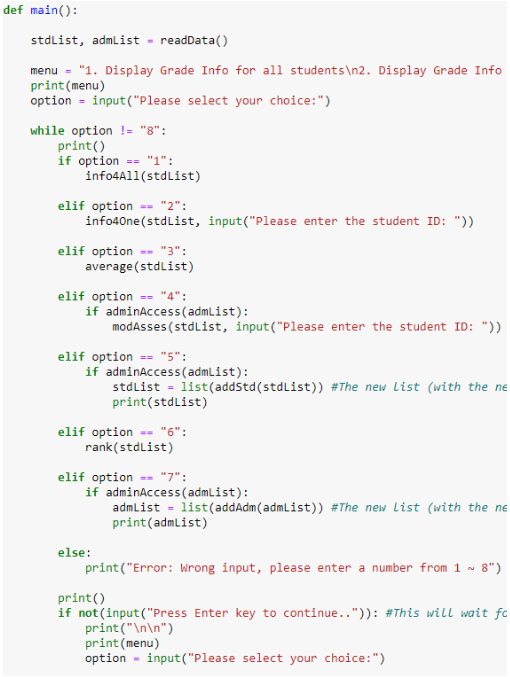
- Then we show the menu and ask the user for the desired option



```
menu = "1. Display Grade Info for all students\n2. Display Grade Info for a particular student\n3. Display asse
print(menu)
option = input("Please select your choice:")
```

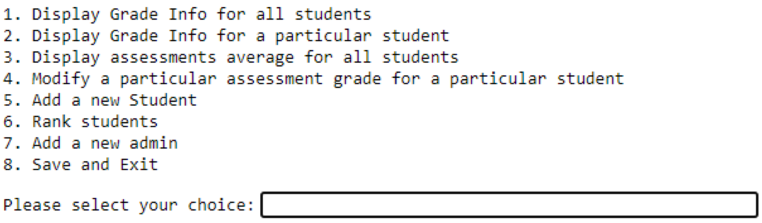
- We use while loop to keep the program running (until option 8 is chosen) and if statements to operate the functions and then wait for the user to print enter

Code



```
def main():

    stdList, admList = readData()

    menu = "1. Display Grade Info for all students\n2. Display Grade Info
    print(menu)
    option = input("Please select your choice:")

    while option != "8":
        print()
        if option == "1":
            info4All(stdList)

        elif option == "2":
            info4One(stdList, input("Please enter the student ID: "))

        elif option == "3":
            average(stdList)

        elif option == "4":
            if adminAccess(admList):
                modAsses(stdList, input("Please enter the student ID: "))

        elif option == "5":
            if adminAccess(admList):
                stdList = list(addStd(stdList)) #The new list (with the ne
                print(stdList)

        elif option == "6":
            rank(stdList)

        elif option == "7":
            if adminAccess(admList):
                admList = list(addAdm(admList)) #The new list (with the ne
                print(admList)

        else:
            print("Error: Wrong input, please enter a number from 1 ~ 8")

        print()
        if not(input("Press Enter key to continue..")): #This will wait fo
            print("\n\n")
            print(menu)
            option = input("Please select your choice:")
```

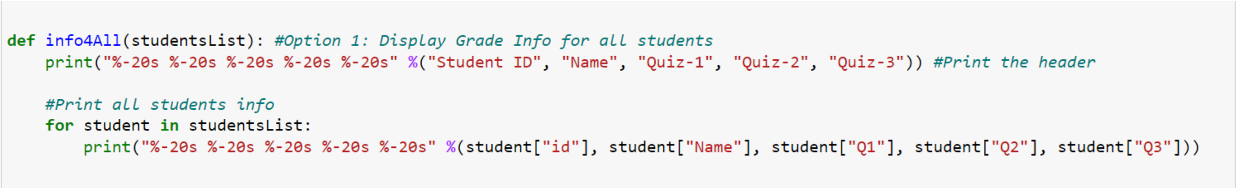
Output



```
1. Display Grade Info for all students
2. Display Grade Info for a particular student
3. Display assessments average for all students
4. Modify a particular assessment grade for a particular student
5. Add a new Student
6. Rank students
7. Add a new admin
8. Save and Exit

Please select your choice:
```

