

Project Report
On
“PYTHON PROGRAMMING LAB”
Submitted in partial fulfillment of the requirement for the award of degree of
Bachelor of Technology
in
COMPUTER SCIENCE & ENGINEERING
Batch (2019-2023)



Subject - PYTHON PROGRAMMING LAB (ACCS-16409)

Submitted to:

Er. Ajay Sharma
(Associate Professor)

Submitted by:

Dipesh Paudel	(1900149)
Lovepreet Singh	(1900180)
Manish Kumar	(1900184)
Digambar Thakur	(1900147)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
Amritsar College of Engineering & Technology,
Amritsar
(Autonomous college under UGC Act – 1956[2(f) and 12(B)])

ACKNOWLEDGEMENT

This is a humble effort to express our sincere gratitude towards those who have guided and helped us to complete this project.

A project is major milestone during the study period of a student. As such this project was a challenge to us and was an opportunity to prove our caliber. We are highly grateful and obliged to everyone for helping in problems being faced by us.

It would not have been possible to see through the undertaken project without the guidance of **Er. Ajay Sharma**. It was purely on the basis of his experience and knowledge that we are able to clear all the theoretical and technical hurdles during the development phases of this project work.

Last but not the least we are very thankful to our Head of Department **Er. Vinod Kumar Sharma** and all the members of CSE Department who gave us an opportunity to face real time problems while fulfilling need of an organization by making project for them.

DECLARATION

We hereby declare that this project is an authentic record of our own work carried out as requirements of project for the award of degree of B. Tech (CSE), **Amritsar College of Engineering and Technology, Amritsar**, under the guidance of **Er. Ajay Sharma**.

(Signature of Students)

Dipesh Paudel (1900149)

Lovepreet Singh (1900180)

Digambar Thakur (1900147)

Manish Kumar (1900184)

Certified that above information provided by us is true as per our knowledge and belief.

Teacher Coordinator

Er. Ajay Sharma
(Associate Professor)



INDEX

Sr. No.	Content	Page No.
1.	<u>Introduction To Python Programming</u>	1-7
2.	<u>Objective of the Project</u>	8
3.	<u>Problem 1 – Plotting with matplotlib</u>	13-21
4.	<u>Introduction To CSV</u>	22
5.	<u>Problem 2 - Case Study 1</u> <u>Student Record(4th CSE 1 file)</u>	19-25
6.	<u>Problem 3 - Case Study 2</u> <u>Movie Dataset</u>	26-33
7.	<u>Problem 4 – Automobile Record Dataset</u>	34-49
8.	<u>Problem 5 – GUI with Python-Oracle Connectivity</u>	
9.	<u>Screen Shots</u>	
10.	References	50

Introduction to Language Used

[Python](#) is a general purpose programming language that is often applied in scripting roles. It is also called as Interpreted language.



Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. Python is often described as a "batteries included" language due to its comprehensive [standard library](#).

The Origin of Python:

Python programming language was invented in Netherlands by [Guido Van Rossum](#), in early 90's and its implementation was started in December 1989. Guido Van Rossum is a big fan of „Monty Python's Flying Circus“, famous TV show in Netherlands of that time. So, he named the language after Monty Python.

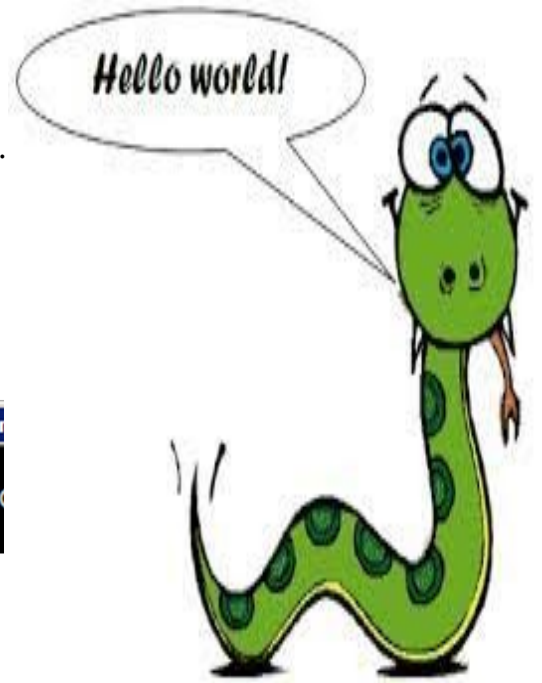
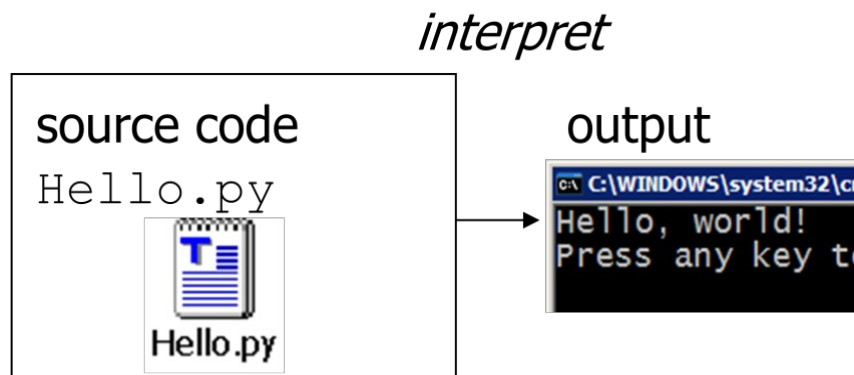


Features Of Python:

- It's Free: Downloading and installing Python is free and easy. Source code is easily accessible.
- Its Portable: Python runs virtually on every major platform used today. Programs runs exactly in the same manner irrespective of platform.
- It's Powerful : Dynamic typing , Built-in types and tools , library utilities , third-party utilities(e.g. Numpy , Scipy)
- It's Mixable : Integration of python with other languages is widely used .

