**ASSIGNMENT - 2**

1. **Write Python code to Create database ‘Student\_Information.db’**

import sqlite3

try:

conn = sqlite3.connect("Student\_Information.db")

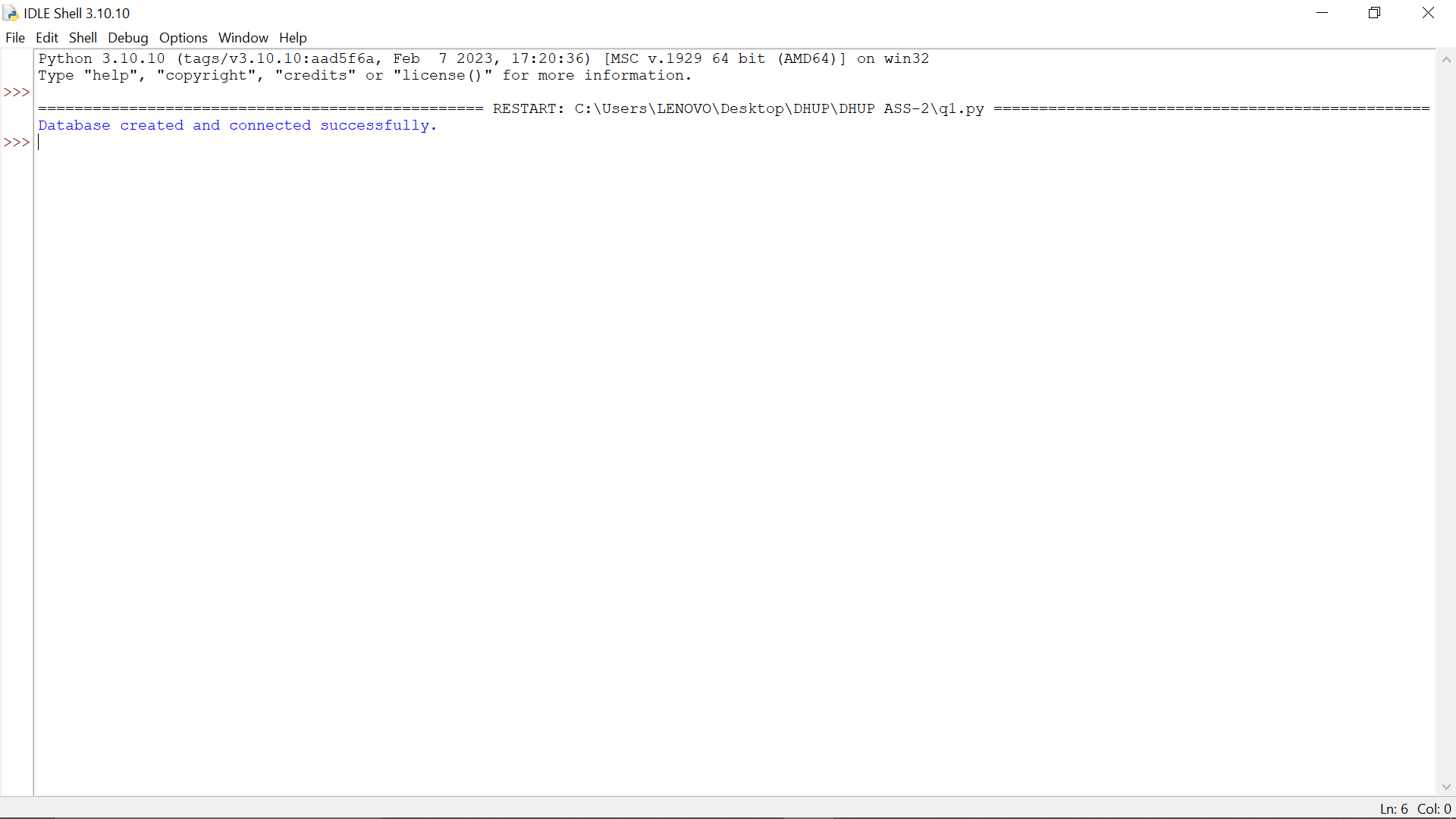
print("Database created and connected successfully.")

except Exception as e:

print("Error creating database:", e)

finally:

conn.close()



**2. Write python code to Create table Student with following constrains.**

**a. RollNumber text primary key**

**b. Name text Not null**

**c. PYTHON int not null marks must greater than 0 and less than 100**

**d. OOPS int not null marks must greater than 0 and less than 100**

**e. WEB int not null marks must greater than 0 and less than 100**

**f. MIL int not null marks must greater than 0 and less than 100**

**g. STATE int not null marks must greater than 0 and less than 100**

import sqlite3

try:

conn = sqlite3.connect("Student\_Information.db")

cur = conn.cursor()

cur.execute("""

CREATE TABLE IF NOT EXISTS Student (

RollNumber TEXT PRIMARY KEY,

Name TEXT NOT NULL,

PYTHON INTEGER NOT NULL CHECK(PYTHON > 0 AND PYTHON < 100),

OOPS INTEGER NOT NULL CHECK(OOPS > 0 AND OOPS < 100),

WEB INTEGER NOT NULL CHECK(WEB > 0 AND WEB < 100),

MIL INTEGER NOT NULL CHECK(MIL > 0 AND MIL < 100),

STATE INTEGER NOT NULL CHECK(STATE > 0 AND STATE < 100)

)

""")

print("Table created successfully.")

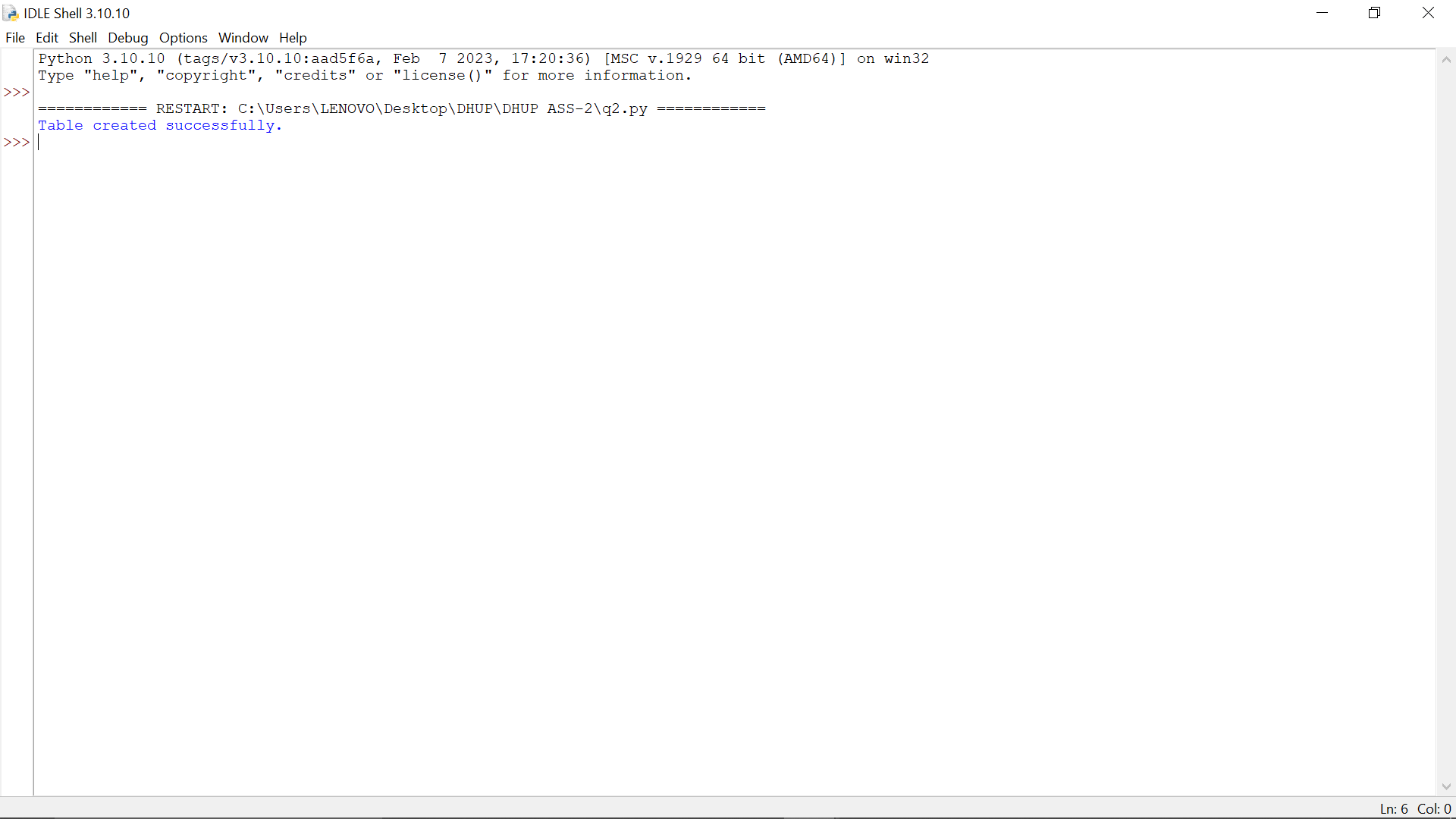
except Exception as e:

print("Error creating table:", e)

finally:

conn.commit()

conn.close()

****

**3. Write python code to create trigger named as ‘rollnumbercheck’ which check for the RollNumber must starts with ‘R/r’.**

import sqlite3

try:

conn = sqlite3.connect("Student\_Information.db")

cur = conn.cursor()

cur.execute("""

CREATE TRIGGER IF NOT EXISTS rollnumbercheck

BEFORE INSERT ON Student

FOR EACH ROW

BEGIN

SELECT CASE

WHEN NEW.RollNumber NOT LIKE 'R%' AND NEW.RollNumber NOT LIKE 'r%'

THEN RAISE(ABORT, 'RollNumber must start with R or r')

END;

END;

""")

print("Trigger created successfully.")