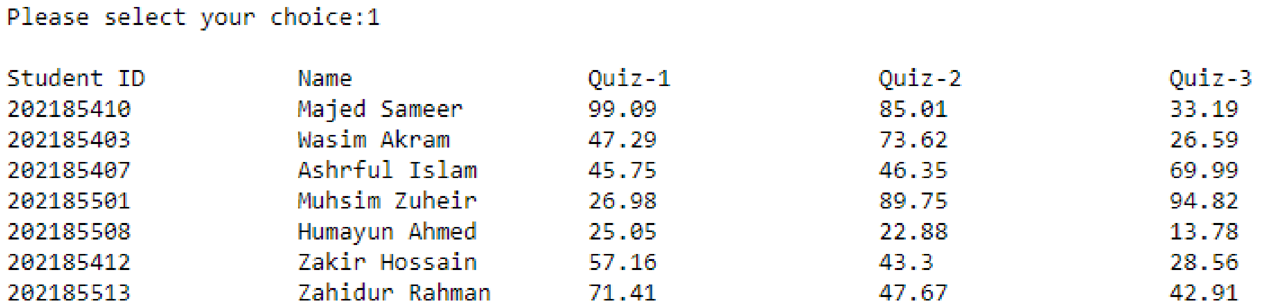
- The display-all-function takes the students list as argument and uses for-loop & print to show all students' data

Code



```
def info4All(studentsList): #Option 1: Display Grade Info for all students
    print("%-20s %-20s %-20s %-20s %-20s" %("Student ID", "Name", "Quiz-1", "Quiz-2", "Quiz-3")) #Print the header

    #Print all students info
    for student in studentsList:
        print("%-20s %-20s %-20s %-20s %-20s" %(student["id"], student["Name"], student["Q1"], student["Q2"], student["Q3"]))
```

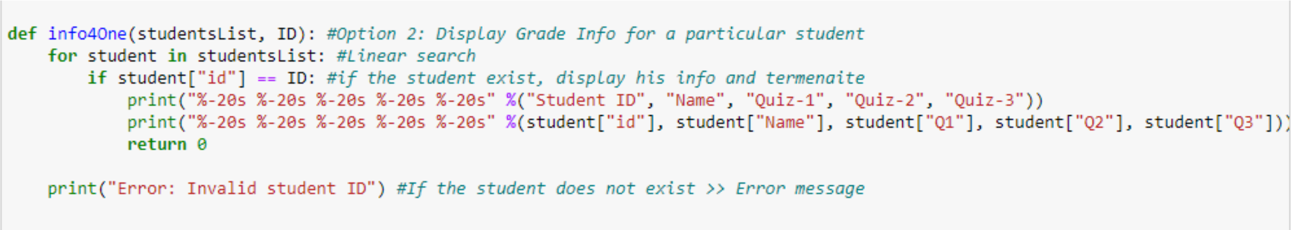
Output



```
Please select your choice:1

Student ID           Name                 Quiz-1               Quiz-2               Quiz-3
202185410            Majed Sameer         99.09                85.01                33.19
202185403            Wasim Akram          47.29                73.62                26.59
202185407            Ashrful Islam        45.75                46.35                69.99
202185501            Muhsim Zuheir        26.98                89.75                94.82
202185508            Humayun Ahmed        25.05                22.88                13.78
202185412            Zakir Hossain        57.16                43.3                 28.56
202185513            Zahidur Rahman       71.41                47.67                42.91
```