- It's Relatively Easy to Use : Python Programs are compiled automatically to an intermediate form called bytecode , which the interpreter then reads.
- It's Relatively Easy to Learn
- Its Object – Oriented and Functional

Python is instead directly *interpreted* into machine instructions.



Applications of Python:

These are some real-world Python applications:

1. Web and Internet Development:

Python lets you develop a web application without too much trouble. It has libraries for internet protocols like HTML and XML, JSON, e-mail processing, FTP, IMAP, and easy-to-use socket interface. Yet, the package index has more libraries:

Requests – An HTTP client library

BeautifulSoup – An HTML parser

Feedparser – For parsing RSS/Atom feeds

Paramiko – For implementing the SSH2 protocol

Twisted Python – For asynchronous network programming

2. Desktop GUI Applications:

Most binary distributions of Python ship with Tk, a standard GUI library. It lets you draft a user interface for an application.

3. Science and Numeric Applications :

This is one of the very common applications of Python programming. With its power, it comes as no surprise that Python finds its place in the scientific community. For this, we have:

SciPy – A collection of packages for mathematics, science, and engineering.

Pandas– A data-analysis and -modeling library

IPython – A powerful shell for easy editing and recording of work sessions. It also supports visualizations and parallel computing.

Software Carpentry Course – It teaches basic skills for scientific computing and running bootcamps. It also provides open-access teaching materials.

Also, NumPy lets us deal with complex numerical calculations.

4. Software Development :

Software developers make use of Python as a support language. They use it for build-control and management, testing, and for a lot of other things:

SCons – for build-control

Buildbot, Apache Gump – for automated and continuous compilation and testing

Roundup, Trac – for project management and bug-tracking.

Roster of Integrated Development Environments

5. Education :

Thanks to its simplicity, brevity, and large community, Python makes for a great introductory programming language. Applications of Python programming in education has a huge scope as it is a great language to teach in schools or even learn on your own.

6. Business :

Python is also a great choice to develop ERP and e-commerce systems:

Tryton - A three-tier, high-level general-purpose application platform.

Odoo – A management software with a range of business applications. With that, it's an all-rounder and forms a complete suite of enterprise-management applications in-effect.

7. Database Access :

With Python, you have:

Custom and ODBC interfaces to MySQL, Oracle, PostgreSQL, MS SQL Server, and others. These are freely available for download.

Object databases like Durus and ZODB

Standard Database API

8. Network Programming :

With all those possibilities, how would Python slack in network programming? It does provide support for lower-level network programming:

Twisted Python – A framework for asynchronous network programming.

An easy-to-use **socket interface**

9. Games and 3D Graphics :

Safe to say, this one is the most interesting. When people hear someone say they're learning Python, the first thing they get asked is – „So, did you make a game yet?“

pygame, PyKyra are two frameworks for game development with Python. Apart from this, we also get a variety of 3D-rendering libraries. If you're one of those game-developers, you can check out PyWeek, a semi-annual game programming contest.

10. Artificial Intelligence and Machine Learning:

The IT industry went far beyond the science fiction ideas and our future is moving towards Artificial Intelligence. People are keen to move to technologies where machines can mimic human behaviours and most of the human work is being automated by machines. There is also a need to process and analyze huge volumes of data.

Statistically speaking that Python is the most popular language for Artificial intelligence and Machine Learning.

Other Python Applications :

There are some of the major Python Applications:

- Console-based Applications

- Audio or Video-based Applications

- Applications for Images

- Enterprise Applications

- 3D CAD Applications

- Computer Vision (Facilities like face-detection and color-detection)

- Machine Learning

- Robotics

- Web Scraping (Harvesting data from websites)

- Scripting

- Artificial Intelligence

- Data Analysis (The Hottest of Python Applications)

Disadvantages of Python:

Speed :

Python is slower than C or C++. But of course, Python is a high-level language, unlike C or C++ it's not closer to hardware.

Mobile Development :

Python is not a very good language for mobile development . It is seen as a weak language for mobile computing. This is the reason very few mobile applications are built in it like Carbonnelle.

Memory Consumption :

Python is not a good choice for memory intensive tasks. Due to the flexibility of the data-types, Python's memory consumption is also high.

Database Access :

Python has limitations with database access . As compared to the popular technologies like JDBC and ODBC, the Python's database access layer is found to be bit underdeveloped and primitive . However, it cannot be applied in the enterprises that need smooth interaction of complex legacy data .

Runtime Errors :

Python programmers cited several issues with the design of the language. Because the language is dynamically typed , it requires more testing and has errors that only show up at runtime .

Keywords:

Python has the following keywords or reserved words; they cannot be used as identifiers.

- and
- as
- assert
- async

- await
- break
- class
- continue
- def
- del
- elif
- else
- except
- False
- finally
- for
- from
- global
- if
- import
- in
- is
- lambda
- None
- nonlocal
- not
- or
- pass
- raise
- return
- True
- try
- while
- with
- yield

Objective of the Project

The basic objective of this project is to get basic knowledge of Python. Python programming is intended for software engineers, system analysts, program managers and user support personnel who wish to learn the Python programming language.

The main objective of this project is to get a good knowledge about python libraries and have good idea about the various Data Visualizations and CSV modules available in python which helps us to generate and solve our queries by visualizing it in an interactive way as well as manipulating the data through comma separated values CSV module in tabular manner.

We have used matplotlib module which is a comprehensive library for creating static, animated, and interactive visualizations in Python.