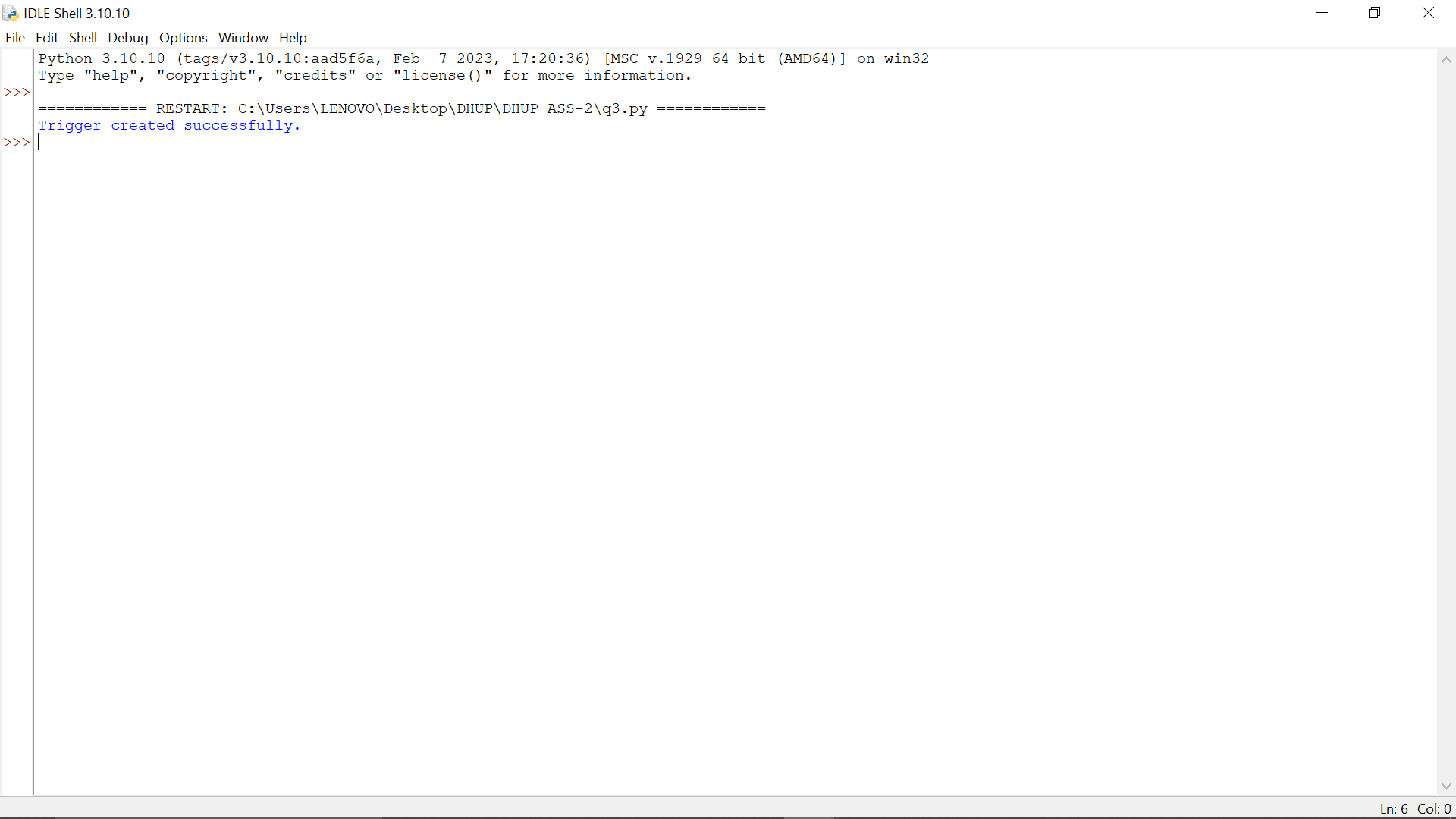
except Exception as e:

print("Error creating trigger:", e)

finally:

conn.commit()

conn.close()

****

**4. Write python code to insert user inputted data into the table by taking appropriate inputs from the user. (Ask user to take number of students to be inserted)**

import sqlite3

try:

conn = sqlite3.connect("Student\_Information.db")

cur = conn.cursor()

n = int(input("Enter number of students: "))

for \_ in range(n):

roll = input("Enter Roll Number: ")

name = input("Enter Name: ")

python\_m = int(input("PYTHON marks: "))

oops\_m = int(input("OOPS marks: "))

web\_m = int(input("WEB marks: "))

mil\_m = int(input("MIL marks: "))

state\_m = int(input("STATE marks: "))

cur.execute("INSERT INTO Student VALUES (?, ?, ?, ?, ?, ?, ?)",

(roll, name, python\_m, oops\_m, web\_m, mil\_m, state\_m))

print("Data inserted successfully.")

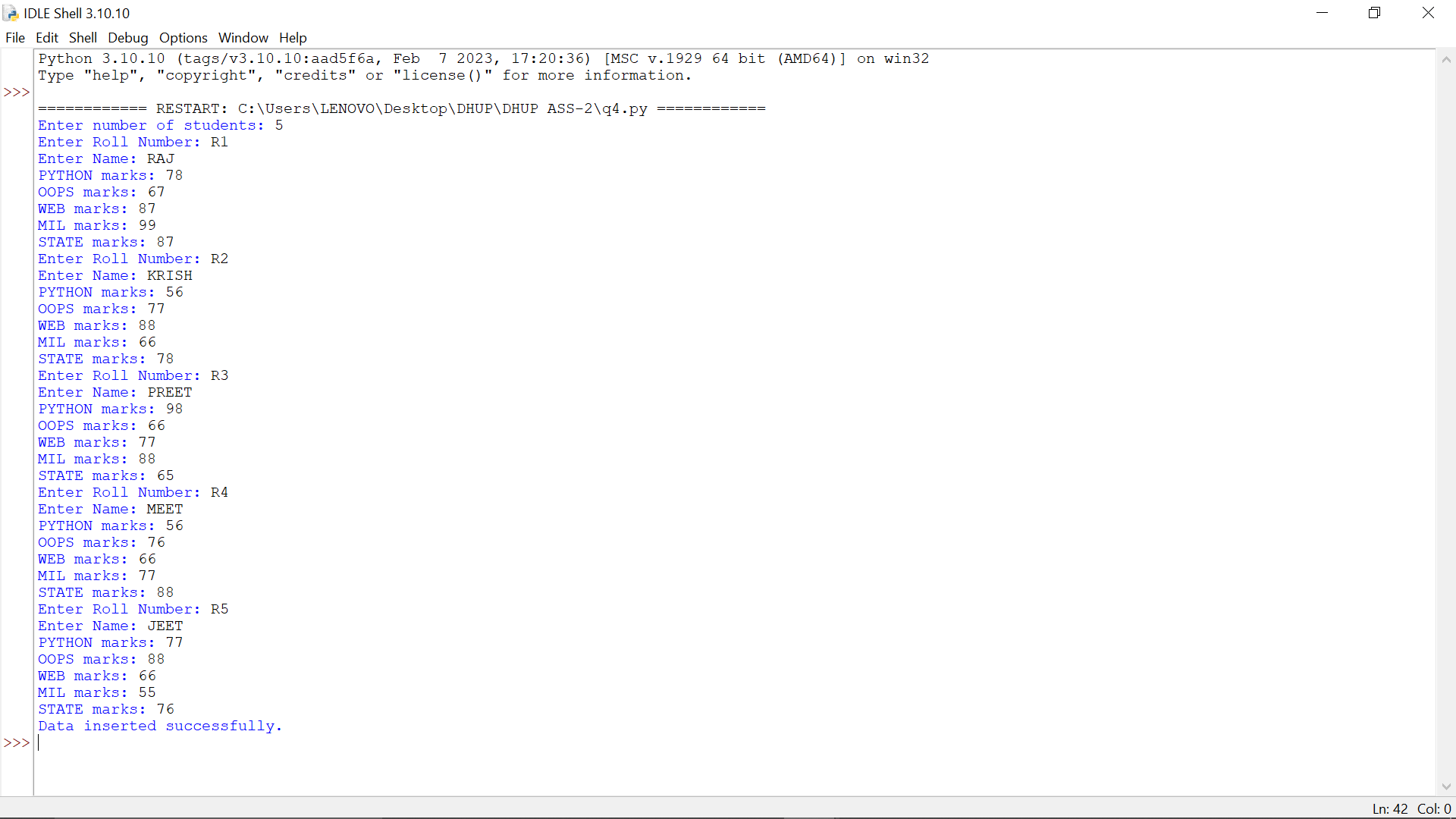
except Exception as e:

print("Error inserting data:", e)

finally:

conn.commit()

conn.close()

****

**5. Write a python code to fetch all the data from the table and display it in appropriate table.**

import sqlite3

import pandas as pd

try:

conn = sqlite3.connect("Student\_Information.db")

df = pd.read\_sql\_query("SELECT \* FROM Student", conn)

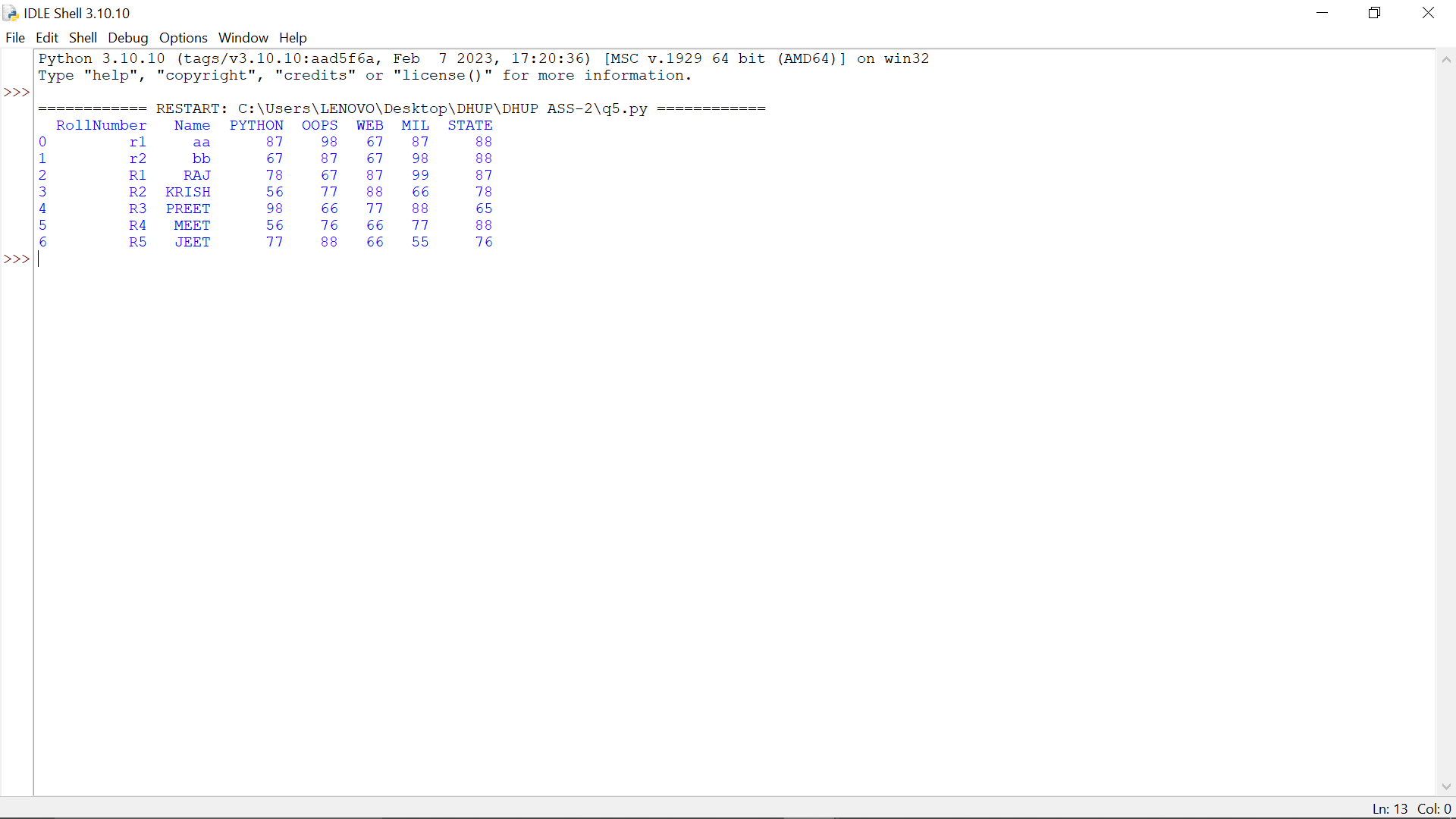
print(df)