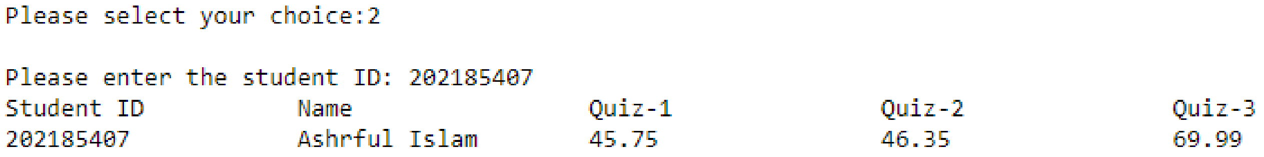
- The display-one-function takes the students list and the ID as argument and uses for-loop to loop through the list. If the student found the function will print the data and terminate, else the function will display error message.

Code



```
def info4One(studentsList, ID): #Option 2: Display Grade Info for a particular student
    for student in studentsList: #Linear search
        if student["id"] == ID: #if the student exist, display his info and termenaite
            print("%-20s %-20s %-20s %-20s %-20s" %("Student ID", "Name", "Quiz-1", "Quiz-2", "Quiz-3"))
            print("%-20s %-20s %-20s %-20s %-20s" %(student["id"], student["Name"], student["Q1"], student["Q2"], student["Q3"]))
            return 0

    print("Error: Invalid student ID") #If the student does not exist >> Error message
```

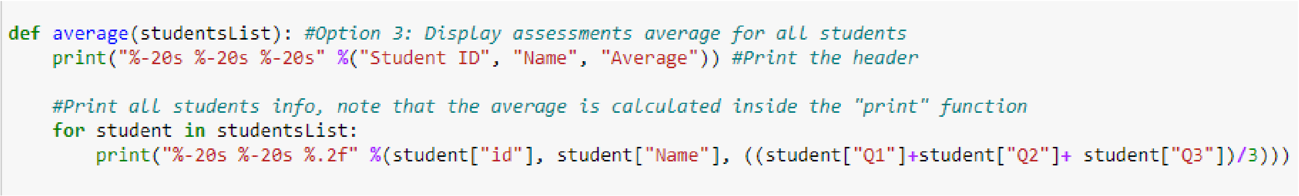
Output



```
Please select your choice:2

Please enter the student ID: 202185407
Student ID           Name                 Quiz-1               Quiz-2               Quiz-3
202185407            Ashrful Islam        45.75                46.35                69.99
```

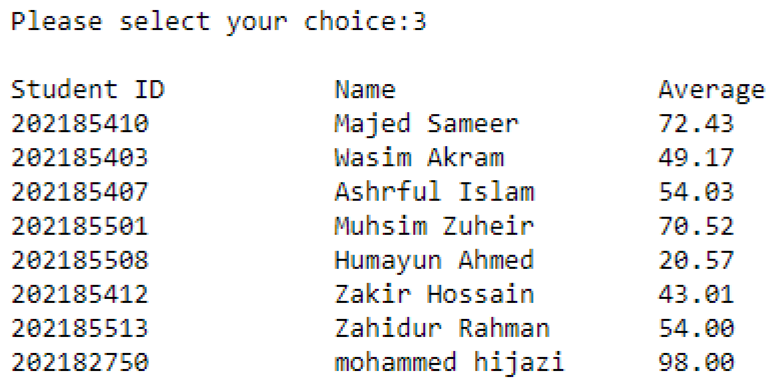
- The average function takes the students list as argument and uses for loop to print their data, note that the average calculation is inside the print function