We have also casted the so-called CSV (Comma Separated Values) format which is the most common import and export format for spreadsheets and databases. Programmers can also describe the CSV formats understood by other applications or define their own special-purpose CSV formats.

Problem 1- Plotting with matplotlib

Q1. Explain the matplotlib module in python. Write in detail about the various plots available in matplotlib. Write programs to explain every type of plot. Choose some dataset of your choice.

Introduction to Matplotlib:

Matplotlib is the brainchild of John Hunter (1968-2012), who, along with its many contributors, have put an immeasurable amount of time and effort into producing a piece of software utilized by thousands of scientists worldwide.

The [matplotlib.pyplot](#) is a collection of functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc.

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in python.

Matplotlib is a plotting library in Python, which produces 2D graphics in different interactive environments. You can draw different types of graphs like line, bar, scatter etc.

```
import matplotlib.pyplot as plt
```

matplotlib.pyplot is a collection of command style functions that make matplotlib work .

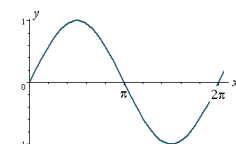
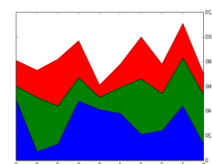
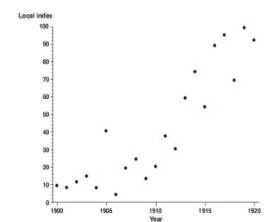
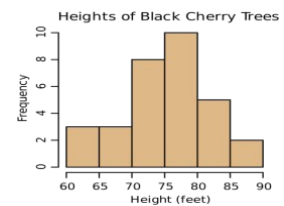
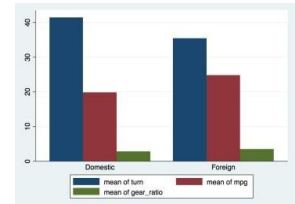
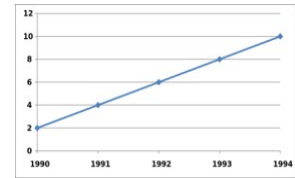
Matplotlib is the whole package,pylab is a module in matplotlib that gets installed alongside matplotlib and matplotlib.pyplot is a module in matplotlib. Difference between pylab and pyplot.

pyplot interface is generally preferred for non - interactive plotting. Pyplot provides the state - machine interface to the underlying plotting library in matplotlib. Pylab combines the pyplot functionality with the NumPy functionality in a single namespace. The pylab interface is convenient for interactive calculations and plotting.

Types of Plots are:-

Some basic types of plots available in matplotlib module are discussed below:

1. **Line Plot:** This is the basic simple graph of matplotlib. This is used to visualize some common things quickly.
2. **Bar Plot:** A bar graph uses bars to compare data among different categories. It is well suited when you want to measure the changes over a period of time. It can be represented horizontally or vertically. Also, the important thing to keep in mind is that longer the bar, greater is the value.
3. **Histogram:** The histogram is used where the data is been distributed while bar graph is used in comparing the two entities. Histograms are preferred during the arrays or data containing the long list.
4. **Pie Chart:** A pie chart refers to a circular graph which is broken down into segments i.e. slices of pie. It is basically used to show the percentage or proportional data where each slice of pie represents a category.
5. **Scatter Plot:** The scatter plots are preferred while comparing the data variables to determine the relationship between dependent and independent variables. The data is displayed as a collection of points, each having the value of one variable which determines the position on the horizontal axis and the value of other variable shows on vertical axis.
6. **Area Plot:** The area plots were also called as stack plots. It is quite similar to the line plots. Area plots are used in tracking the changes over time for two or more related groups that make one whole category.
7. **Curves:** The curves are basically used for showing the trigonometric (e.g. sine, cosine) waves to the user.



Data analysis :-It is defined as a process of cleaning, transforming, and modeling data to discover useful information for business decision-making. The purpose of Data Analysis is to extract useful information from data and taking the decision based upon the data analysis. Whenever we take any decision in our day-to-day life is by thinking about what happened last time or what will happen by choosing that particular decision. This is nothing but analyzing our past or future and making decisions based on it. For that, we gather memories of our past or dreams of our future. So that is nothing but data analysis. Now same thing analyst does for business purposes, is called DataAnalysis.

CSV:- The [csv](#) module implements classes to read and write tabular data in CSV format. It allows programmers to say, “write this data in the format preferred by Excel,” or “read data from this file which was generated by Excel,” without knowing the precise details of the CSV format used by Excel. Programmers can also describe the CSV formats understood by other applications or define their own special-purpose CSV formats.

Some of the areas where they have been used include:

- importing and exporting customer data
- importing and exporting products
- exporting orders
- exporting e-commerce analytic reports

Reader and Writer Modules

The CSV module has several functions and classes available for reading and writing CSVs, and they include:

- csv.readerfunction
- csv.writerfunction
- csv.Dictwriterclass
- csv.DictReaderclass

Problem 1

Case Study 1: Create a CSV file of your class CSE-IV-1 Sem and store the following details in it: URollNo, Name, DOB (dd-mon-yyyy), City, State/Country, and Marks (CGPA). Develop a menu driven application in python which answers the following queries:

(a) List all the students

- (b) Display all the students from a particular state/country
- (c) Display all students from a particular state and born in a particular month
- (d) Display all students with a given CGPA
- (e) Display all students whose name starts with a particular letter and a particular year of birth

```
Python Project > AllData.py > ...
1 #Python Project Program to List all the Student of CSE-1
2
3 import csv
4 f=open("dipesh.csv","r") #opening a file
5 head=[] #create a list to pickup header information of header row
6 rows=[] #List to pickup all the rows
7 csvr=csv.reader(f)
8 head=next(csvr, None)
9 print(head)
10 for i in csvr:
11     rows.append(i)
12     print(i)
13 print("Total no. of rows: %d"%(csvr.line_num))
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE 2: Python

