BSc (Hons) Artificial intelligence and Data Science

Module name	-	Programming Fundamentals
Module number	-	CM1601
Module coordinator	-	Mr.Iresh Bandara
Semester	-	1
Year	-	1
Assignment type	-	Individual coursework
Assignment deadline	-	5th December 2022
IIT Student ID	-	20211344
RGU Student ID	-	2237948
Student Name	-	Pallegama Mudiyanselage Seth Nimthaka Rajarathne

Executive summary

This program is designed to manage a World Rallycross Championship using Python with a detailed report. This report includes flowcharts and test cases for ADD and DDD, a task introduction with code, conclusions, hypotheses, and references. Tasks are fully described in terms of how they work and what happens at the end of each task.

Table of Contents

Table of Figures

1. Flow charts

1.1 Flow chart for the add function

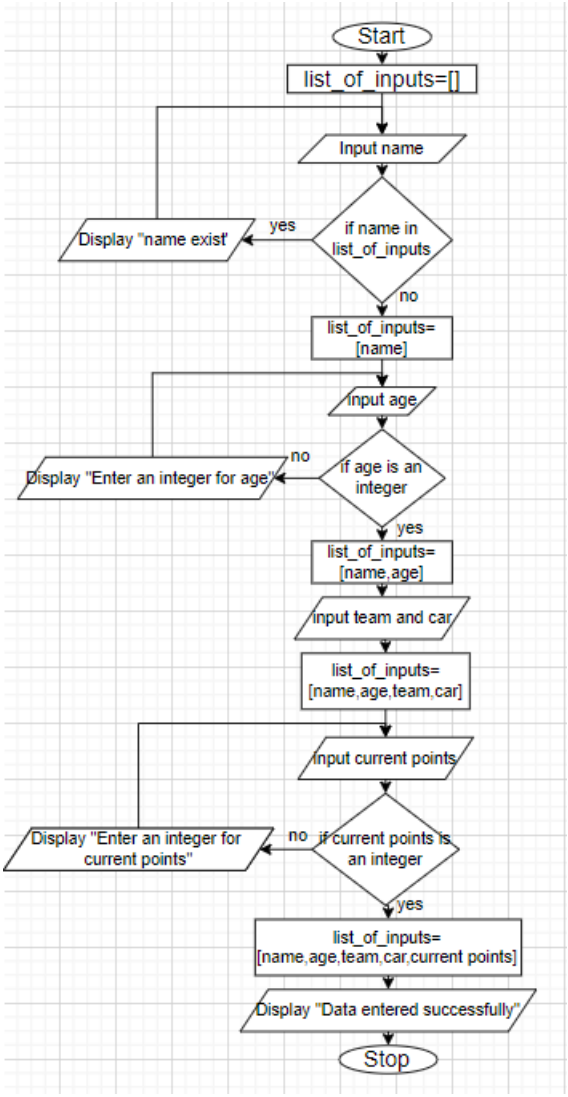

Figure 1 ADD flow chart

1.2 Flow chart for the delete function

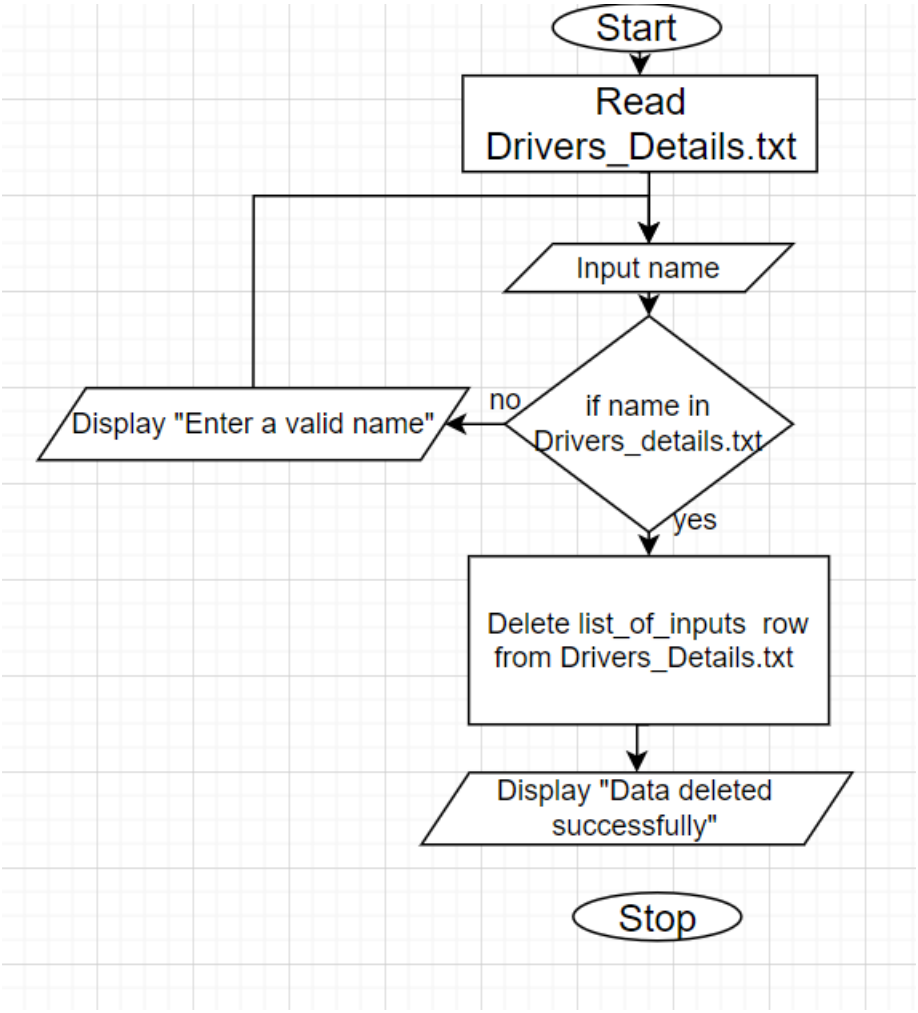

Figure 2 DDD flow chart

2. Introduction to functions with code

i. Main Program

```
import options
import validation

# Main program
while True:
    options.main_menu()
    opt = input("\n\tType the option you want : ")
    opt = opt.upper()
    if opt == 'ADD':
        options.add()
    elif opt == 'DDD':
        options.ddd()
    elif opt == 'UDD':
        options.udd()
    elif opt == 'VCT':
        options.vct()
    elif opt == 'SRR':
        options.srr()
    elif opt == 'VRL':
        options.vrl()
    elif opt == 'STF':
        options.stf()
    elif opt == 'RFF':
        options.rff()
    elif opt == 'ESC':
        options.esc()
        break
    else:
        print("\nInvalid option, Enter a valid option again")
        continue
```

ii. Sub Program 1

```
import os
import validation

#Display main menu
def main_menu():
    print("""\n\tType ADD for adding driver details
    \tType DDD for deleting
    \tType UDD for updating driver details
    \tType VCT for viewing the rally cross standings table
    \tType SRR for simulating a random race
    \tType VRL for viewing race table sorted according to the date
    \tType STF to save the current data to a text file
    \tType RFF to load data from the saved text file
    \tType ESC to exit the program """)

#Function1 ADD
def add():
```

```
    validation.name()
    validation.age()
    validation.team()
    validation.car()
    validation.points()
    f= open('Drivers_Details.txt','a')
    f.write(f"{validation.list_of_inputs}\n")
    f.close()
    del validation.list_of_inputs[:5]
    print("\n\tData entered successfully")


#Function2 DDD
def ddd():
    validation.delete_by_name()

#Function3 UDD
def udd():
        ddd()
        print("Enter details again")
        add()

#Function4 VCT
def vct():
    validation.sorting()
    print("\n")
    print("  'NAME',   'AGE',   'TEAM',   'CAR',   'POINTS'")
    with open("Drivers_Details.txt") as r:
        print(r.read())

#Function5 SRR
def srr():
   validation.random_rase_genarator()


#Function6 VRL
def vrl():
    with open("race_table.txt") as r:
        print(r.read())

#Function7 STF
def stf():
    print("Data Saved into text file called 'Drivers_Details.txt'")

#Function8 RFF
def rff():
    print("\n")
    print("  'NAME',   'AGE',   'TEAM',   'CAR',   'POINTS'")
    with open("Drivers_Details.txt") as r:
        print (r.read())

#Function9 ESC
def esc():
    print ("Programm Terminated")
```