Code



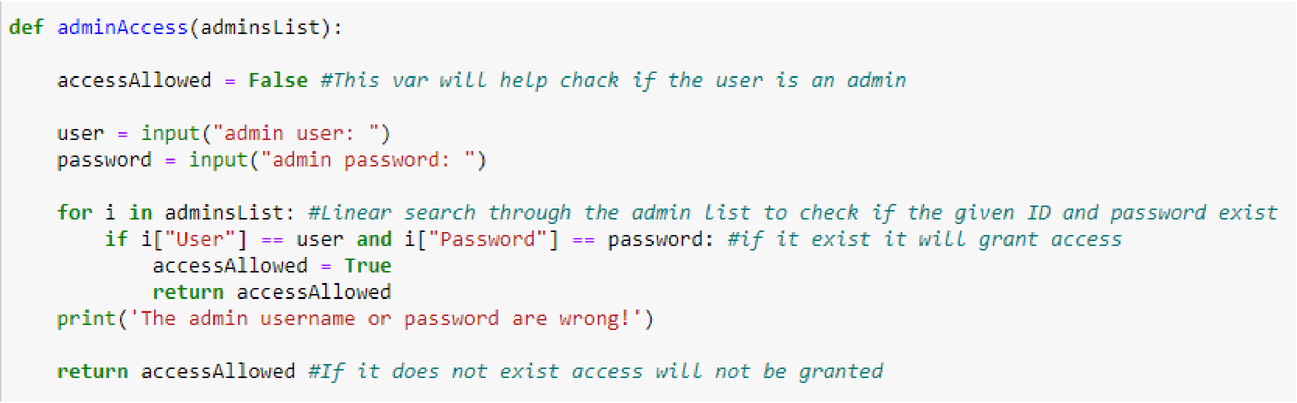
```
def average(studentsList): #Option 3: Display assessments average for all students
    print("%-20s %-20s %-20s" %("Student ID", "Name", "Average")) #Print the header

    #Print all students info, note that the average is calculated inside the "print" function
    for student in studentsList:
        print("%-20s %-20s %.2f" %(student["id"], student["Name"], ((student["Q1"]+student["Q2"]+ student["Q3"])/3)))
```

Output



```
Please select your choice:3

Student ID           Name                 Average
202185410            Majed Sameer         72.43
202185403            Wasim Akram          49.17
202185407            Ashrful Islam        54.03
202185501            Muhsim Zuheir        70.52
202185508            Humayun Ahmed        20.57
202185412            Zakir Hossain        43.01
202185513            Zahidur Rahman       54.00
202182750            mohammed hijazi      98.00
```

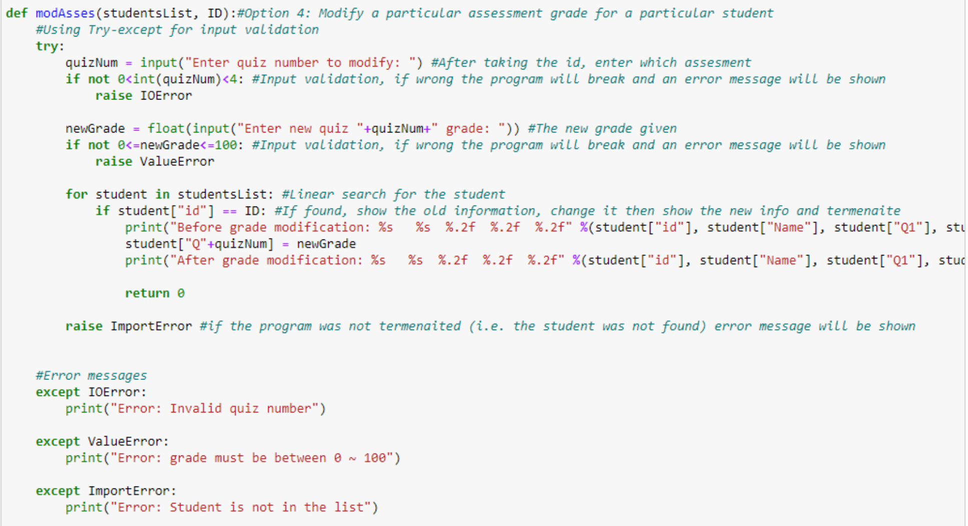
- The admin checking function works as a condition for the if statement that decides weather the user is allowed to use some options, the function takes the admins list as an argue and asks the user for admin ID and password, based on that it will return True or False



```
def adminAccess(adminsList):

    accessAllowed = False #This var will help chack if the user is an admin

    user = input("admin user: ")
    password = input("admin password: ")

    for i in adminsList: #Linear search through the admin list to check if the given ID and password exist
        if i["User"] == user and i["Password"] == password: #if it exist it will grant access
            accessAllowed = True
            return accessAllowed
    print('The admin username or password are wrong!')

    return accessAllowed #If it does not exist access will not be granted
```