except Exception as e:

print("Error fetching data:", e)

finally:

conn.close()

****

**6. Write a python code to fetch all the records in data frame. Use the appropriate method to describe all the data.**

import sqlite3

import pandas as pd

try:

conn = sqlite3.connect("Student\_Information.db")

df = pd.read\_sql\_query("SELECT \* FROM Student", conn)

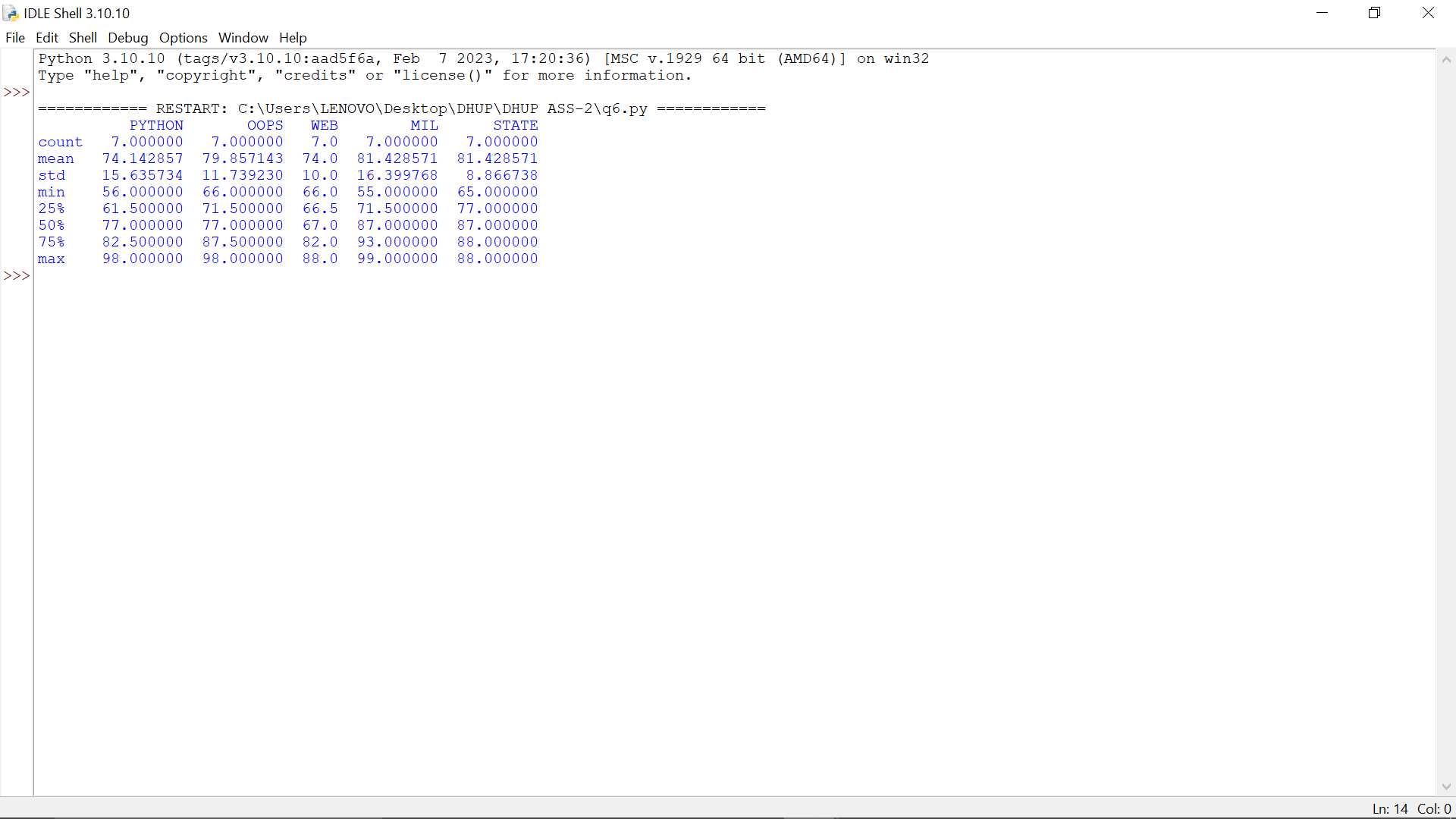
print(df.describe())

except Exception as e:

print("Error describing data:", e)

finally:

conn.close()

****

**7. Write a python code to add following columns to data frame. Use appropriate methods of pandas module.**

**a. Total**

**b. Percentage**

**c. Minimum**

**d. Maximum**

import sqlite3

import pandas as pd

try:

conn = sqlite3.connect("Student\_Information.db")

df = pd.read\_sql\_query("SELECT \* FROM Student", conn)

df['Total'] = df[['PYTHON','OOPS','WEB','MIL','STATE']].sum(axis=1)

df['Percentage'] = df['Total'] / 5

df['Minimum'] = df[['PYTHON','OOPS','WEB','MIL','STATE']].min(axis=1)

df['Maximum'] = df[['PYTHON','OOPS','WEB','MIL','STATE']].max(axis=1)

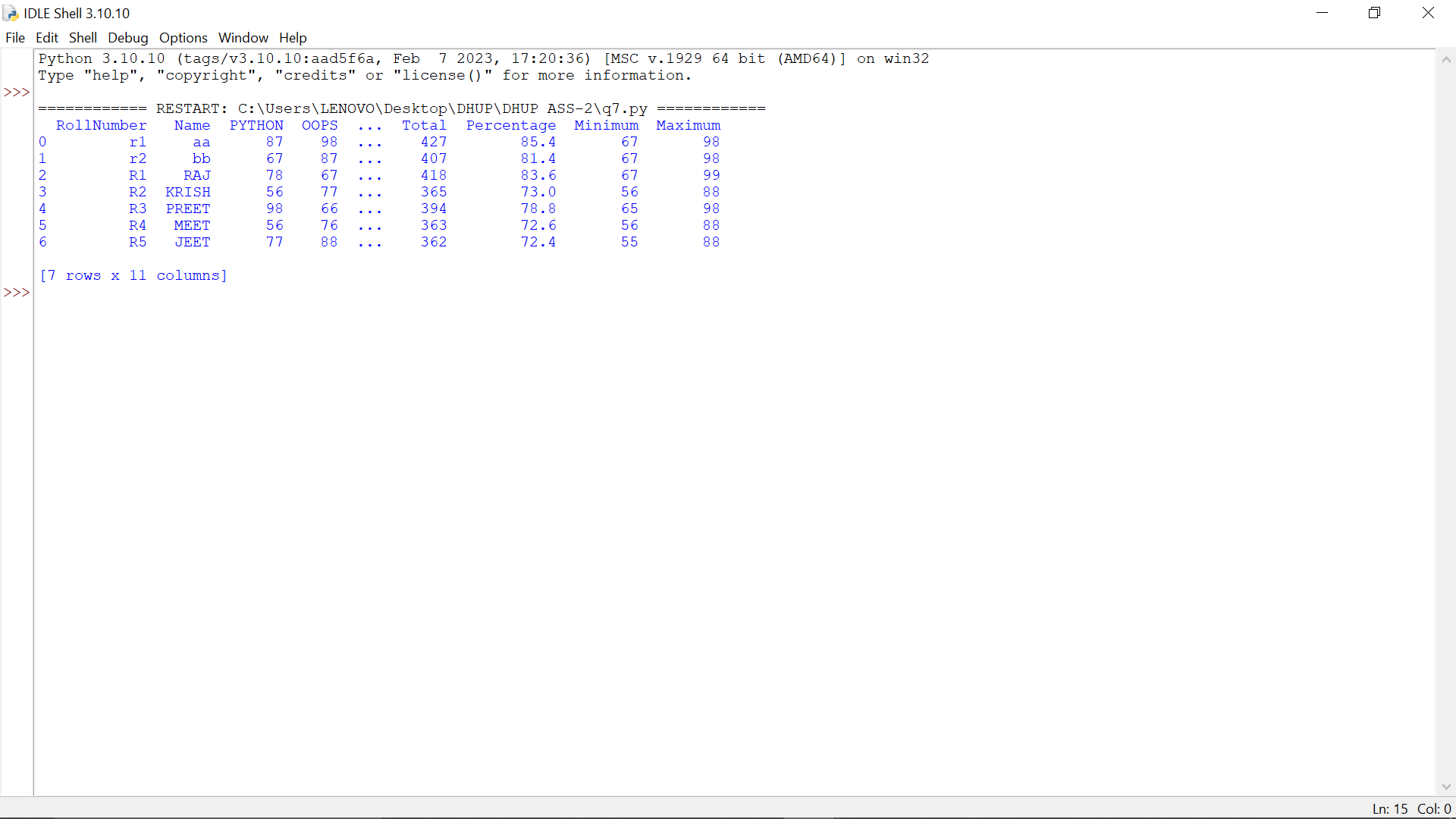
print(df)

except Exception as e:

print("Error adding columns:", e)

finally:

conn.close()

