Project Report

1- Main function

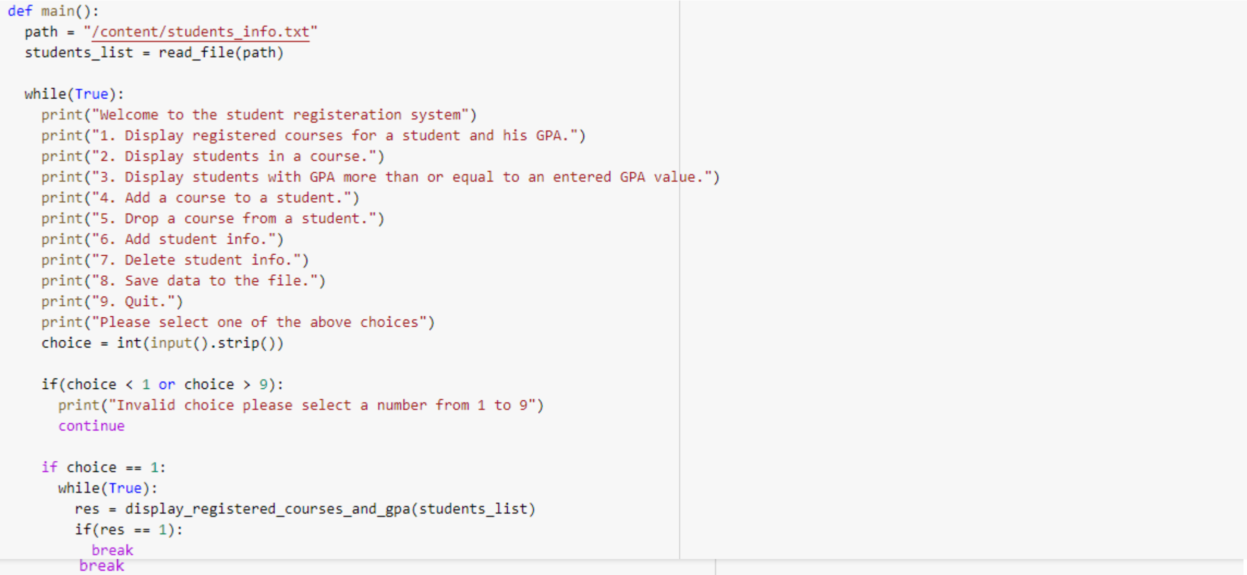
* stores the path of the text file

*Promotes the user to choose one of the options printed before them

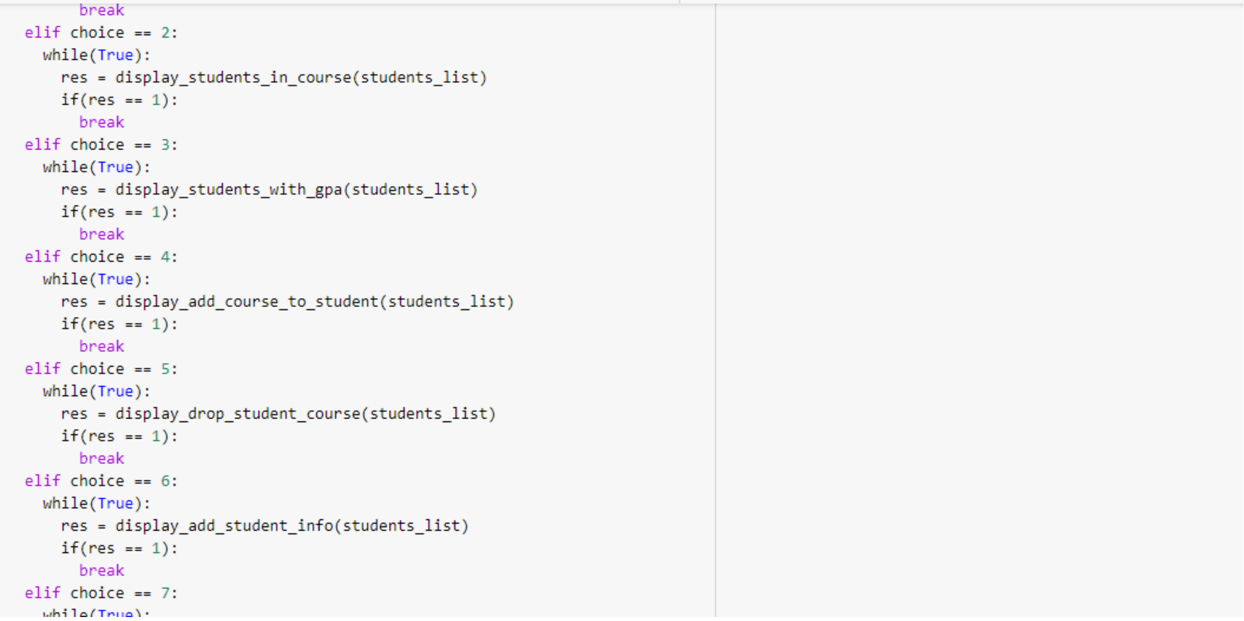
*based on the choice of the user this function directs the program to the proper function