```
['Arpit punj', '1900133', '2002/01/28', 'Amritsar', 'Punjab', '8.2']
['Karanbir Singh', '1900174', '1999/09/15', 'Amritsar', 'Punjab', '7.8']
['Sparsh Mehta', '1900277', '2001/11/25', 'Amritsar', 'Punjab', '8.3']
['Angadjit Singh', '1900120', '2000/11/11', 'Amritsar', 'Punjab', '8.1']
['Meetal', '1900196', '2000/08/15', 'Amritsar', 'Punjab', '8.2']
['Manpreet Singh', '1900188', '2001/12/20', 'Amritsar', 'Punjab', '8.8']
['Parteetjot Singh', '1900215', '2001/02/09', 'Amritsar', 'Punjab', '8.3']
['Jasmeet', '1900169', '2001/04/29', 'Amritsar', 'Punjab', '8.4']
['Abhay Chugh', '1900102', '2001/12/28', 'Muktsar', 'Punjab', '8.5']
['Kabir Piplani', '1900173', '2000/12/12', 'Amritsar', 'Punjab', '7']
['Aman Gupta', '1900115', '2000/06/12', 'Ghaziabad', 'UP', '7.5']
['Lovepreet', '1900100', '2000/04/22', 'Jaipur', 'Bihar', '9']
Total no. of rows: 59
```

Ln 8, Col 22 Spaces: 4 UTF-8 LF Python

Output Of Problem 1:

Visual Studio Code May 29 11:50

AllData.py - py - Visual Studio Code

File View Go Run Terminal Help

... Restaurant.py AllData.py X

Python Project > AllData.py > ...

```
2
3 import csv
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE 2: Python

```
['Anup Kumar', '1900130', '2000/01/31', 'Bettiah', 'Bihar', '7']
['Archana Kumari', '1900132', '2002/01/05', 'Motihari', 'Bihar', '6.7']
['Charandeep Singh', '1900144', '2000/09/15', 'Amritsar', 'Punjab', '7.9']
['Nitika Neb', '1900208', '2001/03/14', 'Amritsar', 'Punjab', '8.3']
['Palak Mahajan', '1900209', '2001/01/16', 'Amritsar', 'Punjab', '8.5']
['Gurkirat Singh', '1900156', '2001/03/15', 'Amritsar', 'Punjab', '7.8']
['Aditya Nath Jha', '1900109', '2000/01/02', 'Darbhanga', 'Bihar', '6.7']
['Geetika Kapoor', '1900154', '2000/09/10', 'Amritsar', 'Bihar', '8.5']
['Anmol Chawla', '1900126', '2001/02/15', 'Batala', 'Punjab', '8']
['Amit Vijay', '1900117', '2000/12/07', 'Bettiah', 'Bihar', '7']
['Radhika Mittal', '1900228', '2000/09/01', 'Amritsar', 'Punjab', '8.5']
['Inderjit Singh', '1900162', '2000/11/18', 'Amritsar', 'Punjab', '8.5']
['Nimisha Kumari', '1900204', '2001/03/18', 'Nawada', 'Bihar', '8.3']
['Priya Mehta', '1900225', '2000/10/24', 'Amritsar', 'Punjab', '8.2']
['Simran Bhogal', '1900272', '2001/11/10', 'Dhanbad', 'Jharkhand', '8.4']
['Anisha Mahajan', '1900121', '2001/03/07', 'Amritsar', 'Punjab', '8.5']
['Arpit punj', '1900133', '2002/01/28', 'Amritsar', 'Punjab', '8.2']
['Karanbir Singh', '1900174', '1999/09/15', 'Amritsar', 'Punjab', '7.8']
['Sparsh Mehta', '1900277', '2001/11/25', 'Amritsar', 'Punjab', '8.3']
['Angadjit Singh', '1900120', '2000/11/11', 'Amritsar', 'Punjab', '8.1']
['Meetali', '1900196', '2000/08/15', 'Amritsar', 'Punjab', '8.2']
['Manpreet Singh', '1900188', '2001/12/20', 'Amritsar', 'Punjab', '8.8']
['Parteetjot Singh', '1900215', '2001/02/09', 'Amritsar', 'Punjab', '8.3']
['Jasmeet', '1900169', '2001/04/29', 'Amritsar', 'Punjab', '8.4']
['Abhay Chugh', '1900102', '2001/12/28', 'Muktsar', 'Punjab', '8.5']
['Kabir Piplani', '1900173', '2000/12/12', 'Amritsar', 'Punjab', '7']
['Aman Gupta', '1900115', '2000/06/12', 'Ghaziabad', 'UP', '7.5']
['Lovepreet', '1900100', '2000/04/22', 'Jaipur', 'Bihar', '9']
Total no. of rows: 59
```

Ln 8, Col 22 Spaces: 4 UTF-8

Problem !