****

**8. Write a python code to write the data frame in the csv file. Name csv file as “studentinfo.csv”**

import sqlite3

import pandas as pd

try:

conn = sqlite3.connect("Student\_Information.db")

df = pd.read\_sql\_query("SELECT \* FROM Student", conn)

df.to\_csv("studentinfo.csv", index=False)

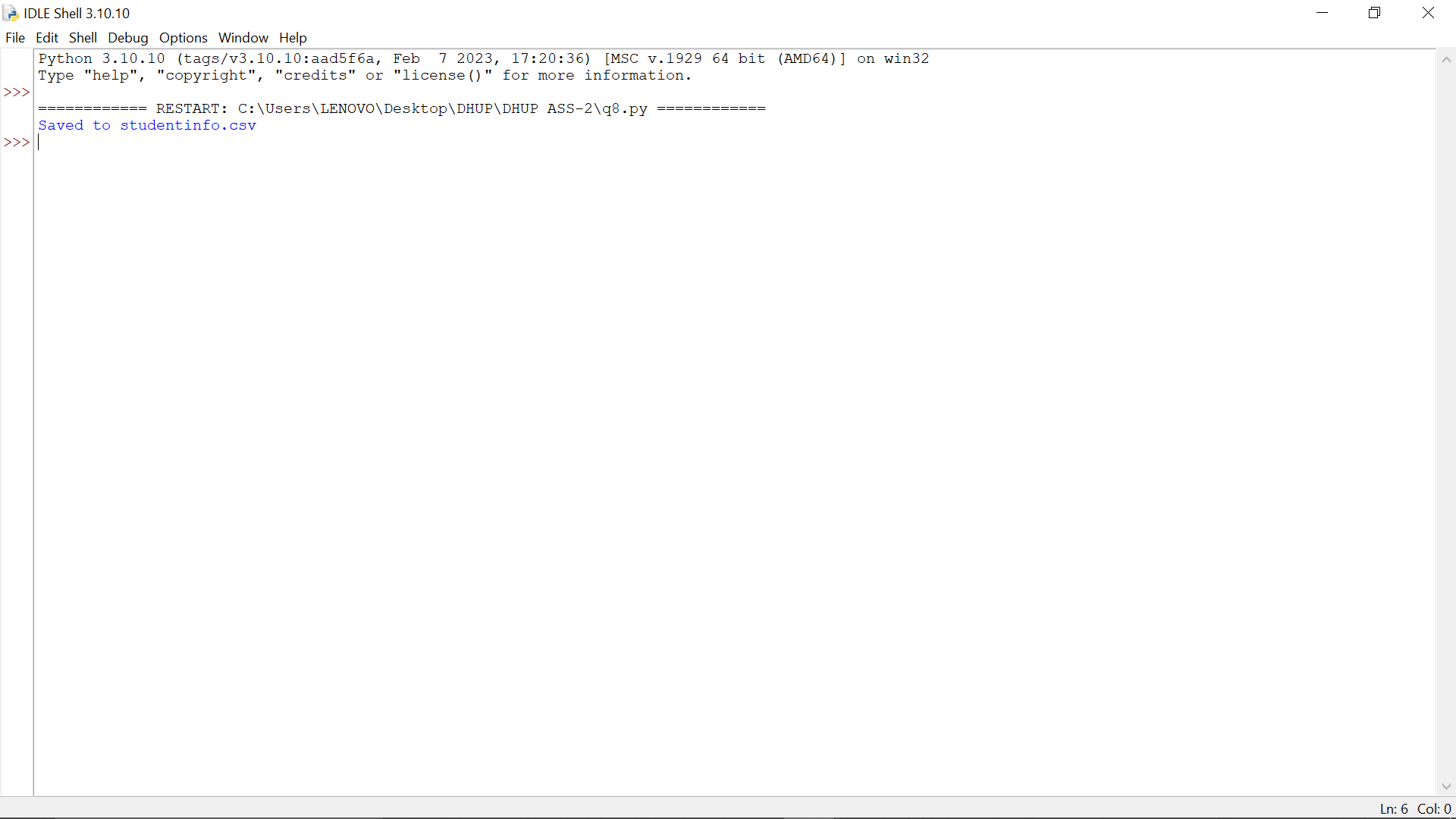
print("Saved to studentinfo.csv")

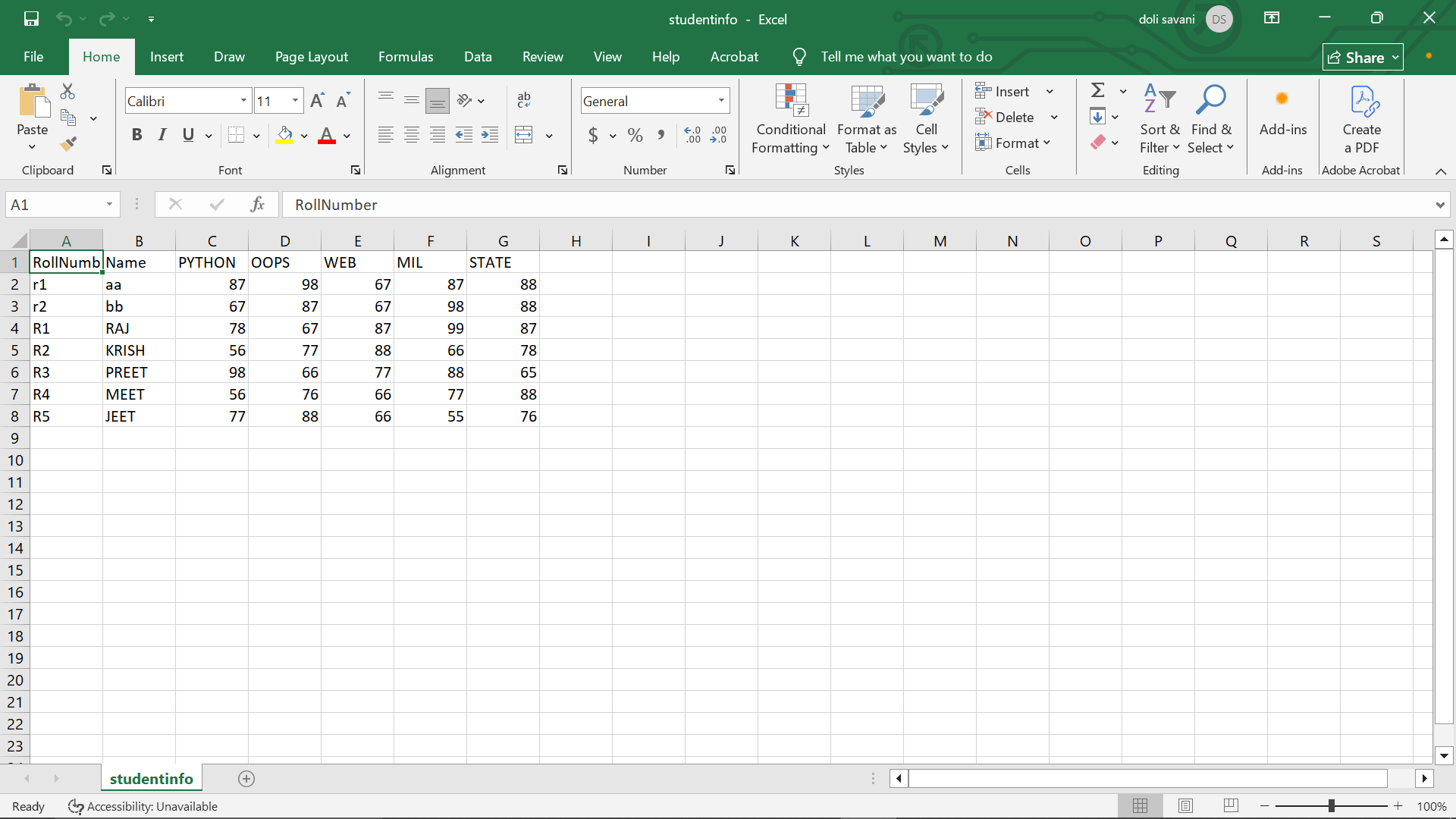
except Exception as e:

print("Error saving CSV:", e)

finally:

conn.close()

****

****

**9. Convert the data frame into the numpy ndarray and display it.**

import sqlite3

import pandas as pd

import numpy as np

try:

conn = sqlite3.connect("Student\_Information.db")

df = pd.read\_sql\_query("SELECT \* FROM Student", conn)

arr = df.to\_numpy()