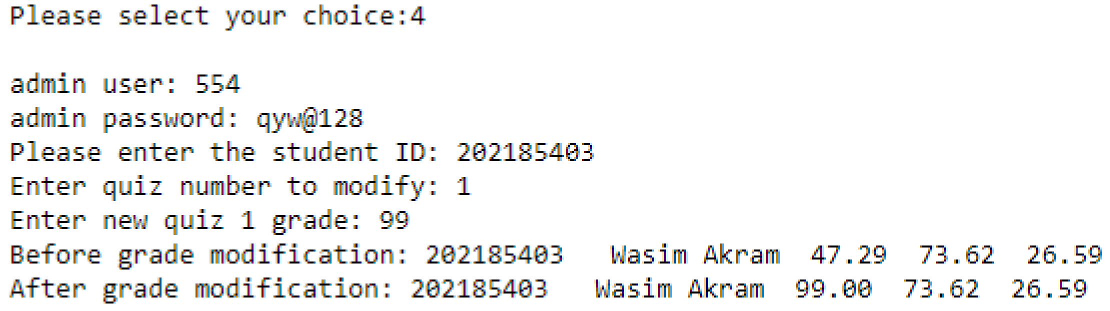
- The modify assess function takes students list and the needed student ID as arguments, it then uses try-except for input validation, the function will rise different error for each wrong input. If all input was correct the function will print the student's old data followed by the updated one (note that the function will ask you to log in as an admin)

Code



```python
def modAsses(studentsList, ID):#Option 4: Modify a particular assessment grade for a particular student
    #Using Try-except for input validation
    try:
        quizNum = input("Enter quiz number to modify: ") #After taking the id, enter which assesment
        if not 0<int(quizNum)<4: #Input validation, if wrong the program will break and an error message will be shown
            raise IOError

        newGrade = float(input("Enter new quiz "+quizNum+" grade: ")) #The new grade given
        if not 0<=newGrade<=100: #Input validation, if wrong the program will break and an error message will be shown
            raise ValueError

        for student in studentsList: #Linear search for the student
            if student["id"] == ID: #If found, show the old information, change it then show the new info and termenaite
                print("Before grade modification: %s   %s  %.2f  %.2f  %.2f" %(student["id"], student["Name"], student["Q1"], stu
                student["Q"+quizNum] = newGrade
                print("After grade modification: %s   %s  %.2f  %.2f  %.2f" %(student["id"], student["Name"], student["Q1"], stud

                return 0

        raise ImportError #if the program was not termenaited (i.e. the student was not found) error message will be shown


    #Error messages
    except IOError:
        print("Error: Invalid quiz number")

    except ValueError:
        print("Error: grade must be between 0 ~ 100")

    except ImportError:
        print("Error: Student is not in the list")
```