*the function contains while loops to keep the program running

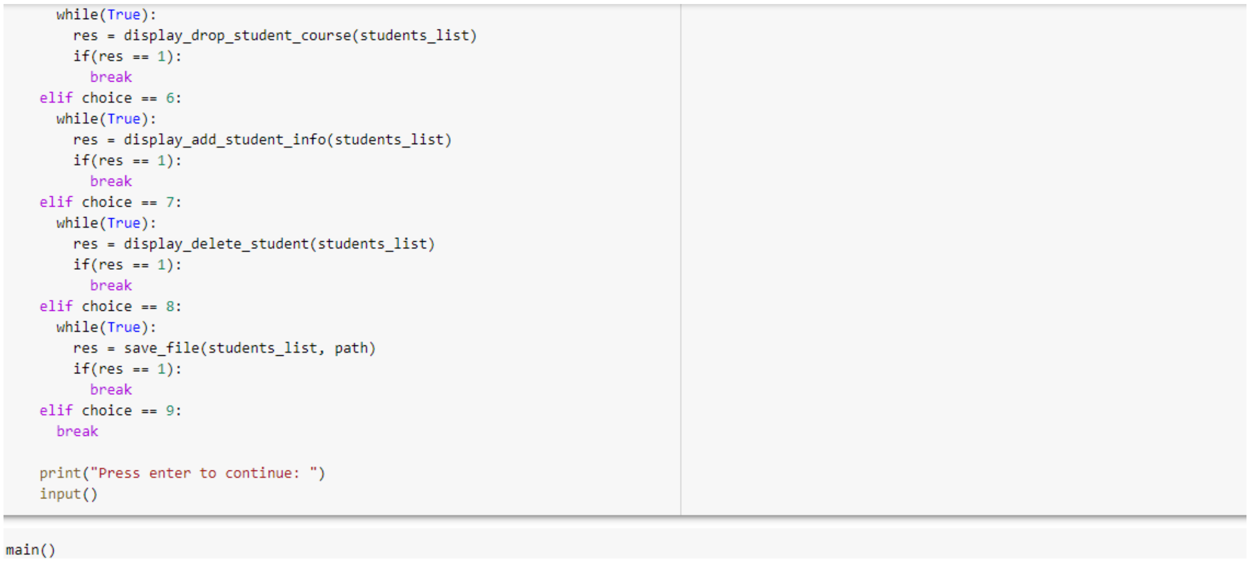
*the function handles the ninth option and it terminates the program if asked to



```
def main():
  path = "/content/students_info.txt"
  students_list = read_file(path)

  while(True):
    print("Welcome to the student registeration system")
    print("1. Display registered courses for a student and his GPA.")
    print("2. Display students in a course.")
    print("3. Display students with GPA more than or equal to an entered GPA value.")
    print("4. Add a course to a student.")
    print("5. Drop a course from a student.")
    print("6. Add student info.")
    print("7. Delete student info.")
    print("8. Save data to the file.")
    print("9. Quit.")
    print("Please select one of the above choices")
    choice = int(input().strip())

    if(choice < 1 or choice > 9):
      print("Invalid choice please select a number from 1 to 9")
      continue

    if choice == 1:
      while(True):
        res = display_registered_courses_and_gpa(students_list)
        if(res == 1):
          break
```



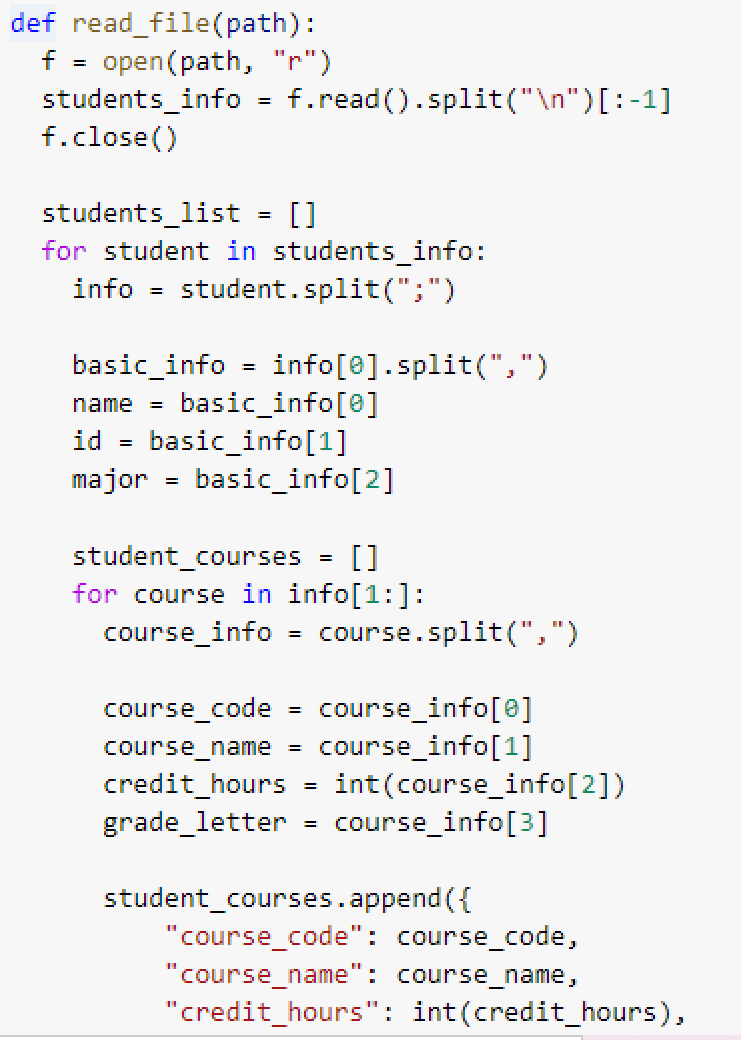
```
          break
    elif choice == 2:
      while(True):
        res = display_students_in_course(students_list)
        if(res == 1):
          break
    elif choice == 3:
      while(True):
        res = display_students_with_gpa(students_list)
        if(res == 1):
          break
    elif choice == 4:
      while(True):
        res = display_add_course_to_student(students_list)
        if(res == 1):
          break
    elif choice == 5:
      while(True):
        res = display_drop_student_course(students_list)
        if(res == 1):
          break
    elif choice == 6:
      while(True):
        res = display_add_student_info(students_list)
        if(res == 1):
          break
    elif choice == 7:
      while(True):
```