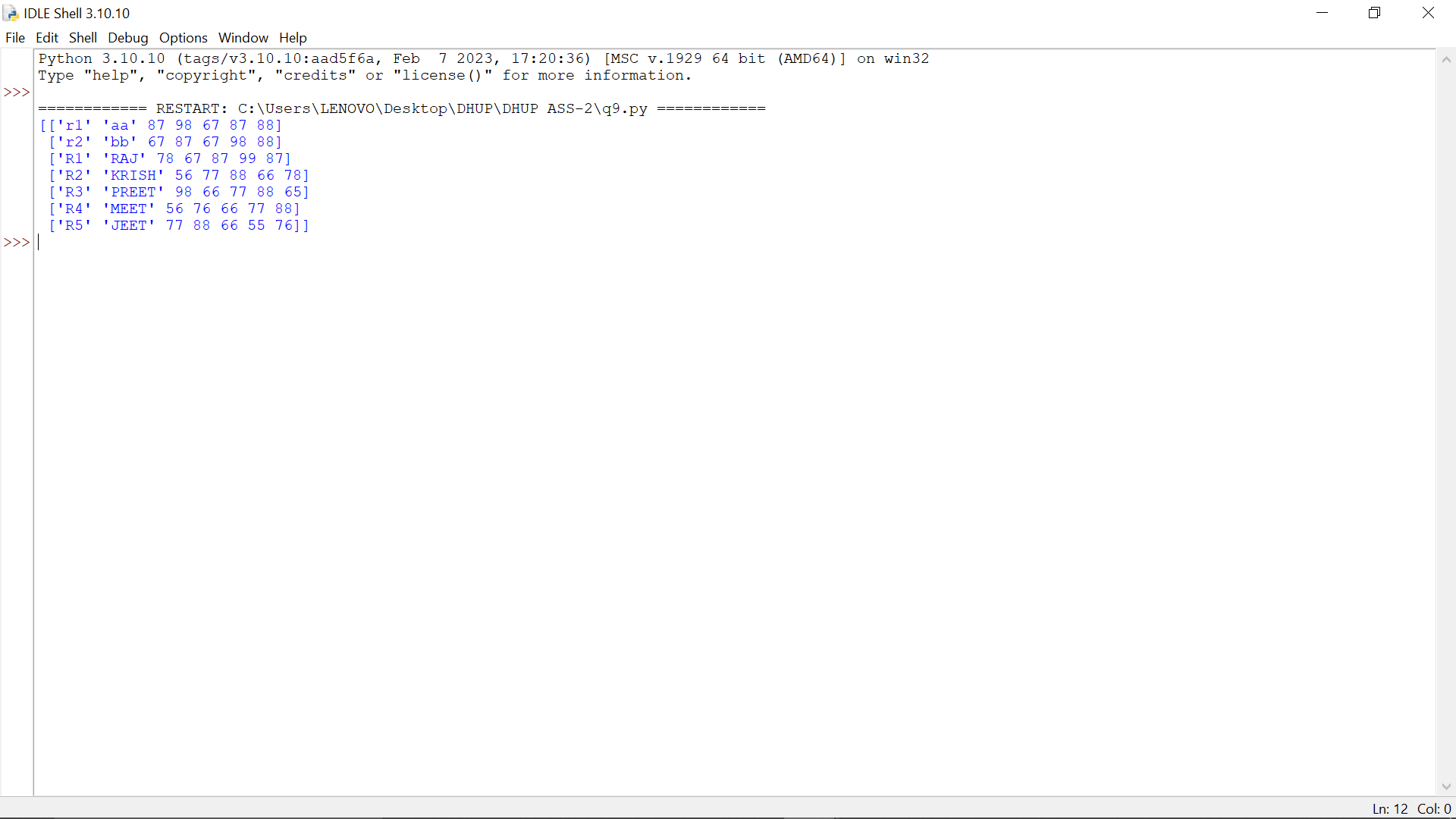
print(arr)

except Exception as e:

print("Error converting to NumPy:", e)

finally:

conn.close()

****

**10. Display all details of student who get more than 85 marks in Python.**

import sqlite3

import pandas as pd

try:

conn = sqlite3.connect("Student\_Information.db")

df = pd.read\_sql\_query("SELECT \* FROM Student", conn)

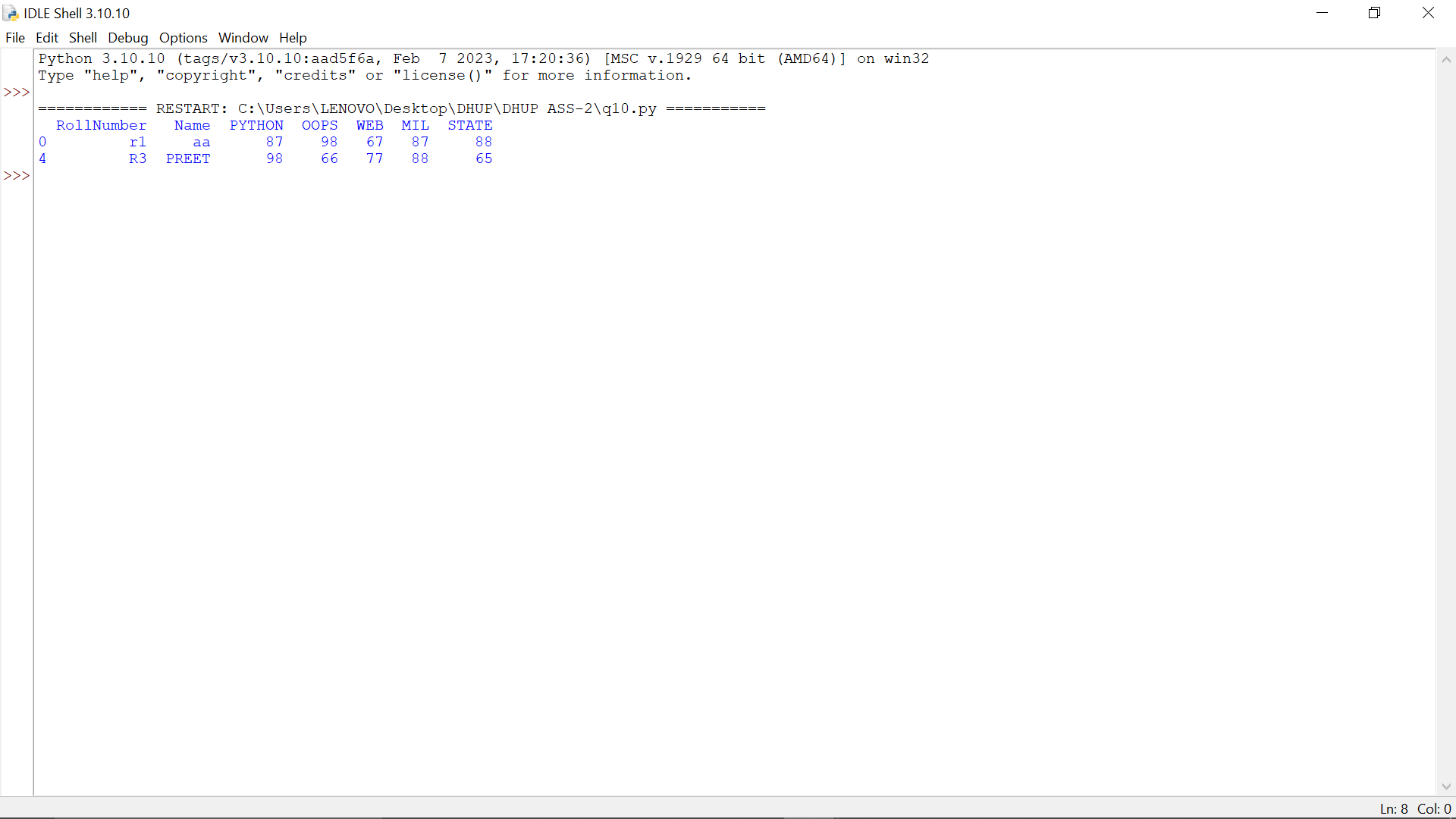
print(df[df['PYTHON'] > 85])

except Exception as e:

print("Error filtering data:", e)

finally:

conn.close()

****

**11. Write a python code to display all the students who get less than 36 Marks in any subject.**

import sqlite3

import pandas as pd

try:

conn = sqlite3.connect("Student\_Information.db")

df = pd.read\_sql\_query("SELECT \* FROM Student", conn)

low = df[(df['PYTHON'] < 36) | (df['OOPS'] < 36) | (df['WEB'] < 36) | (df['MIL'] < 36) | (df['STATE'] < 36)]

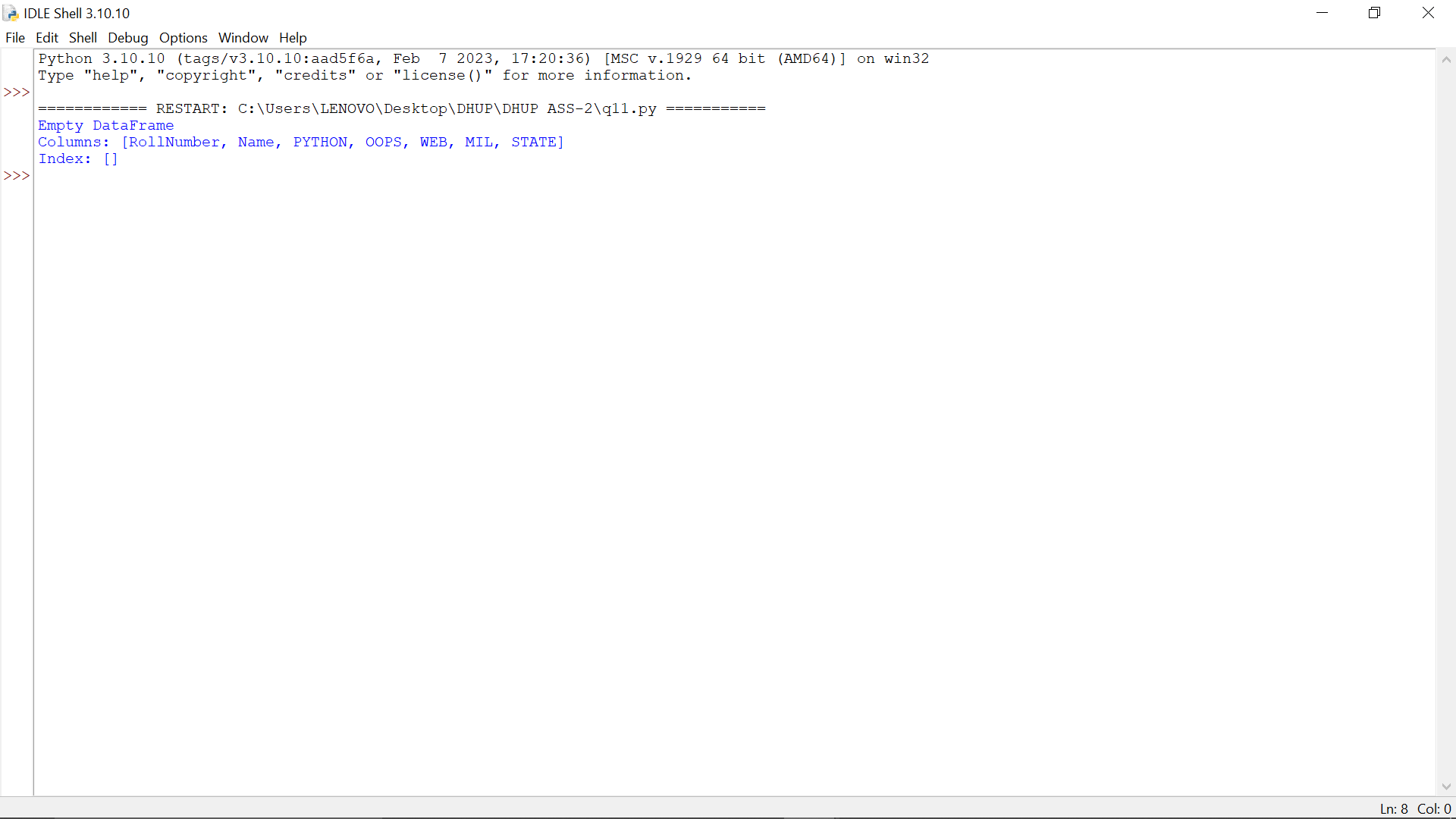
print(low)

except Exception as e:

print("Error filtering data:", e)

finally:

conn.close()