Output

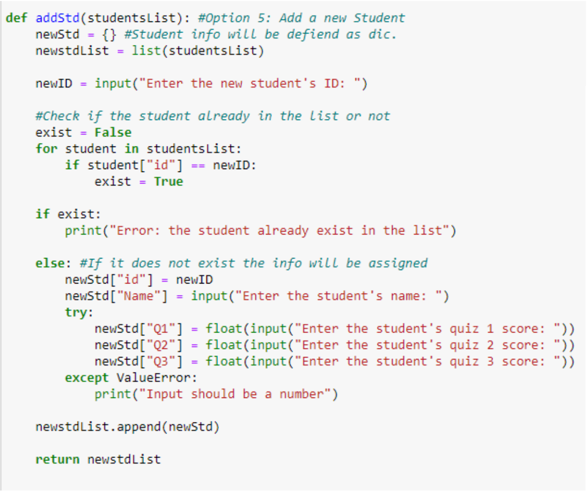


```
Please select your choice:4

admin user: 554
admin password: qyw@128
Please enter the student ID: 202185403
Enter quiz number to modify: 1
Enter new quiz 1 grade: 99
Before grade modification: 202185403   Wasim Akram  47.29  73.62  26.59
After grade modification: 202185403   Wasim Akram  99.00  73.62  26.59
```

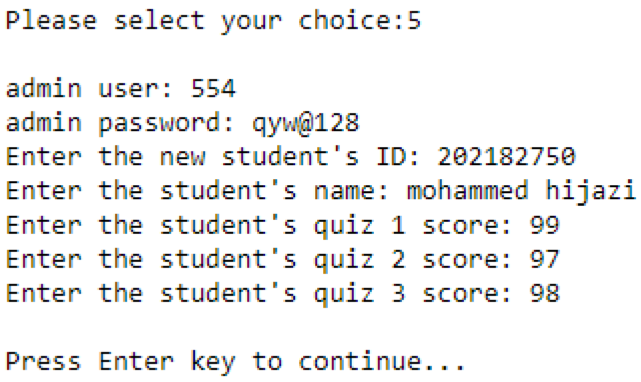
- The add student function takes students list as argument, it then ask the user to enter the new student's ID. If it already exists it will inform the user, else it will continue the register process (note that the function will ask you to log in as an admin)

Code

Output



```python
def addStd(studentsList): #Option 5: Add a new Student
    newStd = {} #Student info will be defiend as dic.
    newstdList = list(studentsList)

    newID = input("Enter the new student's ID: ")

    #Check if the student already in the list or not
    exist = False
    for student in studentsList:
        if student["id"] == newID:
            exist = True

    if exist:
        print("Error: the student already exist in the list")

    else: #If it does not exist the info will be assigned
        newStd["id"] = newID
        newStd["Name"] = input("Enter the student's name: ")
        try:
            newStd["Q1"] = float(input("Enter the student's quiz 1 score: "))
            newStd["Q2"] = float(input("Enter the student's quiz 2 score: "))
            newStd["Q3"] = float(input("Enter the student's quiz 3 score: "))
        except ValueError:
            print("Input should be a number")

    newstdList.append(newStd)

    return newstdList
```

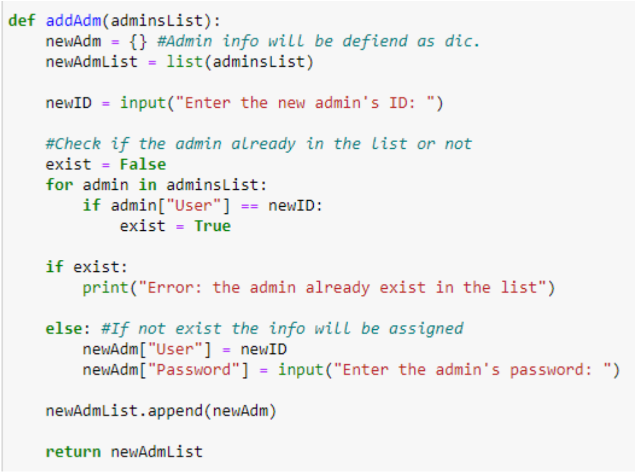


```
Please select your choice:5

admin user: 554
admin password: qyw@128
Enter the new student's ID: 202182750
Enter the student's name: mohammed hijazi
Enter the student's quiz 1 score: 99
Enter the student's quiz 2 score: 97
Enter the student's quiz 3 score: 98

Press Enter key to continue...
```