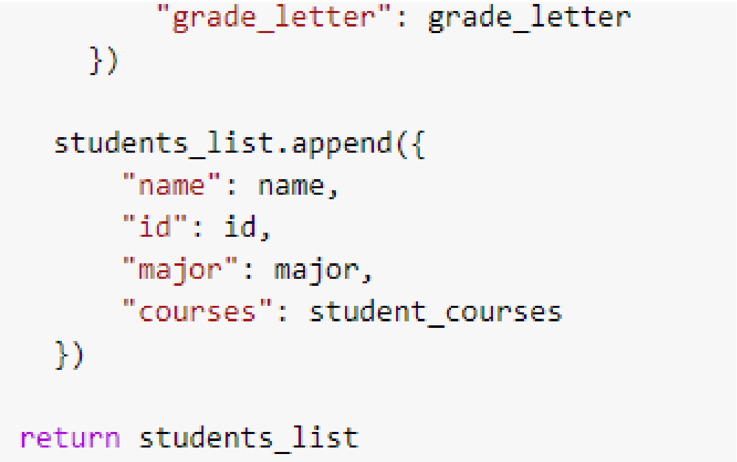
```
      while(True):
        res = display_drop_student_course(students_list)
        if(res == 1):
          break
    elif choice == 6:
      while(True):
        res = display_add_student_info(students_list)
        if(res == 1):
          break
    elif choice == 7:
      while(True):
        res = display_delete_student(students_list)
        if(res == 1):
          break
    elif choice == 8:
      while(True):
        res = save_file(students_list, path)
        if(res == 1):
          break
    elif choice == 9:
      break

    print("Press enter to continue: ")
    input()

main()
```

2- Reading the file

*this function opens the text file and then reads the file
*the function breaks the information of the text file and stores it in the proper variables.



```
def read_file(path):
  f = open(path, "r")
  students_info = f.read().split("\n")[:-1]
  f.close()

  students_list = []
  for student in students_info:
    info = student.split(";")

    basic_info = info[0].split(",")
    name = basic_info[0]
    id = basic_info[1]
    major = basic_info[2]

    student_courses = []
    for course in info[1:]:
      course_info = course.split(",")

      course_code = course_info[0]
      course_name = course_info[1]
      credit_hours = int(course_info[2])
      grade_letter = course_info[3]

      student_courses.append({
          "course_code": course_code,
          "course_name": course_name,
          "credit_hours": int(credit_hours),
```

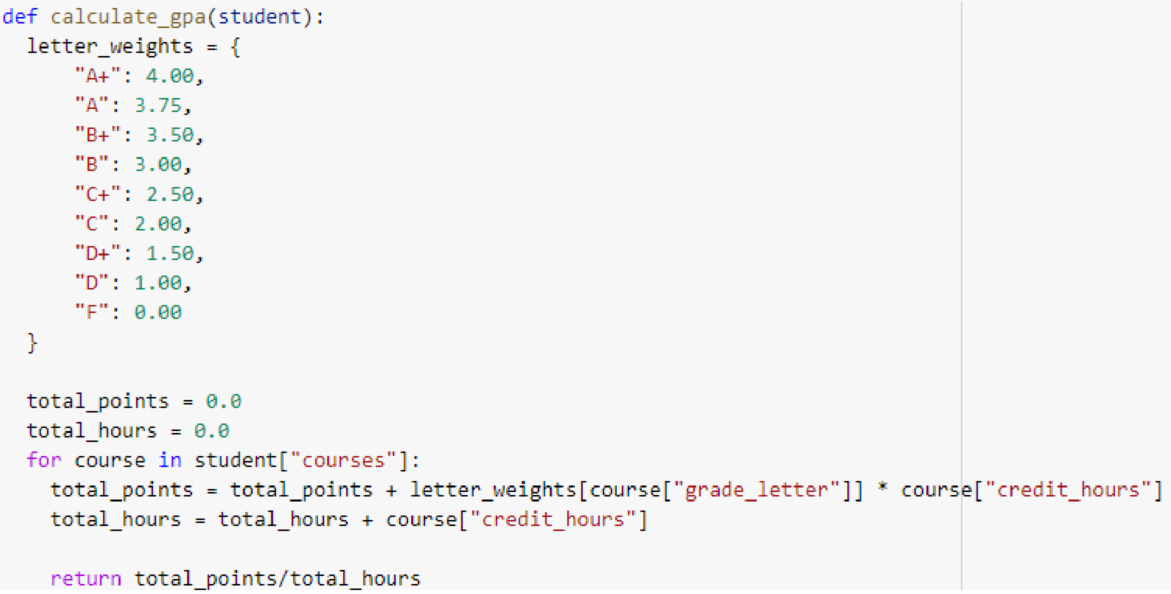


```
        "grade_letter": grade_letter
    })

  students_list.append({
      "name": name,
      "id": id,
      "major": major,
      "courses": student_courses
  })

return students_list
```