****

**12. Write a python code to plot all the marks of students in line chart. Use appropriate legend to describe the lines.**

import sqlite3

import pandas as pd

import matplotlib.pyplot as plt

try:

conn = sqlite3.connect("Student\_Information.db")

df = pd.read\_sql\_query("SELECT \* FROM Student", conn)

df.plot(x='Name', y=['PYTHON','OOPS','WEB','MIL','STATE'], kind='line')

print("Showing a Chart")

plt.show()

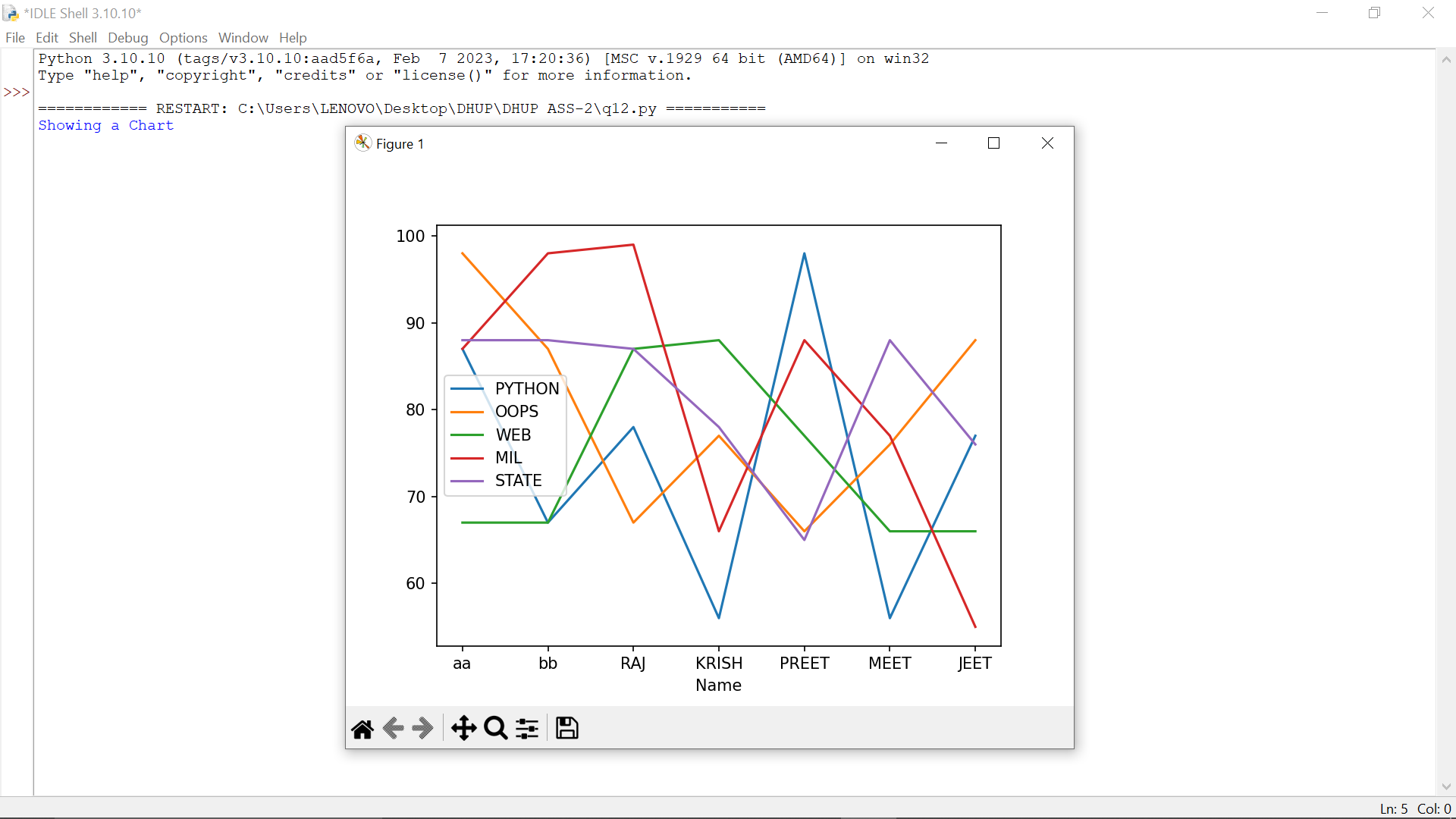
print("Chart Created Successfuly")

except Exception as e:

print("Error plotting chart:", e)

finally:

conn.close()

****

**13. Create a bar chart of the student’s Total marks.**

import sqlite3

import pandas as pd

import matplotlib.pyplot as plt

try:

conn = sqlite3.connect("Student\_Information.db")

df = pd.read\_sql\_query("SELECT \* FROM Student", conn)

df['Total'] = df[['PYTHON','OOPS','WEB','MIL','STATE']].sum(axis=1)

df.plot(x='Name', y='Total', kind='bar', color='orange')

print("Showing A Chart")

plt.show()

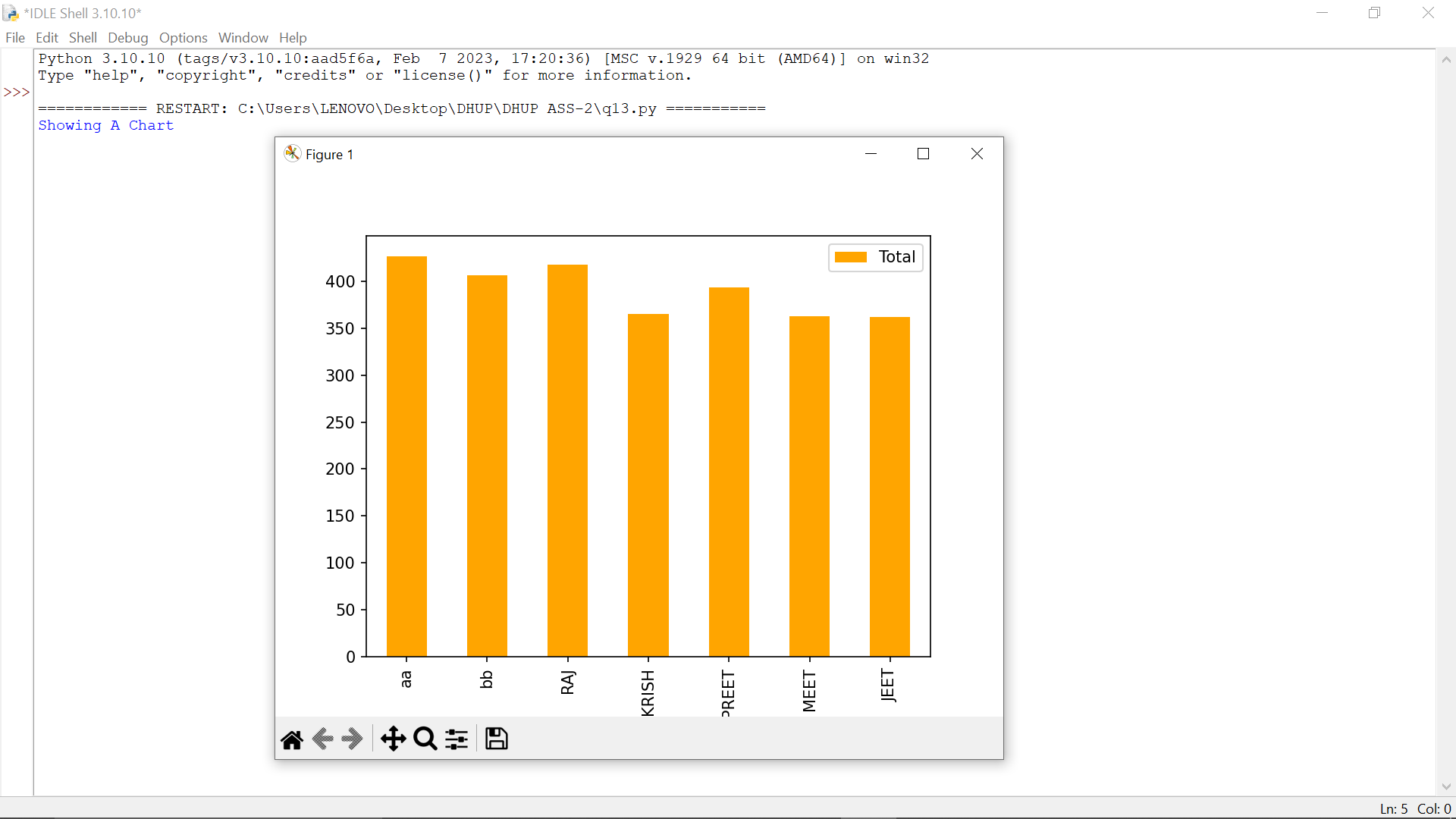
print("Chart Created Successfuly")

except Exception as e:

print("Error plotting bar chart:", e)

finally:

conn.close()

****

**14. Create pie charts for the PYTHON subject.**

import sqlite3

import pandas as pd

import matplotlib.pyplot as plt

try:

conn = sqlite3.connect("Student\_Information.db")

df = pd.read\_sql\_query("SELECT \* FROM Student", conn)

plt.pie(df['PYTHON'], labels=df['Name'], autopct='%1.1f%%')

print("Showing A Chart")

plt.show()