iii. Sub Program 2

```
import options
import random


list_of_inputs=list()
x=0

#Validate the name
def name():
    file = open("Drivers_Details.txt","r")
    name = []
    for x in file:
        x = x.replace("[", "").replace("]", "").replace("\"",
"").replace("\'", "").replace(" ","").replace("\n", "").strip().split(",")
        name.append(x[0])
    file.close()
    while True:
        try:
            d_name = input("\n\t\tEnter the driver's name\t\t\t: ")
            if d_name not in name:
                d_name.lower()
                list_of_inputs.append(d_name)
                break
            else:
                print("name exist")
                continue
        except:
            break

#Validate the age
def age():
    try:
        d_age =int(input("\t\tEnter driver's age\t\t\t: "))
        list_of_inputs.append(d_age)
    except ValueError:
        print(("\t\tEnter an integer for age"))
        age()
    return

#Team for driver
def team():
    try:
        d_team = input("\t\tEnter the driver's team\t\t\t: ")
        list_of_inputs.append(d_team)
    except  ValueError:
        print(("\t\tEnter string for team"))
        team()
    return

#Input details to car
def car():
    try:
        d_car = input("\t\tEnter driver's car \t\t\t: ")
        list_of_inputs.append(d_car)
    except  :
```

```
            car()
        return

    #Input points
    def points():
        try:
            d_point = int(input("\t\tEnter the driver's current points\t: "))
            list_of_inputs.append(d_point)

            global x
            x+=1
        except  ValueError:
            print(("\t\tEnter integer for current points"))
            points()
        return


    #search and deleting part
    def delete_by_name():
        try:
            rd=0
            rdt=0
            name = str(input("\n\t\tEnter the driver's name to delete \t\t: "))
            name=name.lower()
            with open("Drivers_Details.txt", "r") as file_input:
                with open("newfile.txt", "w") as output:
                    for line in file_input:
                        rd+=1
                        data = line.replace("[", "").replace("]",
    "").replace("\"", "").replace("\'", "").replace(" ", "").replace("\n",
    "").strip().split(",")
                        if data[0] == name:
                            print("Data deleted successfully")

                            # output.write(line)
                        elif data[0]!=name:
                            rdt+=1
                            output.write(line)
                    if rd==rdt:
                        print("Name Not found\nEnter the name again")
                        delete_by_name()
            f = open('newfile.txt')
            f1 = open('Drivers_Details.txt', 'r+')
            f1.truncate()
            for x in f.readlines():
                f1.write(x)
            f.close()
            f1.close()
        except ValueError:
            print(("\t\tEnter a valid name"))
            delete_by_name()
        return

    #Sort acoding to the points
    def sorting():
        driver_list = []
        file = open("Drivers_Details.txt","r")
```

```
    contents = file.readlines()
    file.close()

    for x in contents:
        x = x.replace("[", "").replace("]", "").replace("\"",
"").replace("\'", "").replace(" ","").replace("\n", "").strip().split(",")
        driver_list.append(x)

    points = []
    for y in driver_list:
        points.append(int(y[4]))


    max_list = []

    while points:
        max_points = points[0]
        for xy in points:
            if xy > max_points:
                max_points = xy

        max_list.append(max_points)
        points.remove(max_points)

    new_driver = []
    for xxx in max_list:
        # count = 0
        for z in driver_list:

            if int(z[-1]) == xxx:
                new_driver.append(z)
                driver_list.remove(z)
                break

    file = open("Drivers_Details.txt", "w")
    for xvc in new_driver:
        file.write(str(xvc) + "\n")

    file.close()


#Creat random races and positions and update points acording to the positions
#sort acording to the positions
def random_rase_genarator():
    driver_list_rox = []
    location = ["Nyirad", "Holjes", "Montalegre", "Barcelona", "Riga",
"Norway"]
    date = "2022-11-"

    file = open("Drivers_Details.txt", "r")
    driver_no = 0
    no_race = 0
    for x in file:
        driver_no += 1
        no_race += 1
        y = x.replace("[", "").replace("]", "").replace("\"",
"").replace("\'", "").replace(" ", "").replace("\n", "").  \
```

```
        strip().split(",")
    driver_list_rox.append(y)
file.close()

generate_day = []
count1 = 0
while driver_no >= count1:
    i = random.randrange(1, 30)
    if i not in generate_day:
        generate_day.append(i)
        count1 += 1

sort_day = []
while generate_day:
    small_day = generate_day[0]
    for xy in generate_day:
        if xy < small_day:
            small_day = xy

    sort_day.append(small_day)
    generate_day.remove(small_day)

count = 0
f = open("race_table.txt", 'w')
while no_race >= count:
    driver_position = []
    # location
    rand_race = random.choice(location)

    # day
    day = date + str(sort_day[count])

    f.write(str(day) + "\t" + rand_race + "\n")

    random.shuffle(driver_list_rox)
    position = 1

    for y in driver_list_rox:
        driver = []
        driver.append(position)

        count_no = 0
        for x in y:
            count_no += 1
            if count_no < 5:
                driver.append(x)
            if count_no == 5:
                if position == 1:
                    point = int(x) + 10
                    driver.append(point)

                elif position == 2:
                    point = int(x) + 7
                    driver.append(point)

                elif position == 3:
                    point = int(x) + 5
```

```
                    driver.append(point)

                else:
                    driver.append(x)

        f.write(str(driver) + "\n")
        driver_position.append(driver)
        position += 1
    f.write("\n")
    count += 1

    list_update = []
    for y in driver_list_rox:
        list_driver_update = []
        for z in driver_position:
            count22 = 0

            if y[:-1] == z[1:-1]:
                for k in y:
                    if count22 < 4:
                        list_driver_update.append(k)
                    count22 += 1
                list_driver_update.append(z[-1])
        list_update.append(list_driver_update)
    driver_list_rox = list_update

f.close()

f1 = open("Drivers_Details.txt", "w")
for x in driver_list_rox:
    f1.write(str(x) + "\n")
f1.close()
```

iv. ADD

```
list_of_inputs=list()
x=0

#Validate the name
def name():
    file = open("Drivers_Details.txt","r")
    name = []
    for x in file:
        x = x.replace("[", "").replace("]", "").replace("\"",
"").replace("\'", "").replace(" ","").replace("\n", "").strip().split(",")
        name.append(x[0])
    file.close()
    while True:
        try:
            d_name = input("\n\t\tEnter the driver's name\t\t\t: ")
            if d_name not in name:
                d_name.lower()
                list_of_inputs.append(d_name)
                break
            else:
                print("name exist")
                continue
        except:
            break

#Validate the age
def age():
    try:
        d_age =int(input("\t\tEnter driver's age\t\t\t: "))
        list_of_inputs.append(d_age)
    except ValueError:
        print(("\t\tEnter an integer for age"))
        age()
    return

#Team for driver
def team():
    try:
        d_team = input("\t\tEnter the driver's team\t\t\t: ")
        list_of_inputs.append(d_team)
    except  ValueError:
        print(("\t\tEnter string for team"))
        team()
    return

#Input details to car
def car():
    try:
        d_car = input("\t\tEnter driver's car \t\t\t: ")
        list_of_inputs.append(d_car)
    except  :
        car()
    return
```

```
#Input points
def points():
    try:
        d_point = int(input("\t\tEnter the driver's current points\t: "))
        list_of_inputs.append(d_point)

        global x
        x+=1
    except  ValueError:
        print(("\t\tEnter intiger for current points"))
        points()
    return
```