3- Calculating the GPA

*this function stores the weight of every grade

*this function performs the mathematical operations required to find the GPA

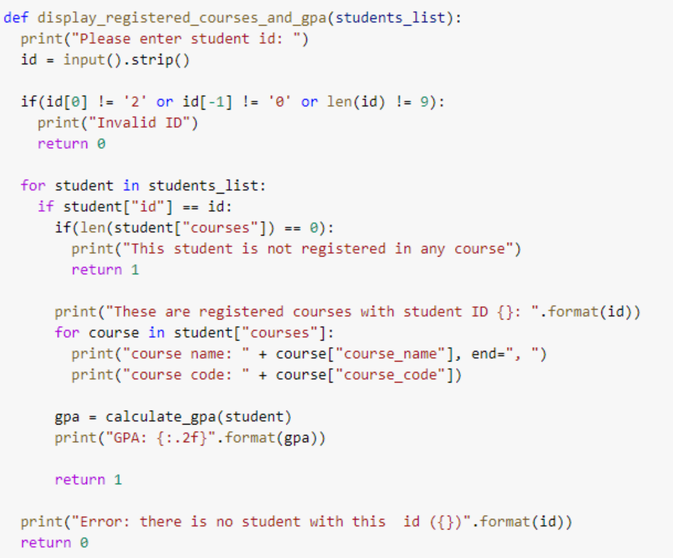


```
def calculate_gpa(student):
  letter_weights = {
      "A+": 4.00,
      "A": 3.75,
      "B+": 3.50,
      "B": 3.00,
      "C+": 2.50,
      "C": 2.00,
      "D+": 1.50,
      "D": 1.00,
      "F": 0.00
  }

  total_points = 0.0
  total_hours = 0.0
  for course in student["courses"]:
    total_points = total_points + letter_weights[course["grade_letter"]] * course["credit_hours"]
    total_hours = total_hours + course["credit_hours"]

    return total_points/total_hours
```

4- Displaying registered courses and GPA

*this function promotes the user to enter an ID, and then checks if the ID is valid

*this function finds the student with the entered ID, and then prints the student's registered courses and his GPA using the previous function and the student's information.

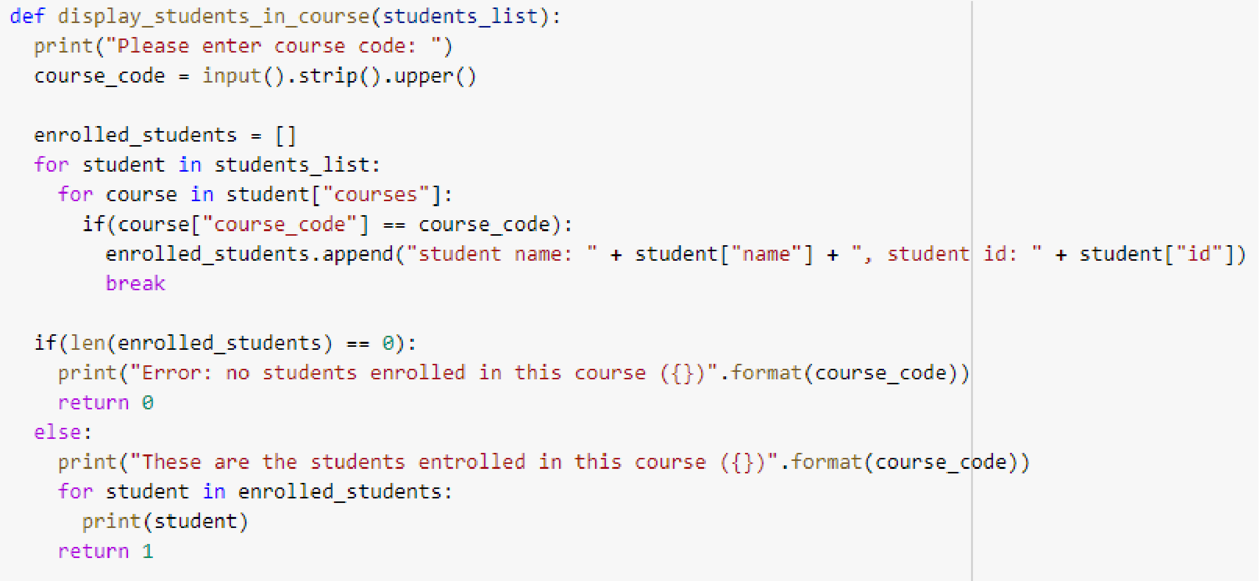


```
def display_registered_courses_and_gpa(students_list):
  print("Please enter student id: ")
  id = input().strip()

  if(id[0] != '2' or id[-1] != '0' or len(id) != 9):
    print("Invalid ID")
    return 0

  for student in students_list:
    if student["id"] == id:
      if(len(student["courses"]) == 0):
        print("This student is not registered in any course")
        return 1

      print("These are registered courses with student ID {}: ".format(id))
      for course in student["courses"]:
        print("course name: " + course["course_name"], end=", ")
        print("course code: " + course["course_code"])

      gpa = calculate_gpa(student)
      print("GPA: {:.2f}".format(gpa))

      return 1

  print("Error: there is no student with this  id ({})".format(id))
  return 0
```