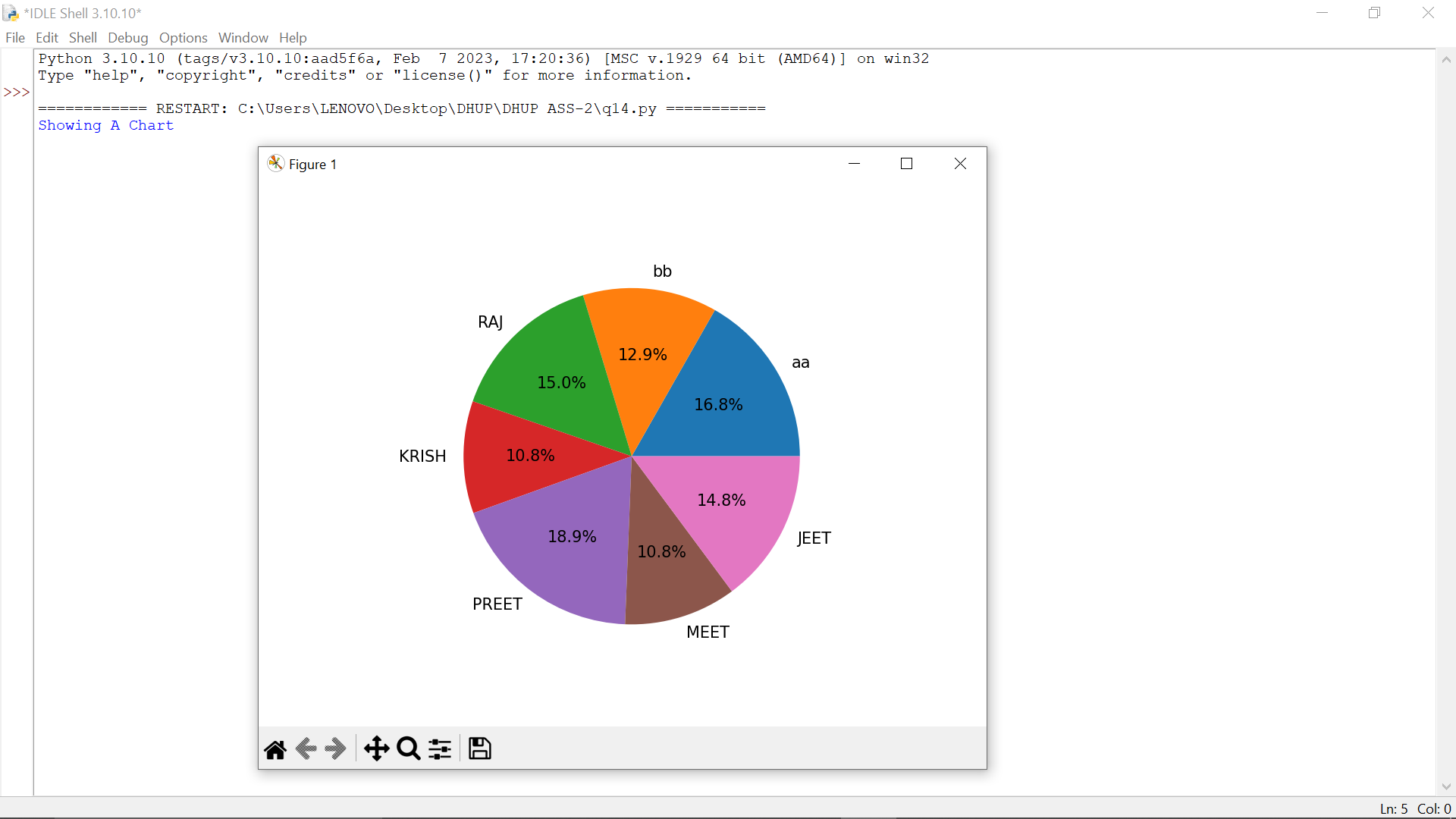
print("Chart Created Successfuly")

except Exception as e:

print("Error plotting pie chart:", e)

finally:

conn.close()

****

**15. Create line chart for the OOPS marks.**

import sqlite3

import pandas as pd

import matplotlib.pyplot as plt

try:

conn = sqlite3.connect("Student\_Information.db")

df = pd.read\_sql\_query("SELECT \* FROM Student", conn)

df.plot(x='Name', y='OOPS', kind='line', marker='o')

print("Showing A Chaert")

plt.show()

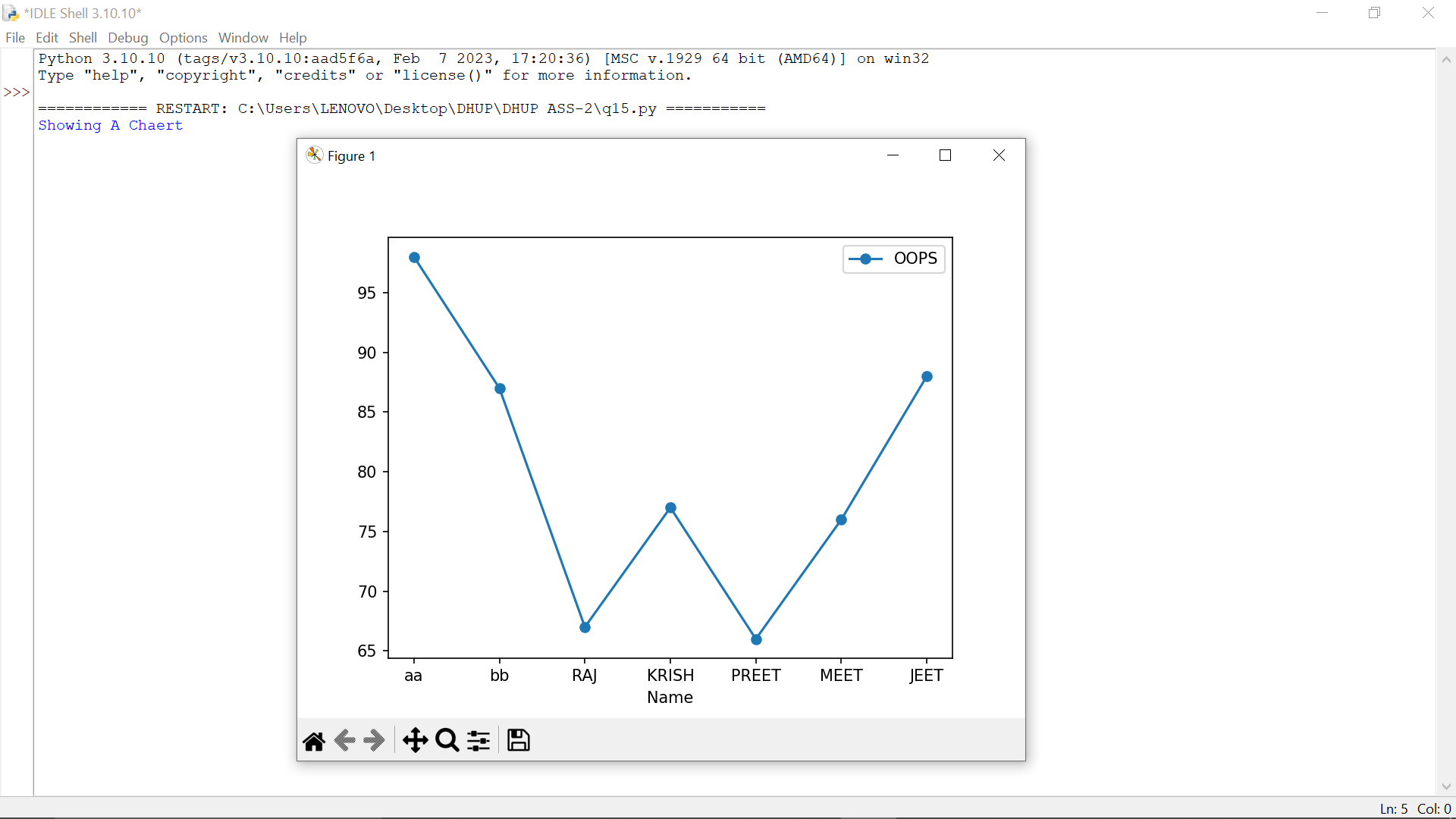
print("Chart Created Successfuly")

except Exception as e:

print("Error plotting OOPS chart:", e)

finally:

conn.close()

****

**16. Create a scatter plot chart for the web marks.**

import sqlite3

import pandas as pd

import matplotlib.pyplot as plt

try:

conn = sqlite3.connect("Student\_Information.db")

df = pd.read\_sql\_query("SELECT \* FROM Student", conn)

plt.scatter(df['Name'], df['WEB'], color='red')

print("Showing A Chart")

plt.show()

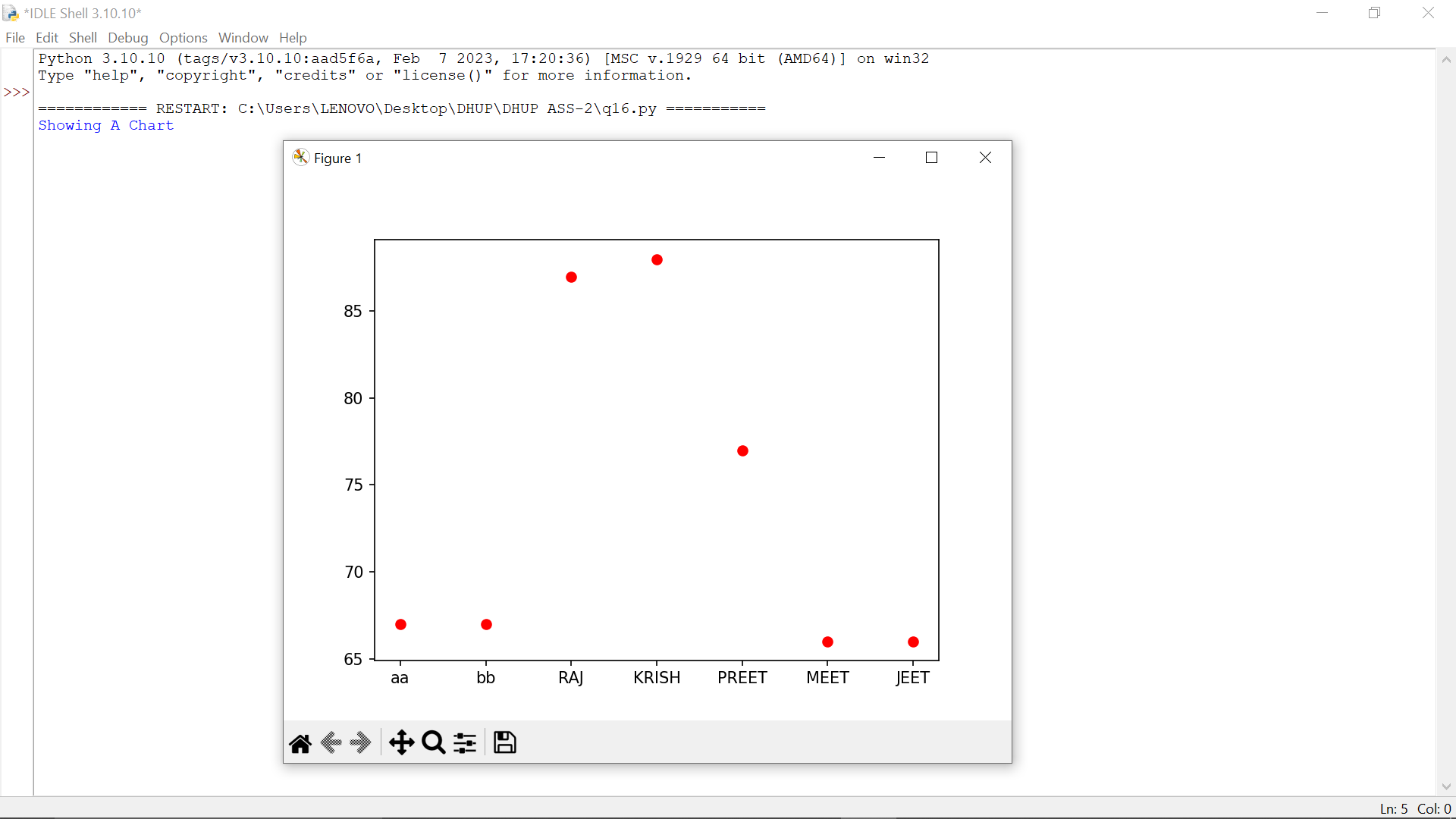
print("Chart Created Successfuly")

except Exception as e:

print("Error plotting scatter:", e)

finally:

conn.close()