ADD function is used to insert data into the program. Users can input the driver's name, the driver's age, the driver's team, the driver's car, and the driver's current points. When the user enters the proper name, the user can easily move on to the next step. If users enter the name already inserted, the program immediately prints "Name exist" and again asks for the name until a unique name is entered. Then add the name to a list called list_of_inputs. Then ask for age. When the user enters a string value for age, the program immediately prints "Enter an integer for age", Then asks for team and car and finally for current points. All the inputs are appended to the list called list_of_inputs and written into a text file called Drivers_Details.txt. When all the inputs were entered ADD function is over for the moment and again prints the Main menu.

v. DDD

```
#search and deleting part
def delete_by_name():
    try:
        rd=0
        rdt=0
        name = str(input("\n\t\tEnter the driver's name to delete \t\t: "))
        name=name.lower()
        with open("Drivers_Details.txt", "r") as file_input:
            with open("newfile.txt", "w") as output:
                for line in file_input:
                    rd+=1
                    data = line.replace("[", "").replace("]",
"").replace("\"", "").replace("\'", "").replace(" ", "").replace("\n",
"").strip().split(",")
                    if data[0] == name:
                        print("Data deleted successfully")

                        # output.write(line)
                    elif data[0]!=name:
```

```
                        rdt+=1
                        output.write(line)
                if rd==rdt:
                    print("Name Not found\nEnter the name again")
                    delete_by_name()
        f = open('newfile.txt')
        f1 = open('Drivers_Details.txt', 'r+')
        f1.truncate()
        for x in f.readlines():
            f1.write(x)
        f.close()
        f1.close()
    except ValueError:
        print(("\t\tEnter a valid name"))
        delete_by_name()
    return

#Function2 DDD
def ddd():
    validation.delete_by_name()
```

DDD function is used to delete data from the saved text file. “Enter the driver’s name to delete” is displayed in the DDD function. When the user inputs the correct name that is in the Drivers_Details.txt can be removed. If the user inserts another name program Display “Enter a valid name”. When the user inserts the name in Drivers_Details.txt, all the data about that driver is removed from a text file.

vi. UDD

```
#Function3 UDD
def udd():
        ddd()
        print("Enter details again")
        add()
```

UDD function is used to update drivers’ details. In this program, the update is the sum of the DDD function and ADD function. Ask from user to insert the name of the driver that the user wants to update the details. When the user inserts a name, the program deleted the records about that driver and asks for All the details mentioned in ADD function, and saves new details into a text file called ”Drivers_Details.txt”.

vii. VCT

```
#Sort acoding to the points
def sorting():
    driver_list = []
    file = open("Drivers_Details.txt","r")
    contents = file.readlines()
    file.close()

    for x in contents:
        x = x.replace("[", "").replace("]", "").replace("\"",
"").replace("\'", "").replace(" ","").replace("\n", "").strip().split(",")
        driver_list.append(x)

    points = []
    for y in driver_list:
        points.append(int(y[4]))


    max_list = []

    while points:
        max_points = points[0]
        for xy in points:
            if xy > max_points:
                max_points = xy

        max_list.append(max_points)
        points.remove(max_points)

    new_driver = []
    for xxx in max_list:
        # count = 0
        for z in driver_list:

            if int(z[-1]) == xxx:
                new_driver.append(z)
                driver_list.remove(z)
                break

    file = open("Drivers_Details.txt", "w")
    for xvc in new_driver:
        file.write(str(xvc) + "\n")

    file.close()


#Function4 VCT
def vct():
    validation.sorting()
    print("\n")
    print("  'NAME',    'AGE',    'TEAM',    'CAR',    'POINTS'")
    with open("Drivers_Details.txt") as r:
        print(r.read())
```

VCT function is used to get a table of driver details according to the points by displaying a table in the python console. VCT function loops the driver points one by one and rearranges them with the highest points to the lowest points from inserted data. Then display the sorted data.

viii. SRR

```
#Creat random races and positions and update points acording to the positions
#sort acording to the positions
def random_rase_genarator():
    driver_list_rox = []
    location = ["Nyirad", "Holjes", "Montalegre", "Barcelona", "Riga",
"Norway"]
    date = "2022-11-"

    file = open("Drivers_Details.txt", "r")
    driver_no = 0
    no_race = 0
    for x in file:
        driver_no += 1
        no_race += 1
        y = x.replace("[", "").replace("]", "").replace("\"",
"").replace("\'", "").replace(" ", "").replace("\n", ""). \
            strip().split(",")
        driver_list_rox.append(y)
    file.close()

    generate_day = []
    count1 = 0
    while driver_no >= count1:
        i = random.randrange(1, 30)
        if i not in generate_day:
            generate_day.append(i)
            count1 += 1

    sort_day = []
    while generate_day:
        small_day = generate_day[0]
        for xy in generate_day:
            if xy < small_day:
                small_day = xy

        sort_day.append(small_day)
        generate_day.remove(small_day)

    count = 0
    f = open("race_table.txt", 'w')
    while no_race >= count:
        driver_position = []
        # location
        rand_race = random.choice(location)

        # day
```

```
        day = date + str(sort_day[count])

        f.write(str(day) + "\t" + rand_race + "\n")

        random.shuffle(driver_list_rox)
        position = 1

        for y in driver_list_rox:
            driver = []
            driver.append(position)

            count_no = 0
            for x in y:
                count_no += 1
                if count_no < 5:
                    driver.append(x)
                if count_no == 5:
                    if position == 1:
                        point = int(x) + 10
                        driver.append(point)

                    elif position == 2:
                        point = int(x) + 7
                        driver.append(point)

                    elif position == 3:
                        point = int(x) + 5
                        driver.append(point)

                    else:
                        driver.append(x)

            f.write(str(driver) + "\n")
            driver_position.append(driver)
            position += 1
        f.write("\n")
        count += 1

        list_update = []
        for y in driver_list_rox:
            list_driver_update = []
            for z in driver_position:
                count22 = 0

                if y[:-1] == z[1:-1]:
                    for k in y:
                        if count22 < 4:
                            list_driver_update.append(k)
                        count22 += 1
                    list_driver_update.append(z[-1])
            list_update.append(list_driver_update)
        driver_list_rox = list_update

f.close()

f1 = open("Drivers_Details.txt", "w")
for x in driver_list_rox:
```

```
        f1.write(str(x) + "\n")
    f1.close()

#Function5 SRR
def srr():
   validation.random_rase_genarator()
```

When the user types SRR program simulates random races in different locations and on different dates. In this function, the program adds points to the driver's current points according to the positions that the program randomly simulates. Then all updated points add to the text file, also race simulating details insert into another text file called race_table.txt.

ix. VRL

```
#Function6 VRL
def vrl():
    with open("race_table.txt") as r:
        print(r.read())
```

The VRL option, program should have to display data about races that are simulated in the SRR function above. Also, the program should sort data according to the date and save it into a text file. Then display data when the user type 'VRL'.