Case Study 1: Create a CSV file of your class CSE-IV-1 Sem and store the following details in it: URollNo, Name, DOB (dd-mon-yyyy), City, State/Country, and Marks (CGPA). Develop a menu driven application in python which answers the following queries:

(a) List all the students

(b) Display all the students from a particular state/country

(c) Display all students from a particular state and born in a particular month

(d) Display all students with a given CGPA

(e) Display all students whose name starts with a particular letter and a particular year of birth

The screenshot shows the Visual Studio Code interface. The top bar indicates the file is 'P_StateCountry.py - py - Visual Studio Code' and the date is 'May 29 11:52'. The menu bar includes 'File', 'Edit', 'View', 'Go', 'Run', 'Terminal', and 'Help'. The Explorer sidebar on the left shows a project structure with files like 'Restaurant.py', 'P_StateCountry.py', and others. The main editor area displays the following Python code:

```
1 #Python Project Program to Display all the students from a particular state/country
2
3 import csv
4 f=open("dipesh.csv","r")
5 head=[]
6 rows=[]
7 csvr=csv.reader(f)
8 head=next(csvr, None)
9 print(head)
10 for i in csvr:
11     rows.append(i)
12 count=0
13 for i in rows:
14     if i[4]=="Nepal":
15         print(i[0])
16         count+=1
17
18 print("Total no. of rows:",count)
```

The bottom panel shows the 'TERMINAL' tab with the command prompt 'e4xcaptain@e4xcaptain-TravelMate-P2510-G2-MG:~/Desktop/py\$'.

P_StateCountry.py - py - Visual Studio Code

Go Run Terminal Help

... Restaurant.py P_StateCountry.py X

Python Project > P_StateCountry.py > ...

```
1 #Python Project Program to Display all the students from a particular state/coun
2
3 import csv
4 f=open("dipesh.csv","r")
5 head=[]
6 rows=[]
7 csvr=csv.reader(f)
8 head=next(csvr, None)
9 print(head)
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

2: Python