****

**17. Find highest marks in each subject and make a bar chart of the same.**

import sqlite3

import pandas as pd

import matplotlib.pyplot as plt

try:

conn = sqlite3.connect("Student\_Information.db")

df = pd.read\_sql\_query("SELECT \* FROM Student", conn)

subjects = ['PYTHON','OOPS','WEB','MIL','STATE']

max\_marks = [df[s].max() for s in subjects]

plt.bar(subjects, max\_marks, color='green')

print("Showing A Chart")

plt.show()

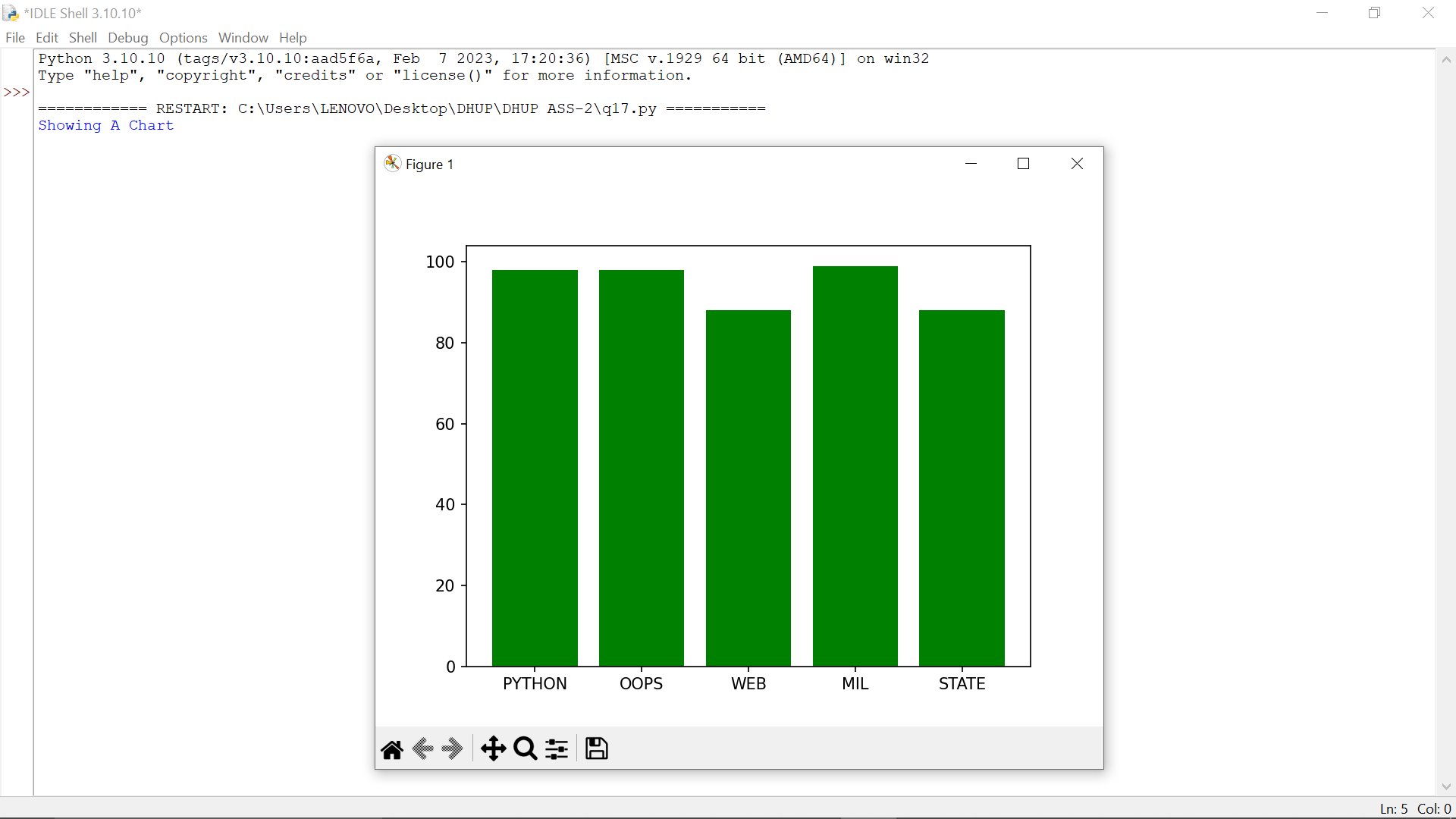
print("Chart Created Successfuly")

except Exception as e:

print("Error plotting highest marks:", e)

finally:

conn.close()

****

**18. Create a pie chart of the student’s Total marks and save this marks in a file named as “result.png”**

import sqlite3

import pandas as pd

import matplotlib.pyplot as plt

try:

conn = sqlite3.connect("Student\_Information.db")

df = pd.read\_sql\_query("SELECT \* FROM Student", conn)

df['Total'] = df[['PYTHON','OOPS','WEB','MIL','STATE']].sum(axis=1)

plt.pie(df['Total'], labels=df['Name'], autopct='%1.1f%%')

plt.savefig("result.png")

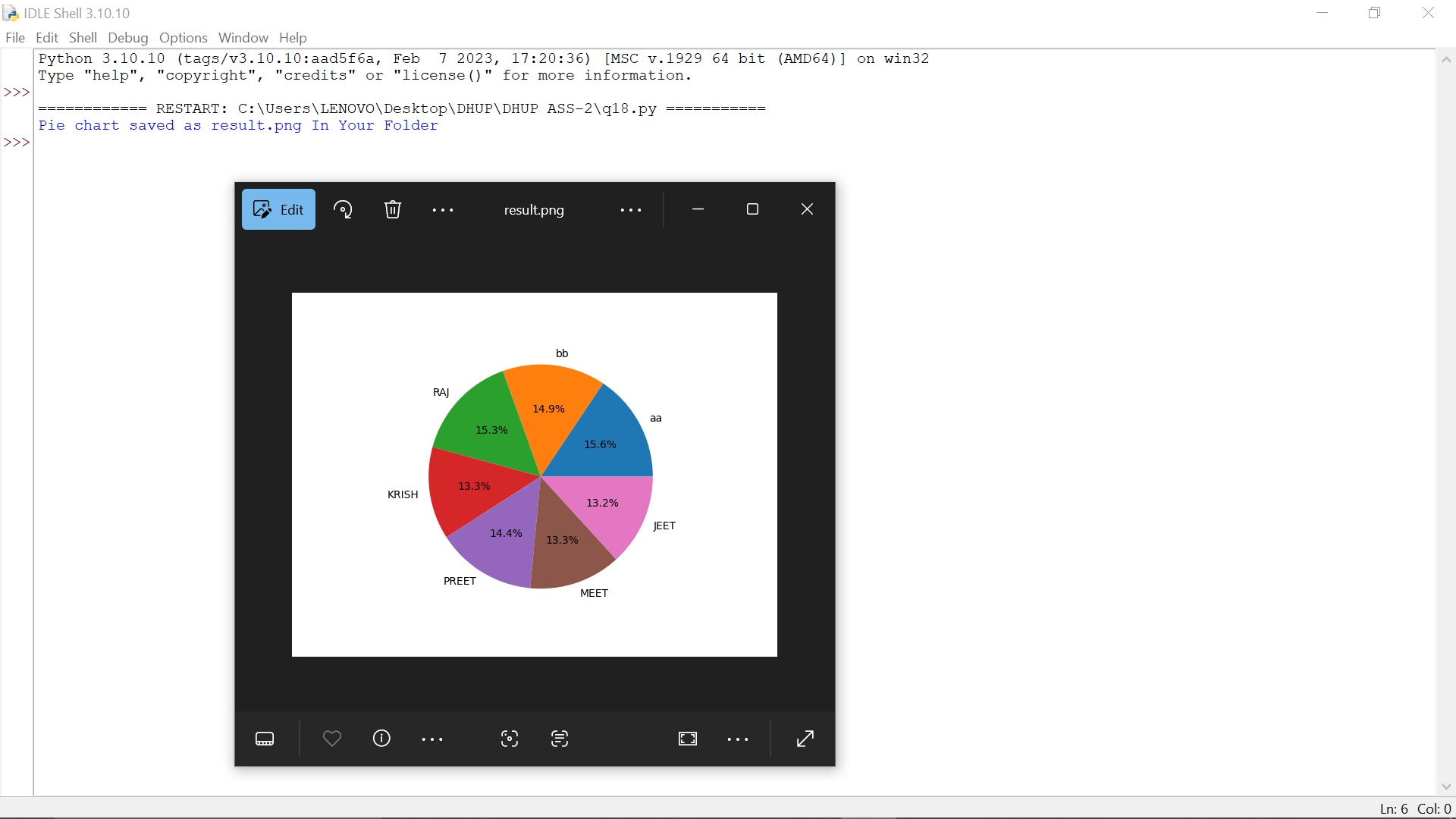
print("Pie chart saved as result.png In Your Folder")

except Exception as e:

print("Error saving pie chart:", e)

finally:

conn.close()

****