5- Displaying students in a course

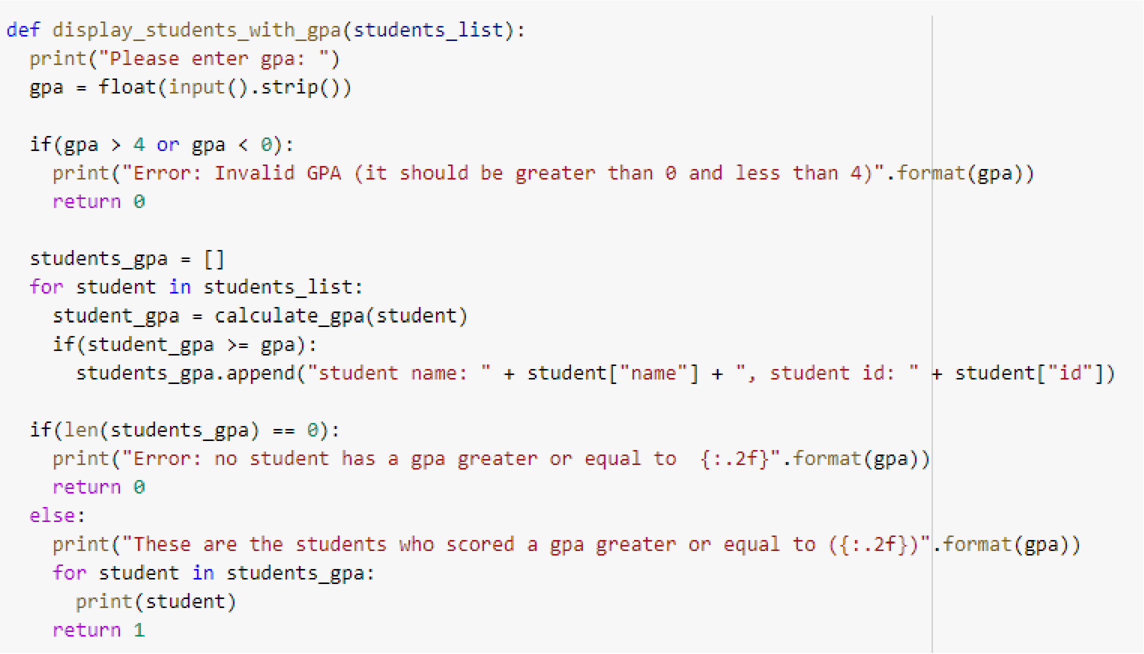
*this function is used to display students in a course by promoting the user to enter a course code, and then checking all the students to see who is enrolled in the course

*if no students is enrolled it prints a proper error massage



```
def display_students_in_course(students_list):
  print("Please enter course code: ")
  course_code = input().strip().upper()

  enrolled_students = []
  for student in students_list:
    for course in student["courses"]:
      if(course["course_code"] == course_code):
        enrolled_students.append("student name: " + student["name"] + ", student id: " + student["id"])
        break

  if(len(enrolled_students) == 0):
    print("Error: no students enrolled in this course ({})".format(course_code))
    return 0
  else:
    print("These are the students entrolled in this course ({})".format(course_code))
    for student in enrolled_students:
      print(student)
    return 1
```

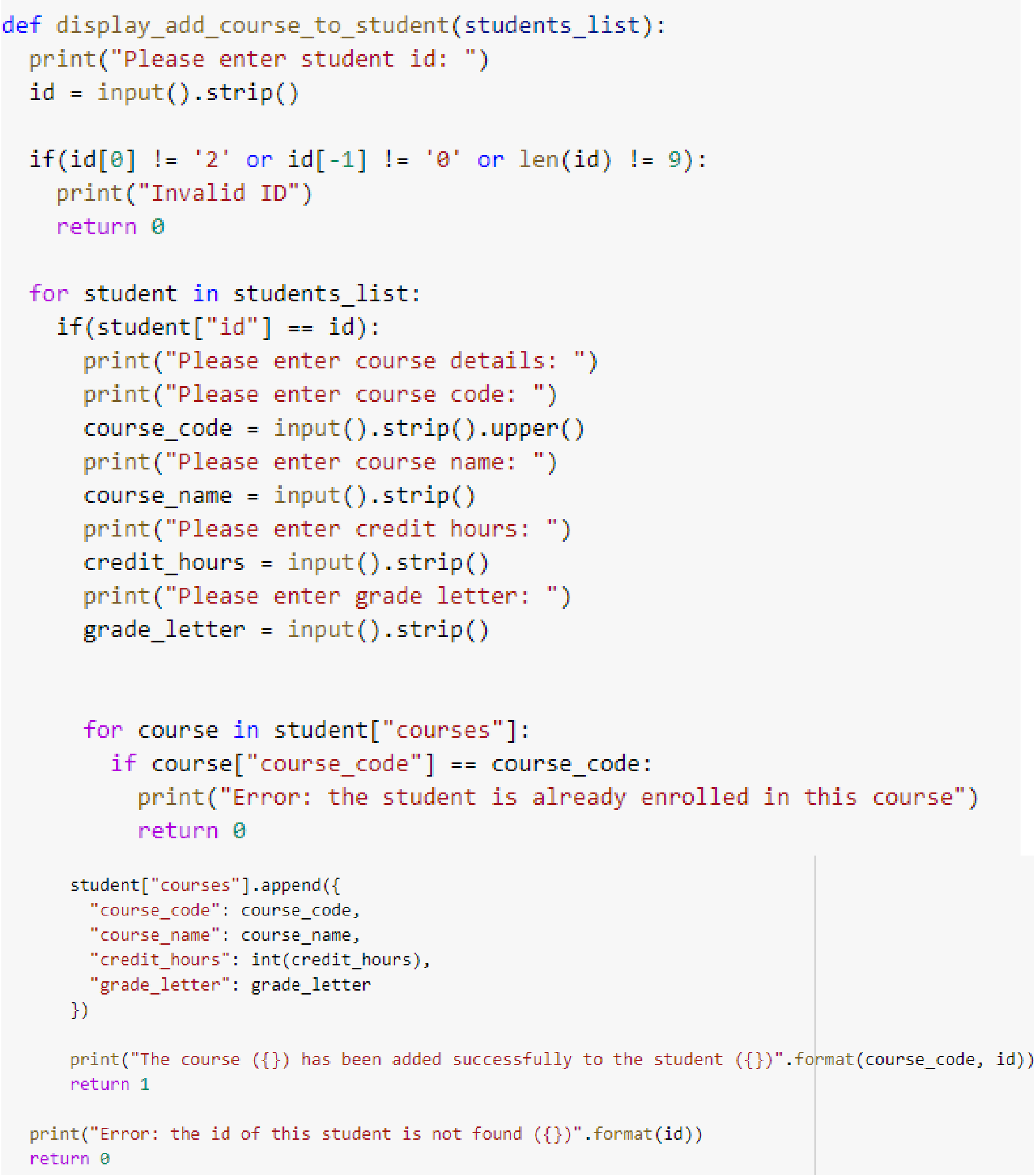
6- Displaying students with GPA or higher
*this function promotes the user to enter a GPA
*this function finds students with GPA higher or equal to the entered GPA, by comparing it to the students data



```
def display_students_with_gpa(students_list):
  print("Please enter gpa: ")
  gpa = float(input().strip())

  if(gpa > 4 or gpa < 0):
    print("Error: Invalid GPA (it should be greater than 0 and less than 4)".format(gpa))
    return 0

  students_gpa = []
  for student in students_list:
    student_gpa = calculate_gpa(student)
    if(student_gpa >= gpa):
      students_gpa.append("student name: " + student["name"] + ", student id: " + student["id"])

  if(len(students_gpa) == 0):
    print("Error: no student has a gpa greater or equal to  {:.2f}".format(gpa))
    return 0
  else:
    print("These are the students who scored a gpa greater or equal to ({:.2f})".format(gpa))
    for student in students_gpa:
      print(student)
    return 1
```