```
e4xcaptain@e4xcaptain-TravelMate-P2510-G2-MG:~/Desktop/py$ /usr/bin/python3 "/home/e4xcaptain/Desktop/py/
Country.py"
['Name', 'URollNo', 'DOB', 'City', 'State', 'Marks(CGPA)']
Dipesh Paudel
Aftab Ansari
Md.Mehboob Raza
Aasha Pun
Total no. of rows: 4
e4xcaptain@e4xcaptain-TravelMate-P2510-G2-MG:~/Desktop/py$
```

Ln 1, Col 1 Spaces:

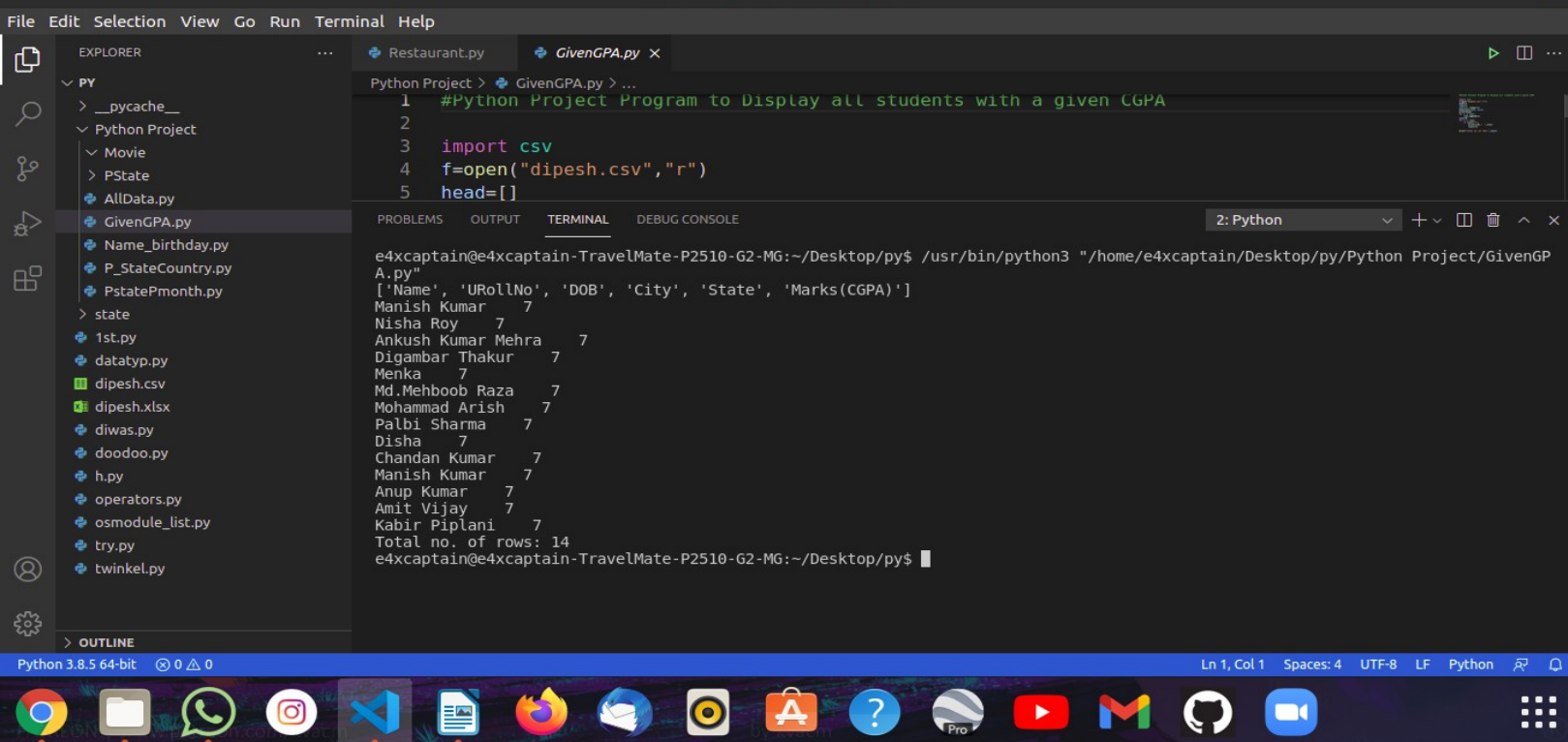
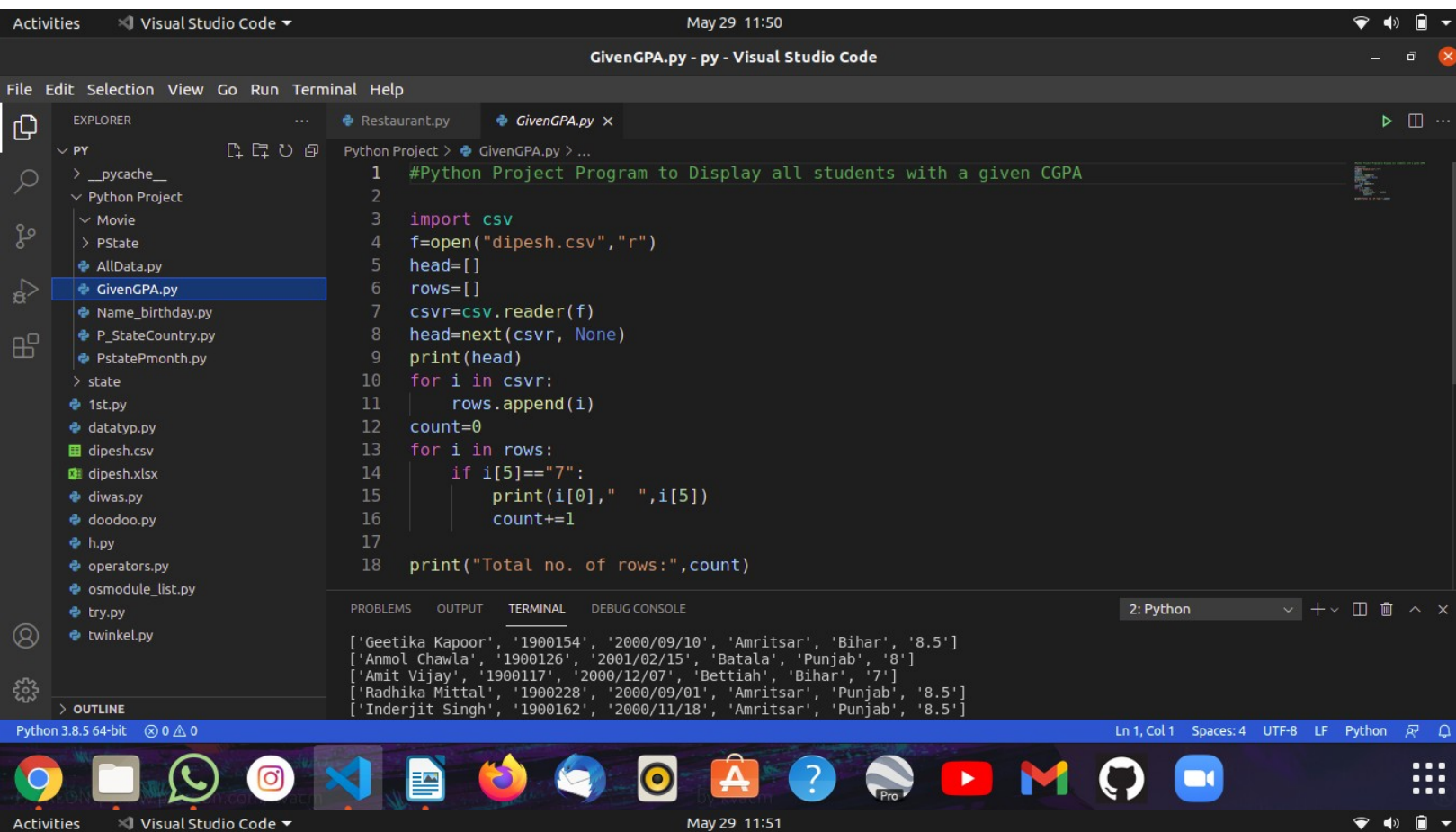


2: Python

... Restaurant.py PstatePmonth.py X

2: Python

Ln 14, Col 16 Spaces: 4 UTF-8



Activities Visual Studio Code May 29 11:51

Name_birthday.py - py - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

- Restaurant.py
- Name_birthday.py

Python Project > Name_birthday.py > ...

```
1 #Python Project Program to Display all students whose name starts with a particular
2 #letter and a particular year of birth'''
3
4 import csv
5 f=open("dipesh.csv","r")
6 head=[]
7 rows=[]
8 csvr=csv.reader(f)
9 head=next(csvr, None)
10 print(head)
11 for i in csvr:
12     rows.append(i)
13 count=0
14 for i in rows:
15     b=str(i[2])
16     c=b[0:4]
17     d=i[0]
18     e=d[0]
19     if e=="A" and c=="2000":
20         print(i[0])
21     count+=1
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

2: Python

Python 3.8.5 64-bit 0 0 0

Ln 9, Col 22 Spaces: 4 UTF-8 LF Python

Activities Visual Studio Code May 29 11:51

Name_birthday.py - py - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

- Restaurant.py
- Name_birthday.py

Python Project > Name_birthday.py > ...

```
1 #Python Project Program to Display all students whose name starts with a particular
2 #letter and a particular year of birth'''
3
4 import csv
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

2: Python

e4xcaptain@e4xcaptain-TravelMate-P2510-G2-MG:~/Desktop/py\$ /usr/bin/python3 "/home/e4xcaptain/Desktop/py/Python Proj
rthday.py"
['Name', 'URollNo', 'DOB', 'City', 'State', 'Marks(CGPA)']
Anchal Sharma
Aftab Ansari
Aasha Pun
Abhishek Poddar
Anup Kumar
Aditya Nath Jha
Amit Vijay
Angadjiit Singh
Aman Gupta
Total no. of rows: 9
e4xcaptain@e4xcaptain-TravelMate-P2510-G2-MG:~/Desktop/py\$

Python 3.8.5 64-bit 0 0 0

Ln 9, Col 22 Spaces: 4 UTF-8 LF