x. STF

```
#Function7 STF
def stf():
    print("Data Saved into text file called 'Drivers_Details.txt'")
```

STF function is used to save the driver's data into the text file. But in ADD function program write data into a text file. Therefore, in STF option program has nothing to do. Then the program only display "Data saved into a text file called 'Drvers_Details.txt'"

xi. RFF

```
#Function8 RFF
def rff():
    print("\n")
    print("  'NAME',   'AGE',   'TEAM',   'CAR',   'POINTS'")
    with open("Drivers_Details.txt") as r:
        print (r.read())
```

RFF function is used to load all the entered details in the text file. Therefore, program simply opens the text file and prints all the data in it.

xii. ESC

```
#Function9 ESC
def esc():
    print ("Program Terminated")

import options
import validation

# Main program
while True:
    options.main_menu()
    opt = input("\n\tType the option you want : ")
    opt = opt.upper()
    if opt == 'ADD':
        options.add()
    elif opt == 'DDD':
        options.ddd()
    elif opt == 'UDD':
        options.udd()
    elif opt == 'VCT':
        options.vct()
    elif opt == 'SRR':
        options.srr()
    elif opt == 'VRL':
        options.vrl()
    elif opt == 'STF':
        options.stf()
    elif opt == 'RFF':
        options.rff()
    elif opt == 'ESC':
        options.esc()
        break
    else:
        print("\nInvalid option, Enter a valid option again")
        continue
```

ESC function is used to terminate the program. When the user inserts ‘ECS’ for the user’s option program should be terminated. For the program termination process program use breaking while loop method.

3. Test cases and plan

Test case	inputs	Expected output	Actual output	Remarks
1	ADD,Seth,18,SL Riders,GTR,242	Data entered Successfully. Main menu	Data entered Successfully. Main menu	PASS
2	ADD,Seth	Name exist	Name exist	PASS
3	DDD,Sankalpa	Data deleted successfully. Main menu	Data deleted successfully. Main menu	PASS
4	DDD,Ammar	Name Not found Enter the name again	Name Not found Enter the name again	PASS
5	UDD,Seth,seth,19, ,SL Riders ,GTR ,242	Data deleted successfully. Enter details again Data entered successfully. Main menu	Data deleted successfully. Enter details again Data entered successfully. Main menu	PASS
6	UDD,sankalpa	Name Not found Enter the name again	Name Not found Enter the name again	PASS

7	VCT	Display table sorted according to points Main menu	Display table sorted according to points Main menu	PASS
8	SRR	Main menu	Main menu	PASS
9	VRL	Display Simulated races with winners' details Main menu	Display Simulated races with winners' details Main menu	PASS
10	STF	Data Saved into text file called 'Drivers_Details.txt Main menu	Data Saved into text file called 'Drivers_Details.txt Main menu	PASS
11	RFF	Display Drivers_details.txt Main menu	Display Drivers_details.txt Main menu	PASS
12	ECS	Terminate program	Terminate program	PASS
13	EDG	Invalid option, Enter a valid option again Main menu	Invalid option, Enter a valid option again Main menu	PASS