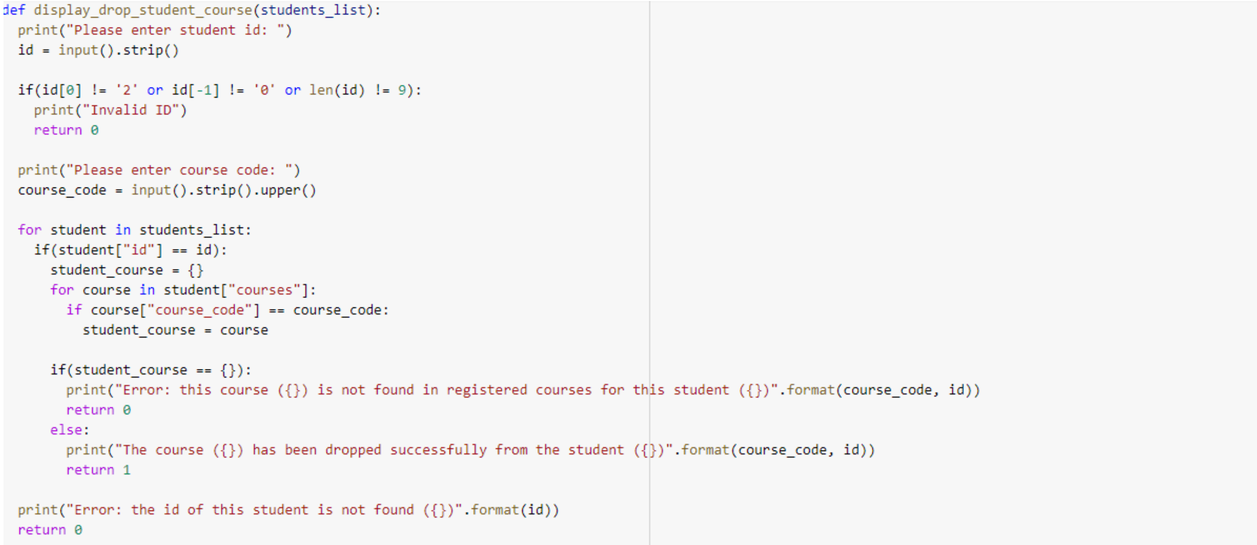
7- Adding a course to a student

- This function adds a course to a student by asking for the student ID and verifying it and then asking for the course information
- This function also checks if the student is already enrolled in the course



```
def display_add_course_to_student(students_list):
  print("Please enter student id: ")
  id = input().strip()

  if(id[0] != '2' or id[-1] != '0' or len(id) != 9):
    print("Invalid ID")
    return 0

  for student in students_list:
    if(student["id"] == id):
      print("Please enter course details: ")
      print("Please enter course code: ")
      course_code = input().strip().upper()
      print("Please enter course name: ")
      course_name = input().strip()
      print("Please enter credit hours: ")
      credit_hours = input().strip()
      print("Please enter grade letter: ")
      grade_letter = input().strip()


      for course in student["courses"]:
        if course["course_code"] == course_code:
          print("Error: the student is already enrolled in this course")
          return 0

      student["courses"].append({
        "course_code": course_code,
        "course_name": course_name,
        "credit_hours": int(credit_hours),
        "grade_letter": grade_letter
      })

      print("The course ({}) has been added successfully to the student ({})".format(course_code, id))
      return 1

  print("Error: the id of this student is not found ({})".format(id))
  return 0
```