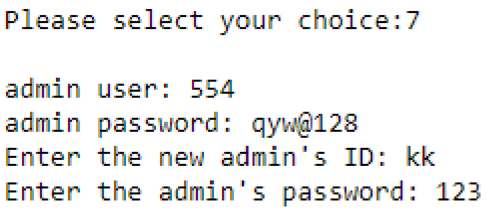
- The add admin function takes admins list as argument, it then asks the user to enter the new admin ID. If it already exists it will inform the user, else it will continue the register process (note that the function will ask you to log in as an admin)

Code

Output

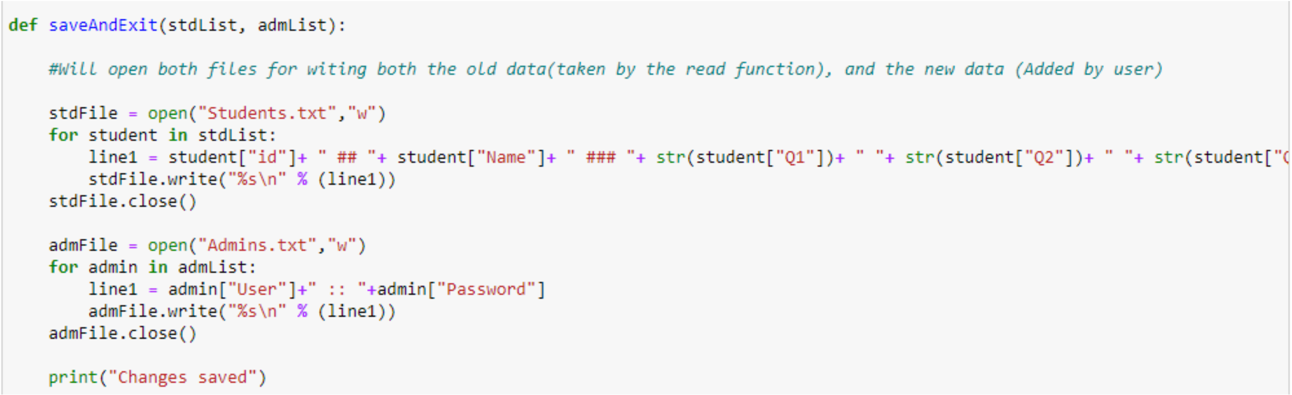


```python
def addAdm(adminsList):
    newAdm = {} #Admin info will be defiend as dic.
    newAdmList = list(adminsList)

    newID = input("Enter the new admin's ID: ")

    #Check if the admin already in the list or not
    exist = False
    for admin in adminsList:
        if admin["User"] == newID:
            exist = True

    if exist:
        print("Error: the admin already exist in the list")

    else: #If not exist the info will be assigned
        newAdm["User"] = newID
        newAdm["Password"] = input("Enter the admin's password: ")

    newAdmList.append(newAdm)

    return newAdmList
```



```
Please select your choice:7

admin user: 554
admin password: qyw@128
Enter the new admin's ID: kk
Enter the admin's password: 123
```

- Finally, using the files functions the changes will be applied and saved to the files that already exist. The new edits will be added to the old data and then all of them will be written in the files. (this happens in the background so no output)



```
def saveAndExit(stdList, admList):

    #Will open both files for witing both the old data(taken by the read function), and the new data (Added by user)

    stdFile = open("Students.txt","w")
    for student in stdList:
        line1 = student["id"]+ " ## "+ student["Name"]+ " ### "+ str(student["Q1"])+ " "+ str(student["Q2"])+ " "+ str(student["
        stdFile.write("%s\n" % (line1))
    stdFile.close()

    admFile = open("Admins.txt","w")
    for admin in admList:
        line1 = admin["User"]+" :: "+admin["Password"]
        admFile.write("%s\n" % (line1))
    admFile.close()

    print("Changes saved")
```