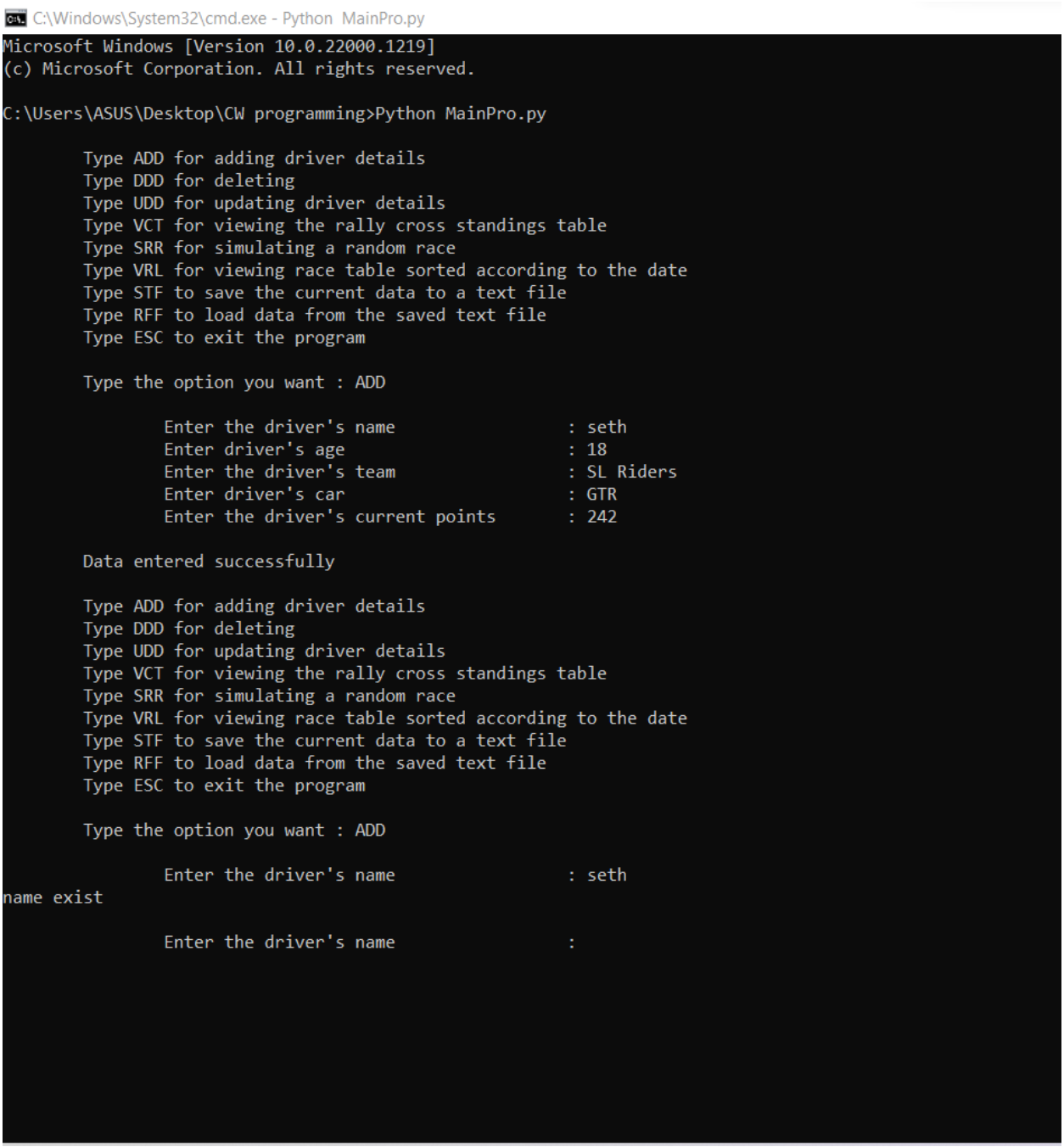

Figure 3 Test Case 1 & 2

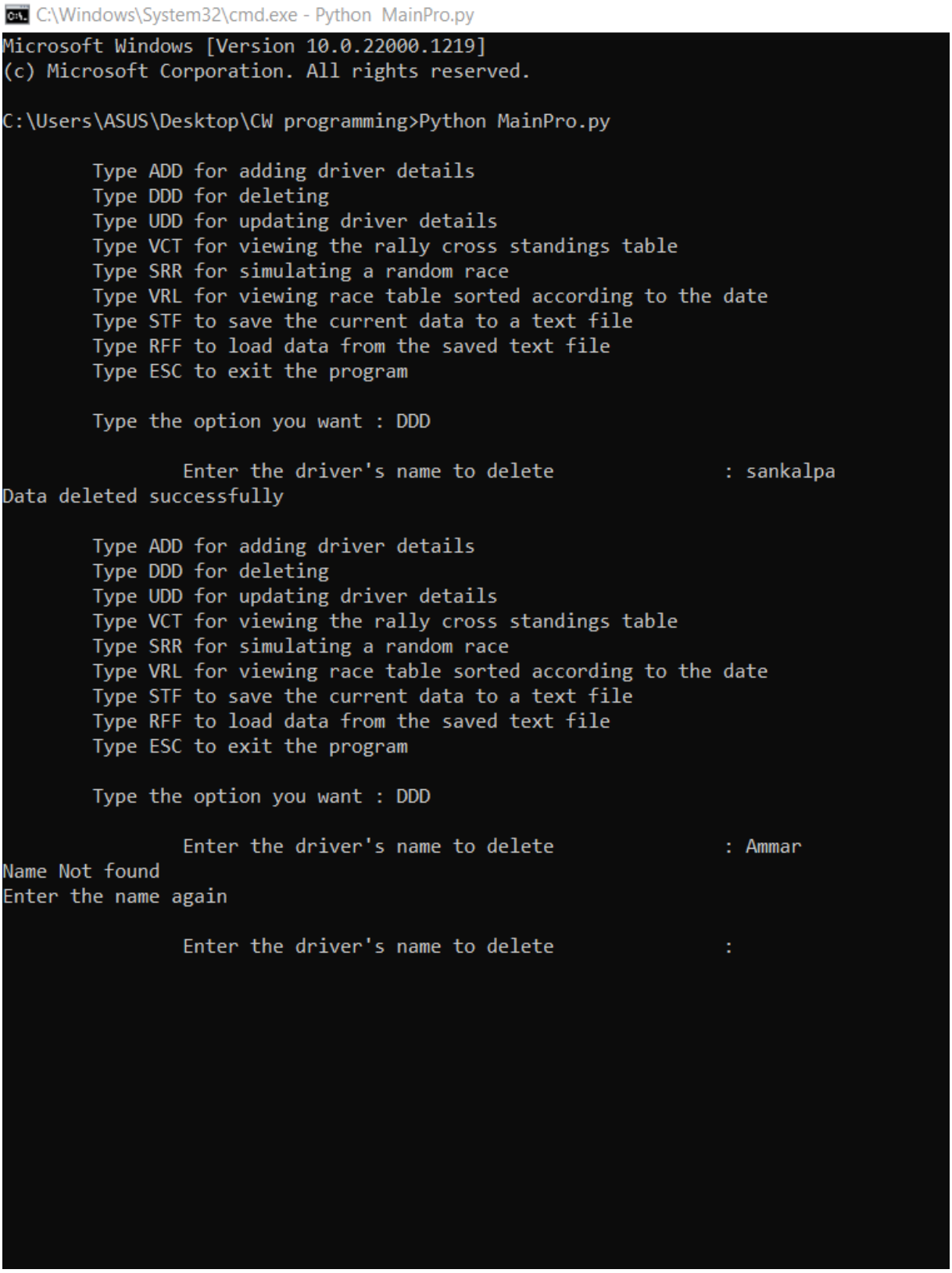

Figure 4 Test Case 3& 4

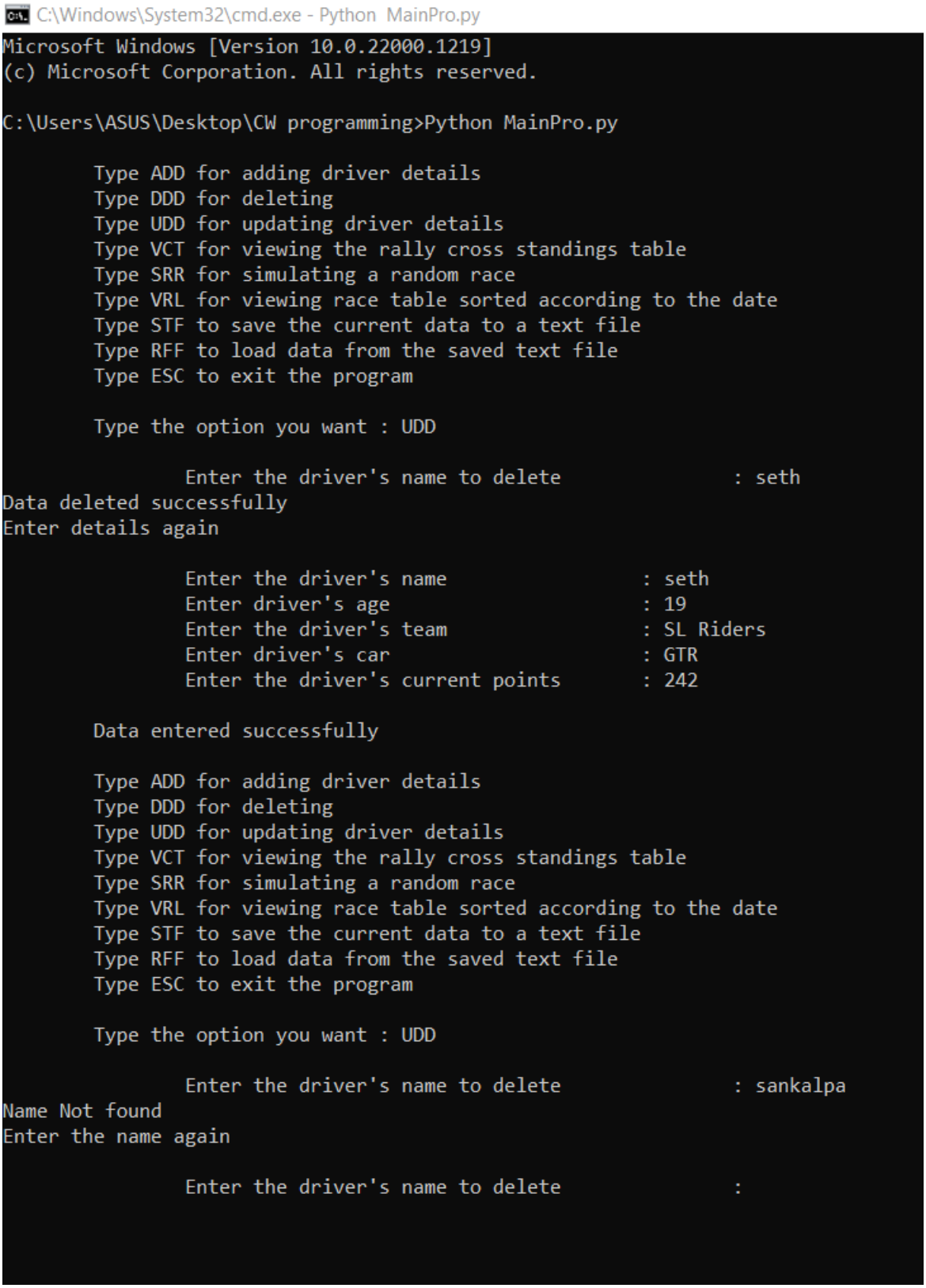

Figure 5 Test Case 5 & 6

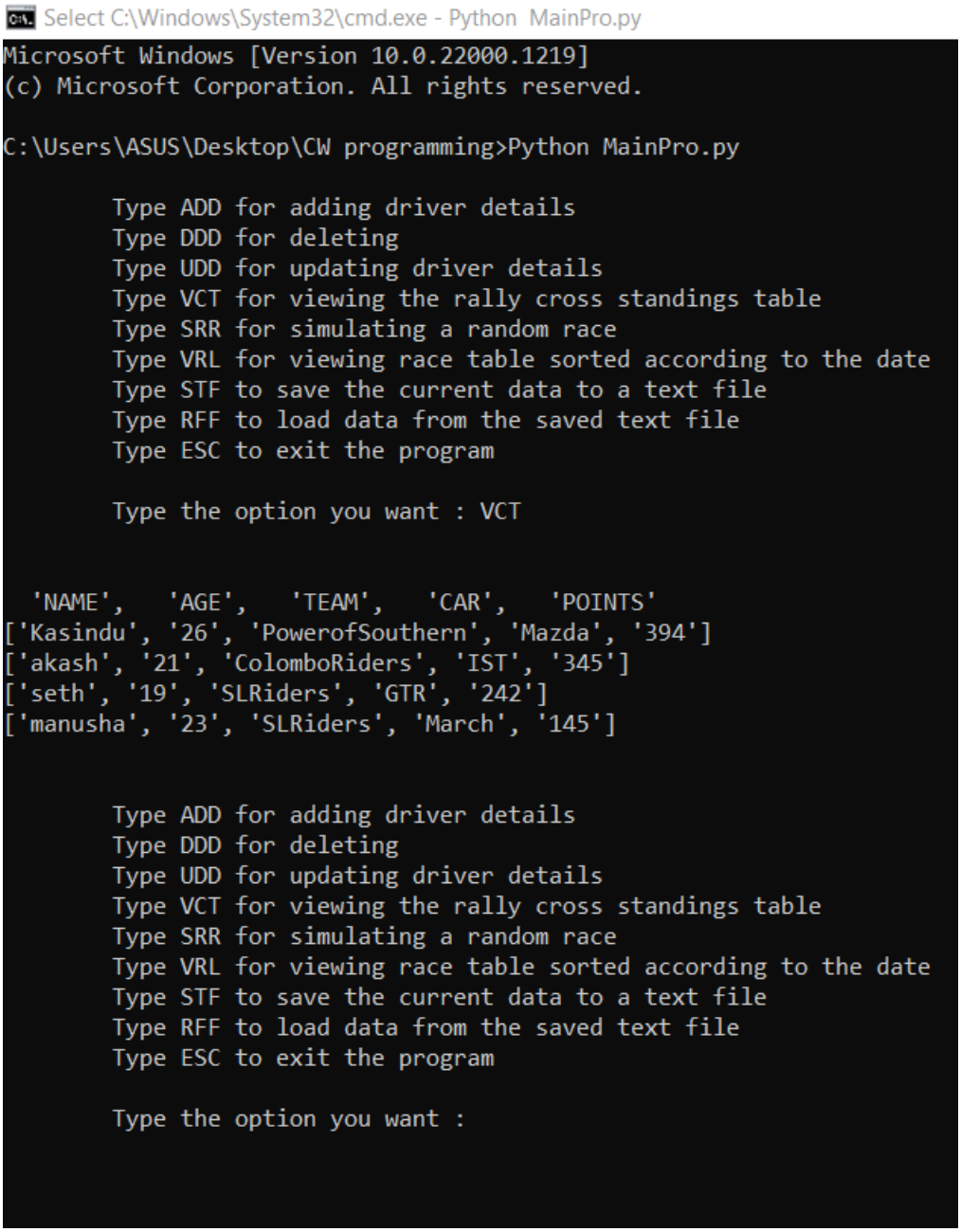

Figure 6 Test Case 7

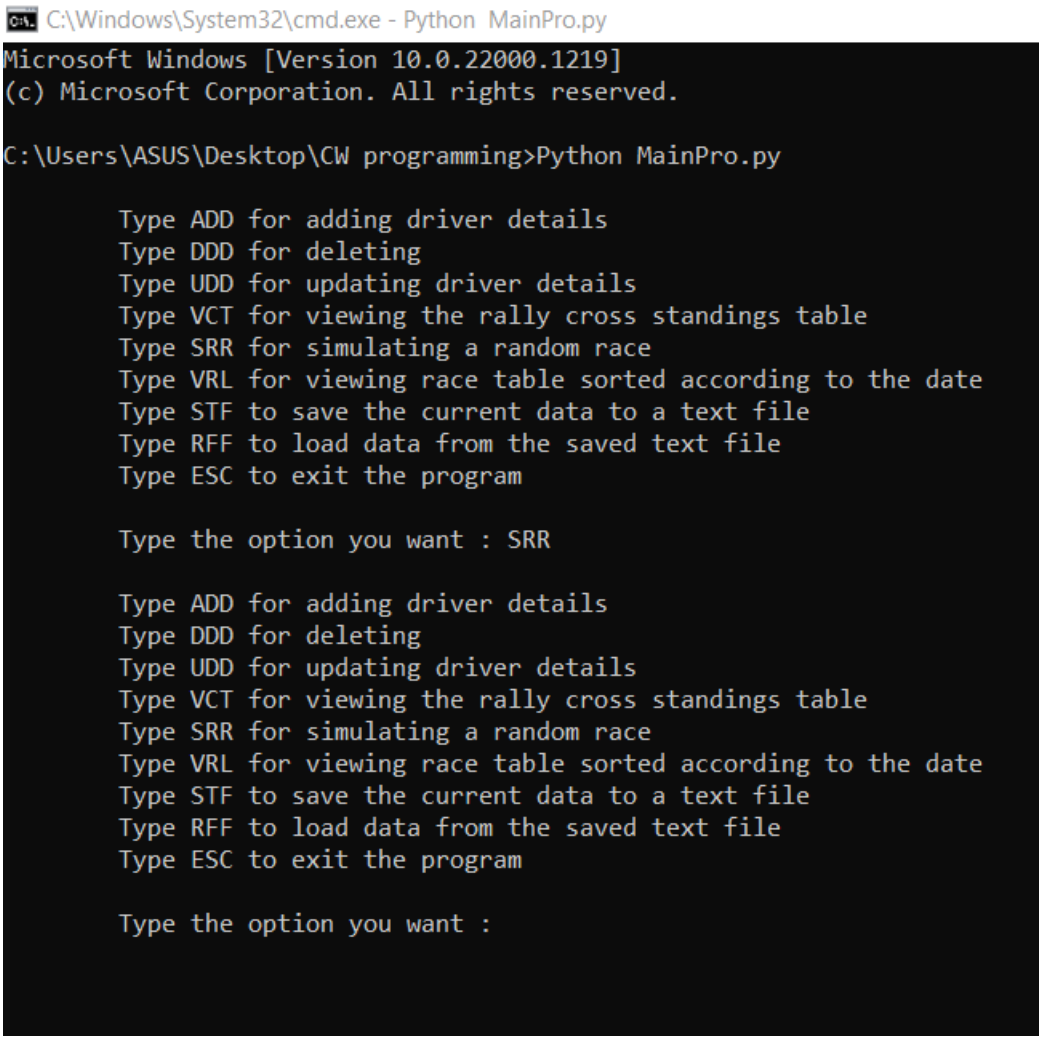

Figure 7 Test Case 8

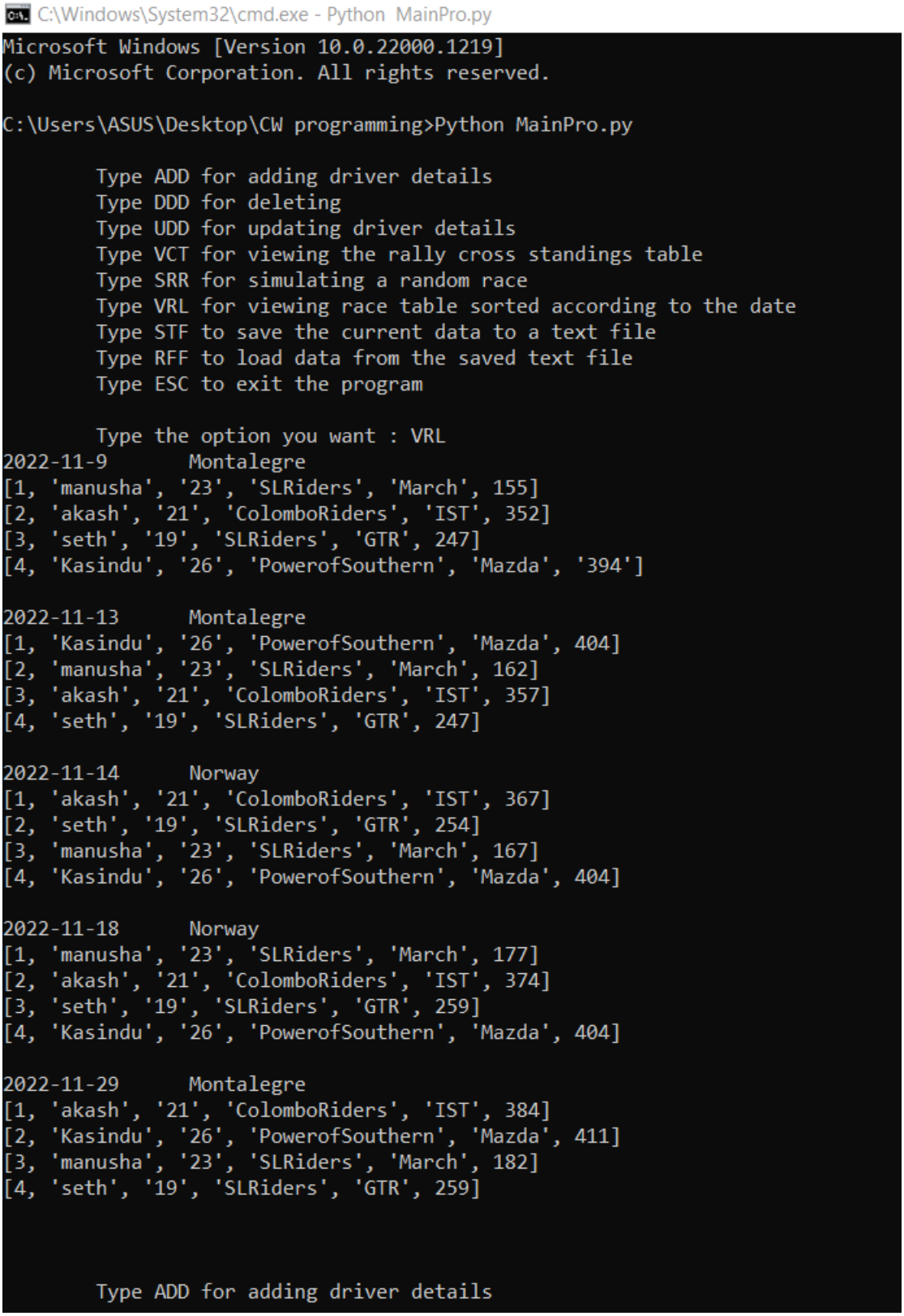

Figure 8 Test Case 9

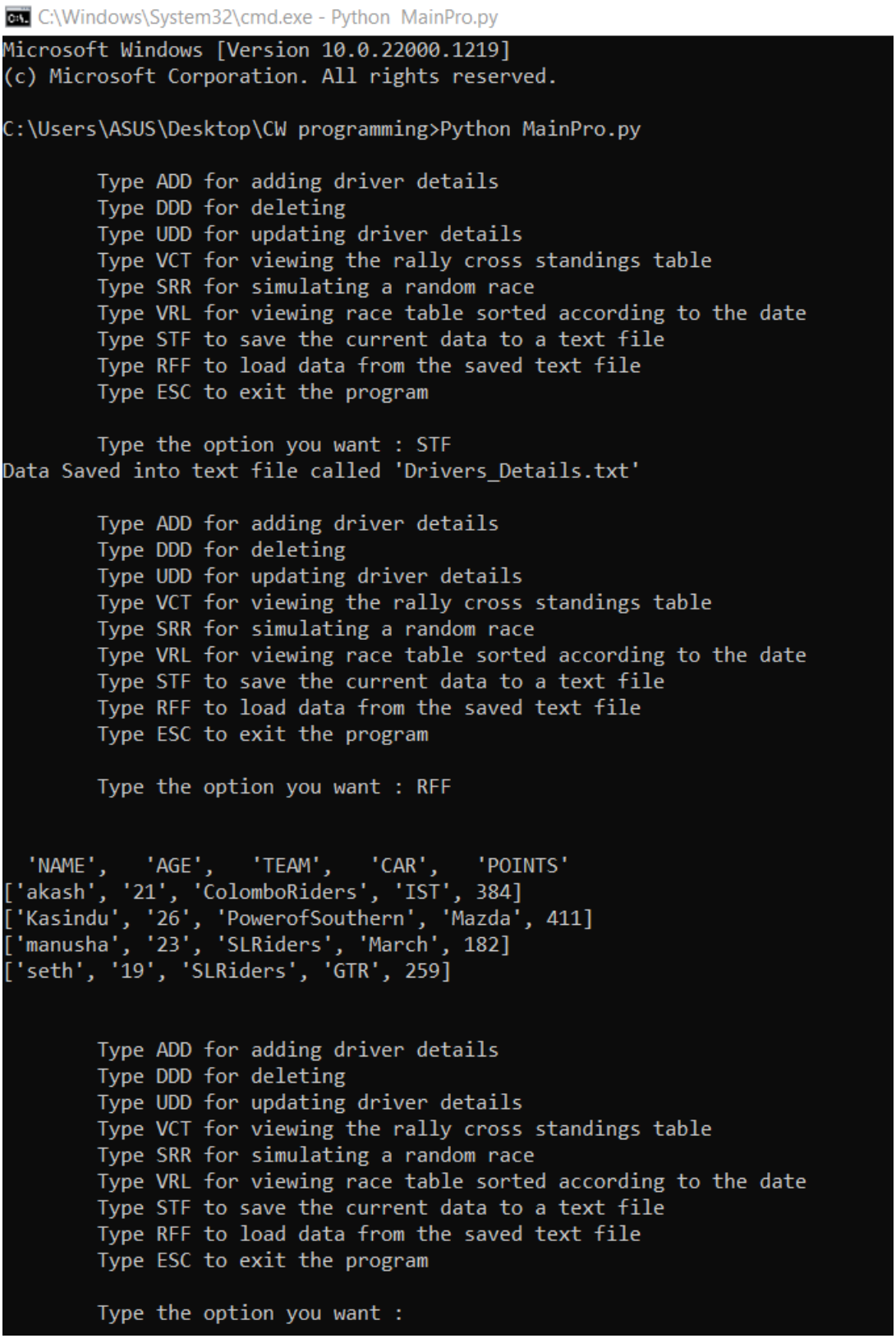

Figure 9 Test Case 10 & 11

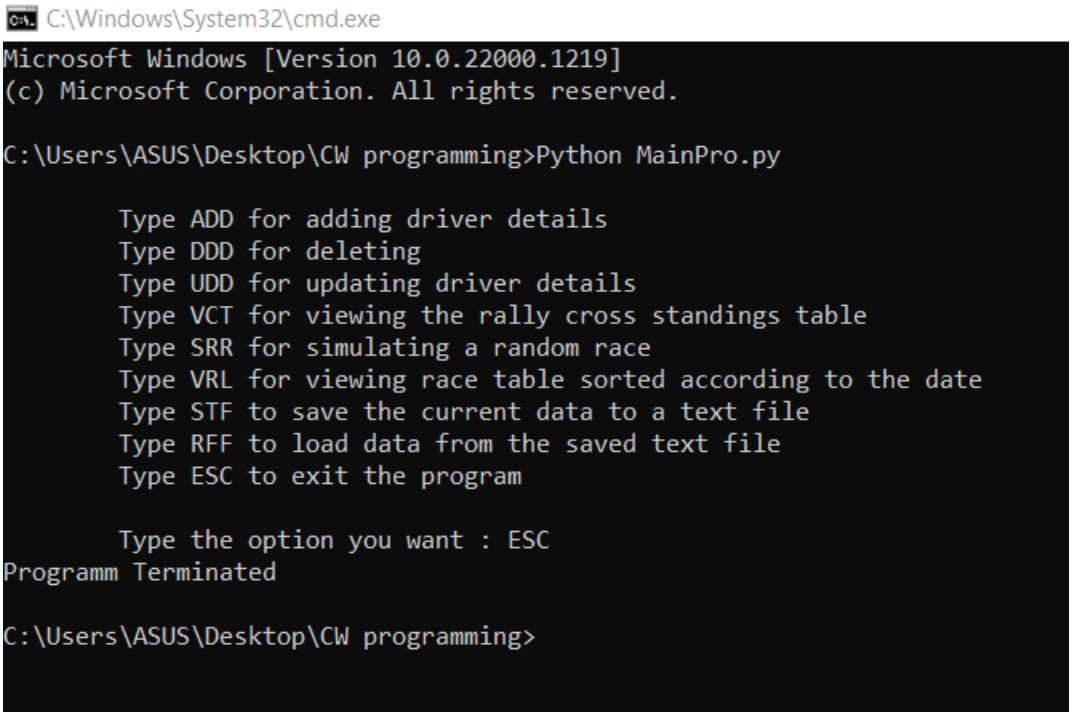

Figure 10 Test Case 12

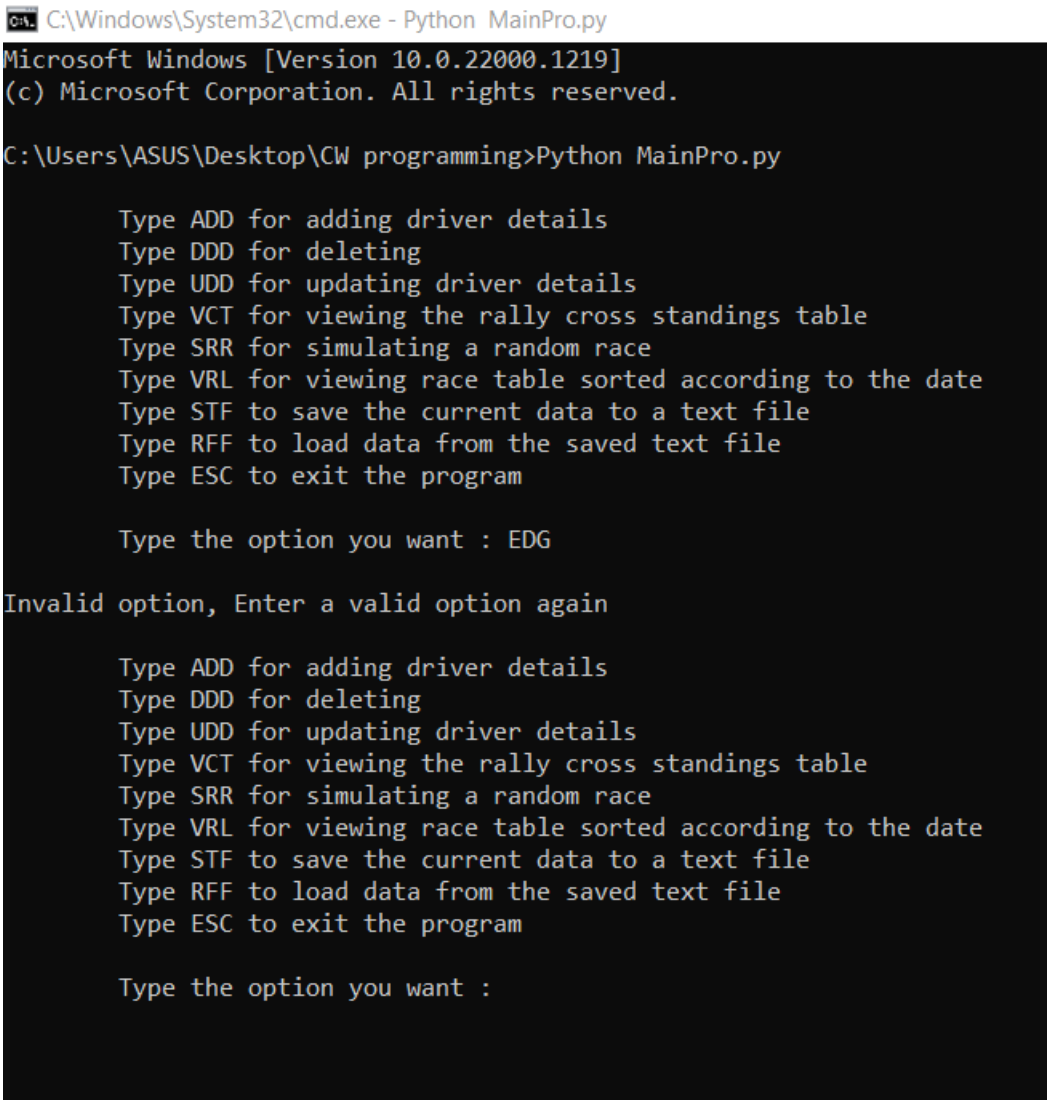

Figure 11 Test Case 13