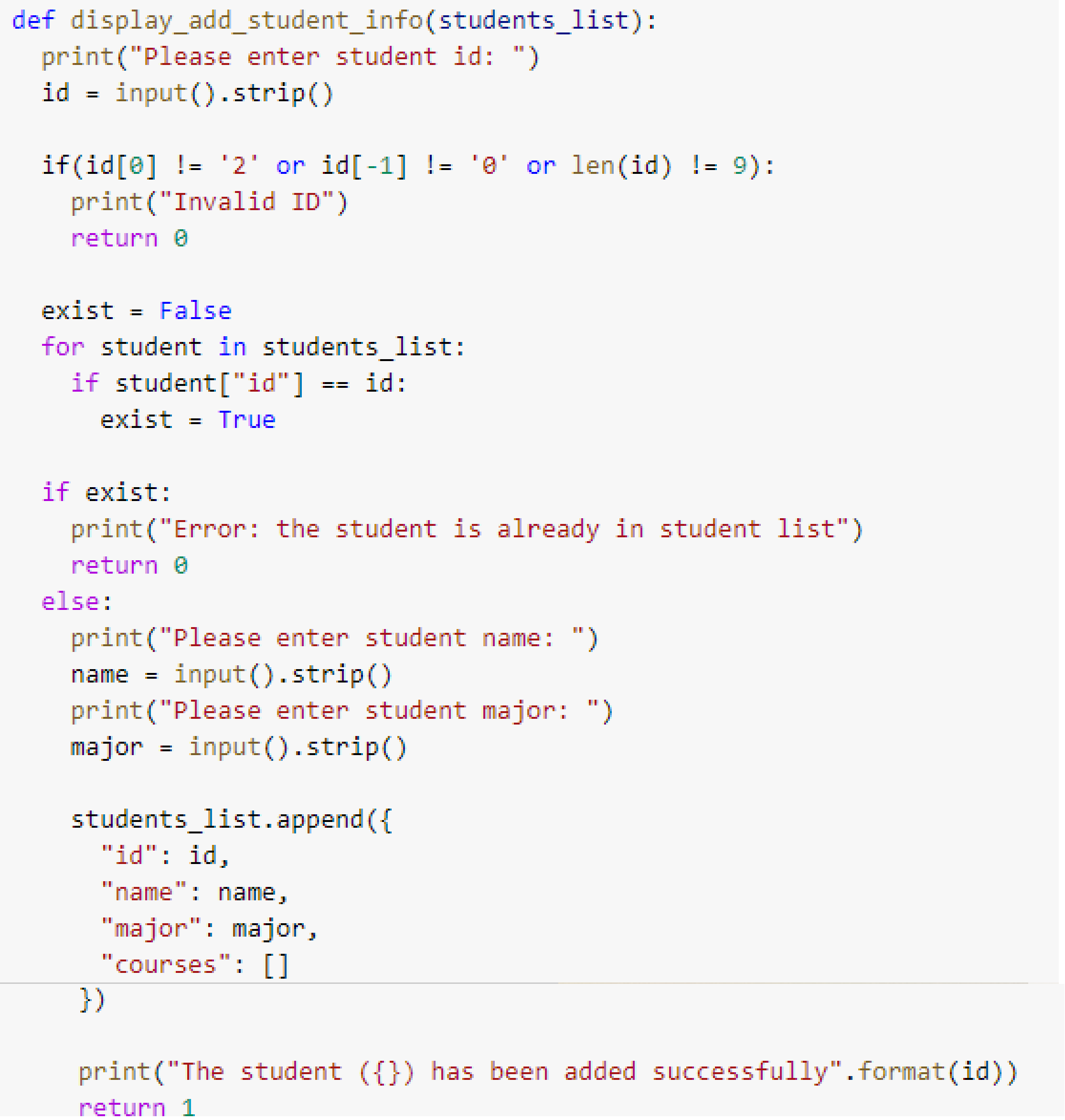
8- Dropping a course from a student
*this function drops a course from a student by asking the user to enter an ID and then verifying this ID and then asking for the course code and looking for it in the data and then dropping it
* if the course is not registered an error massage will appear



```python
def display_drop_student_course(students_list):
  print("Please enter student id: ")
  id = input().strip()

  if(id[0] != '2' or id[-1] != '0' or len(id) != 9):
    print("Invalid ID")
    return 0

  print("Please enter course code: ")
  course_code = input().strip().upper()

  for student in students_list:
    if(student["id"] == id):
      student_course = {}
      for course in student["courses"]:
        if course["course_code"] == course_code:
          student_course = course

      if(student_course == {}):
        print("Error: this course ({}) is not found in registered courses for this student ({})".format(course_code, id))
        return 0
      else:
        print("The course ({}) has been dropped successfully from the student ({})".format(course_code, id))
        return 1

  print("Error: the id of this student is not found ({})".format(id))
  return 0
```

9- Adding student’s information

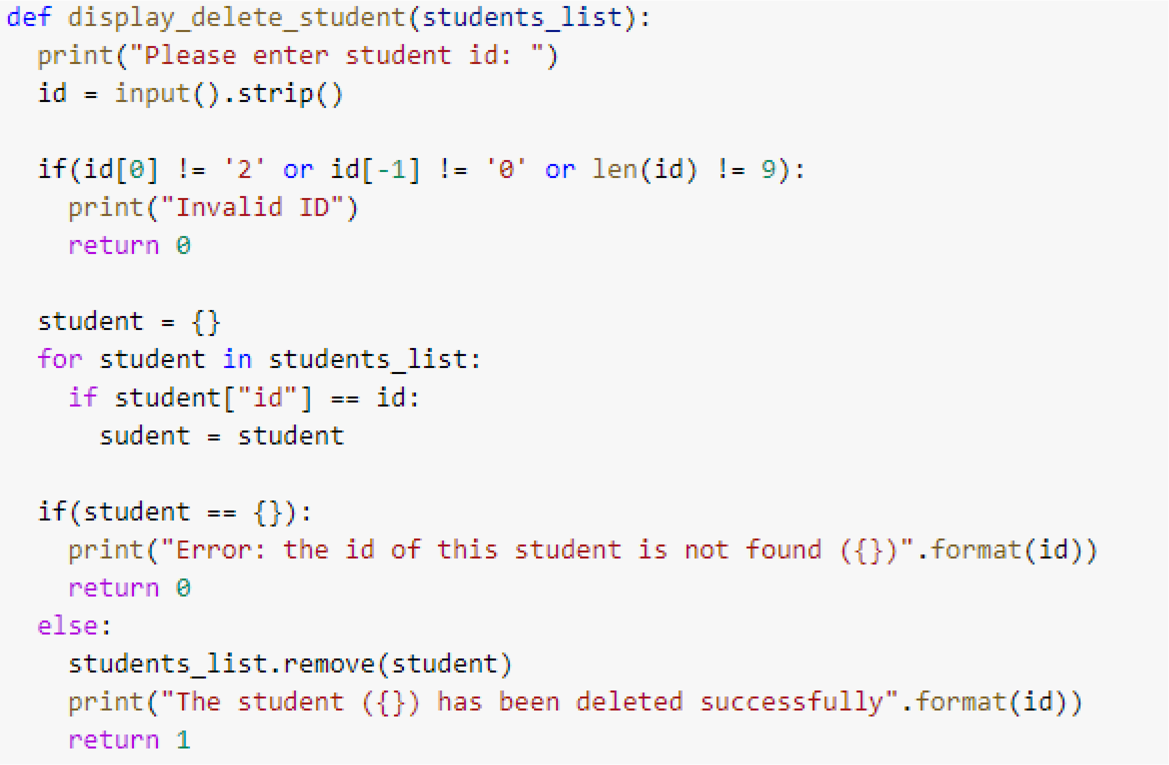
*this function adds a student to the file by asking the user to enter an ID and then verifying this ID and checking if the student doesn’t exist and then asking the user to enter the new student’s information.



```python
def display_add_student_info(students_list):
  print("Please enter student id: ")
  id = input().strip()

  if(id[0] != '2' or id[-1] != '0' or len(id) != 9):
    print("Invalid ID")
    return 0

  exist = False
  for student in students_list:
    if student["id"] == id:
      exist = True

  if exist:
    print("Error: the student is already in student list")
    return 0
  else:
    print("Please enter student name: ")
    name = input().strip()
    print("Please enter student major: ")
    major = input().strip()

    students_list.append({
      "id": id,
      "name": name,
      "major": major,
      "courses": []
    })

    print("The student ({}) has been added successfully".format(id))
    return 1
```