4. Conclusion

This program is designed to manage the World Rallycross Championship. The program also allows users to update their details using the options provided to remove their name. The program can simulate a random race and assign a position and points to each driver. This program will be useful for those who races like this.

5. Reference list

- W3Schools,2019. Random Numbers in NumPy [online]. W3Schools. Available from: https://www.w3schools.com/python/numpy/numpy_random.asp [Accessed 3 December 2022]
- W3Schools,2019. Python Lists[online]. W3Schools. Available from: https://www.w3schools.com/python/python_lists.asp [Accessed 3 December 2022]
- Stackoverflow,2021. How do I print the content of a .txt file in python?[online]. Stackoverflow. Available from: https://stackoverflow.com/questions/18256363/how-do-i-print-the-content-of-a-txt-file-in-python [Accessed 1 December 2022]
- Stackoverflow,2021. How to delete a specific line in a file?[online]. Stackoverflow. Available from: https://stackoverflow.com/questions/4710067/how-to-delete-a-specific-line-in-a-file [Accessed 1 December 2022]
- PYnative,2021. Writing list to a file in python[online]. PYnative. Available from: https://pynative.com/python-write-list-to-file/ [Accessed 1 December 2022]
- Stackoverflow,2010. Writing a list to a file with python, with new lines[online]. Stackoverflow. Available from: https://stackoverflow.com/questions/899103/writing-a-list-to-a-file-with-python-with-newlines [Accessed 1 December 2022]
- Stackoverflow,2017. Sorting user inputs in a list and writing a loop to find a valid value from that list[online]. Stackoverflow. Available from: https://stackoverflow.com/questions/43141691/storing-user-input-in-a-list-and-writing-a-loop-to-find-a-valid-value-from-that [Accessed 1 December 2022]