10- Deleting student’s information

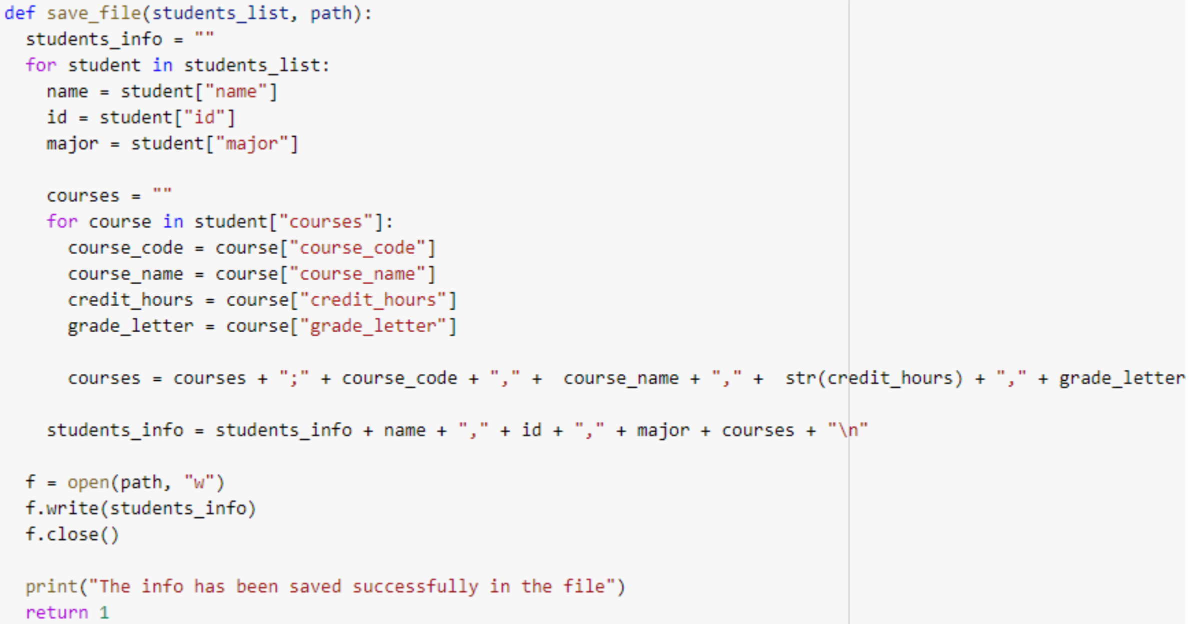
*this function deletes a students information from the file by file by asking the user to enter an ID and then verifying this ID and checking if the student doesn’t exist. If he does the functions deletes him\her.



```
def display_delete_student(students_list):
  print("Please enter student id: ")
  id = input().strip()

  if(id[0] != '2' or id[-1] != '0' or len(id) != 9):
    print("Invalid ID")
    return 0

  student = {}
  for student in students_list:
    if student["id"] == id:
      sudent = student

  if(student == {}):
    print("Error: the id of this student is not found ({})".format(id))
    return 0
  else:
    students_list.remove(student)
    print("The student ({}) has been deleted successfully".format(id))
    return 1
```

11- Saving the information to the file

*this function saves the changes that are done by the program to the text file



```
def save_file(students_list, path):
  students_info = ""
  for student in students_list:
    name = student["name"]
    id = student["id"]
    major = student["major"]

    courses = ""
    for course in student["courses"]:
      course_code = course["course_code"]
      course_name = course["course_name"]
      credit_hours = course["credit_hours"]
      grade_letter = course["grade_letter"]

      courses = courses + ";" + course_code + "," +  course_name + "," +  str(credit_hours) + "," + grade_letter

    students_info = students_info + name + "," + id + "," + major + courses + "\n"

  f = open(path, "w")
  f.write(students_info)
  f.close()

  print("The info has been saved successfully in the file")
  return 1
```