```

ent1=input('Enter the State: ').upper()
ent2=input('Enter the Country: ').upper()
print()

f=open("p1_4cse1_project_dataset_2.csv","r")
csvreader=csv.reader(f)
heading =next(csvreader)
print(heading)
count=0
for i in csvreader:
    if i[4]==ent1 and i[5]==ent2:
        count+=1
        print(i)
if (count ==0):
    print('Sorry Data not Found')
else:
    print('Total no. of records came out =',count)
f.close()
input()

```

```

def query3():    #student from particular state and particular year of birth
    system('cls')
    print('Students from a particular state and born in a particular month\n')

```

```

ent1=input('Enter the State: ').upper()
ent2=input('Enter the Month (in number): ')

print()
f=open("p1_4cse1_project_dataset_2.csv","r")
csvreader=csv.reader(f)
heading =next(csvreader)
print(heading)
count=0
for i in csvreader:
    mon=i[2].split('/')[1]
    if i[4]==ent1 and mon==ent2:
        count+=1
        print(i)
if (count ==0):
    print('Sorry Data not Found')
else:
    print('Total no. of records came out =',count)
f.close()
input()

```

```

def query4():    #student from particular cgpa range
    system('cls')
    print('Students with a given CGPA\n')

```

```

ent1=float(input("Enter the CGPA: "))
print()
f=open("p1_4cse1_project_dataset_2.csv","r")
csvreader=csv.reader(f)
heading =next(csvreader)
print(heading)
count=0
for i in csvreader:
    if float(i[6])==ent1:
        count+=1
        print(i)
if (count ==0):
    print('Sorry Data not Found')
else:
    print('Total no. of records came out =',count)
f.close()
input()

def query5():    #student name startswith particular letter and from particular year of birth
    system('cls')
    print('Student whose name starts with a particular letter and a particular year of birth:
\n')
    ent1=input("Enter the first letter of Student's name : ").upper()
    ent2=input("Enter Year of birth (in yyyy) : ")
    print()

    f=open("p1_4cse1_project_dataset_2.csv","r")
    csvreader=csv.reader(f)
    heading =next(csvreader)
    print(heading)
    count=0
    for i in csvreader:
        if i[1].startswith(ent1) and i[2][-4::]==ent2:
            count+=1
            print(i)
    if (count ==0):
        print('Sorry Data not Found')
    else:
        print('Total no. of records came out =',count)
    f.close()
    input()

while(True):
    system('cls')
    print(colorama.Fore.LIGHTCYAN_EX , "*****CSE-IV-1*****\n")
    print('Queries :')
    print("1.List all the students")
    print("2.Display all the students from particular state/country")

```



```

    print("3.Display all the students from a particular state and born in a particular month")
    print("4.Display all the students with given CGPA")
    print("5.Display all the students whose name start with a particular letter and a particular yearof birth")
    print("press 0 for Exit")

    try:
        ch=int(input('\nEnter Your Choice : '))
        if ch==1:
            query1()
        elif ch==2:
            query2()
        elif ch==3:
            query3()
        elif ch==4:
            query4()
        elif ch==5:
            query5()
        elif ch==0:
            print('BYE!!!')
            exit()
    except ValueError:
        print('Error!! Incorrect value entered')
        input()

```

Output Snapshots of Problem 2:

```
*****CSE-IV-1*****
```

Queries :

- 1.List all the students
 - 2.Display all the students from particular state/country
 - 3.Display all the students from a particular state and born in a particular month
 - 4.Display all the students with given CGPA
 - 5.Display all the students whose name start with a particular letter and a particular yearof birth
- press 0 for Exit

Enter Your Choice : █

***** List Of All Students *****

['U.ROLLNO',	'FULL NAME',	'MM/DD/YYYY',	'CITY',	'STATE',	'COUNTRY',	'MARKS']
1900183	MANINDER SINGH	21/11/2000	TARN TARAN	PUNJAB	INDIA	8.04
1900184	KABIR PIPLANI	25/8/1999	AMRITSAR	PUNJAB	INDIA	8.54
1900185	ANISHA MAHAJAN	19/2/2001	JALANDHAR	PUNJAB	INDIA	9
1900186	RADHIKA MITTAL	5/4/2001	AMRITSAR	PUNJAB	INDIA	9.1
1900187	ANCHAL SHARMA	1/1/2000	KATHMANDU	BAGMATI	NEPAL	7.96
1900188	MANPREET SINGH	20/12/2001	AMRITSAR	PUNJAB	INDIA	8.8
1900189	MANPREET SINGH	21/1/1998	TARN TARAN	PUNJAB	INDIA	6.6
1900190	RUBEENA	11/9/2001	BATHINDA	PUNJAB	INDIA	8.4
1900191	MANMEET KAUR	25/12/2002	KATHMANDU	BAGMATI	NEPAL	8.9
1900192	AMAN	15/12/2000	GUJRAT	GUJRAT	INDIA	7.8
1900193	ARSHPREET KAUR	25/5/1997	JALANDHAR	PUNJAB	INDIA	8.6
1900194	GEETIKA KAPOOR	5/4/1996	MUMBAI	MUMBAI	INDIA	6.6
1900195	RABIA	2/10/2003	TARN TARAN	PUNJAB	INDIA	6.8
1900196	MEHBUB RAJA	30/4/1995	KATHMANDU	BAGMATI	NEPAL	7.4
1900197	ABHISHEK PIPLANI	16/11/2002	AMRITSAR	PUNJAB	INDIA	8.2
1900198	SHIFALI	18/11/1996	KANPUR	U.P	INDIA	9.9
1900199	AKASH	10/2/2001	KATHMANDU	BAGMATI	NEPAL	9.5
1900200	SAVLIN	3/1/1999	GURDASPUR	PUNJAB	INDIA	8.2
1900201	VATAYINA	16/10/2005	BEAS	PUNJAB	INDIA	6.9
1900202	ISHITA	11/12/1992	TARN TARAN	PUNJAB	INDIA	7.5
1900203	RUCHI	15/11/2000	RAJASTHAN	RAJASTHAN	INDIA	8.6
1900204	AJAY SHARMA	19/11/2002	KANPUR	U.P	INDIA	8.4
1900205	ANUSHTHA	15/8/2003	AMRITSAR	PUNJAB	INDIA	5.7
1900206	SANDEEP	5/11/1997	BATALA	PUNJAB	INDIA	7.9
1900207	AMIT	21/12/2004	JALANDHAR	PUNJAB	INDIA	5.5
1900208	PRABHNOOR	9/9/2001	TARN TARAN	PUNJAB	INDIA	4.4
1900209	MUSKAN	18/7/2006	BATALA	PUNJAB	INDIA	8.2
1900210	PAYAL	17/10/2000	AMRITSAR	PUNJAB	INDIA	7.2
1900211	TAYYABA	19/8/2008	KATHMANDU	BAGMATI	NEPAL	9.4
1900222	MEHAK	29/2/2020	KATHMANDU	BAGMATI	NEPAL	6.5

Total no. of records came out = 30

(b) Students from a particular state/country :

Enter the State : Punjab

Enter the Country : India

['U.ROLLNO', 'FULL NAME', 'MM/DD/YYYY', 'CITY', 'STATE', 'COUNTRY', 'MARKS']
1900183 MANINDER SINGH 21/11/2000 TARN TARAN PUNJAB INDIA 8.04
1900184 KABIR PIPLANI 25/8/1999 AMRITSAR PUNJAB INDIA 8.54
1900185 ANISHA MAHAJAN 19/2/2001 JALANDHAR PUNJAB INDIA 9
1900186 RADHIKA MITTAL 5/4/2001 AMRITSAR PUNJAB INDIA 9.1
1900188 MANPREET SINGH 20/12/2001 AMRITSAR PUNJAB INDIA 8.8
1900189 MANPREET SINGH 21/1/1998 TARN TARAN PUNJAB INDIA 6.6
1900190 RUBEENA 11/9/2001 BATHINDA PUNJAB INDIA 8.4
1900193 ARSHPREET KAUR 25/5/1997 JALANDHAR PUNJAB INDIA 8.6
1900195 RABIA 2/10/2003 TARN TARAN PUNJAB INDIA 6.8
1900197 ABHISHEK PIPLANI 16/11/2002 AMRITSAR PUNJAB INDIA 8.2
1900200 SAVLIN 3/1/1999 GURDASPUR PUNJAB INDIA 8.2
1900201 VATAYINA 16/10/2005 BEAS PUNJAB INDIA 6.9
1900202 ISHITA 11/12/1992 TARN TARAN PUNJAB INDIA 7.5
1900205 ANUSHTHA 15/8/2003 AMRITSAR PUNJAB INDIA 5.7
1900206 SANDEEP 5/11/1997 BATALA PUNJAB INDIA 7.9
1900207 AMIT 21/12/2004 JALANDHAR PUNJAB INDIA 5.5
1900208 PRABHNOOR 9/9/2001 TARN TARAN PUNJAB INDIA 4.4
1900209 MUSKAN 18/7/2006 BATALA PUNJAB INDIA 8.2
1900210 PAYAL 17/10/2000 AMRITSAR PUNJAB INDIA 7.2

Total no. of records came out = 19

(c) Students from a particular state and born in a particular month

Enter the State:punjab

Enter the Month (in number):12

['U.ROLLNO', 'FULL NAME', 'MM/DD/YYYY', 'CITY', 'STATE', 'COUNTRY', 'MARKS']

1900188	MANPREET SINGH	20/12/2001	AMRITSAR	PUNJAB	INDIA	8.8
1900202	ISHITA	11/12/1992	TARN TARAN	PUNJAB	INDIA	7.5
1900207	AMIT	21/12/2004	JALANDHAR	PUNJAB	INDIA	5.5

Total no. of records came out = 3

(d) Students with a given CGPA

Enter the CGPA : 8.8

['U.ROLLNO', 'FULL NAME', 'MM/DD/YYYY', 'CITY', 'STATE', 'COUNTRY', 'MARKS']

1900188	MANPREET SINGH	20/12/2001	AMRITSAR	PUNJAB	INDIA	8.8
---------	----------------	------------	----------	--------	-------	-----

Total no. of records came out = 1

(e) Student whose name starts with a particular letter and a particular year of birth:

Enter the first letter of Student's name : m

Enter Year of birth (in yyyy) : 2000

['U.ROLLNO', 'FULL NAME', 'MM/DD/YYYY', 'CITY', 'STATE', 'COUNTRY', 'MARKS']

1900183	MANINDER SINGH	21/11/2000	TARN TARAN	PUNJAB	INDIA	8.04
1900196	MEHBUB RAJA	30/4/2000	KATHMANDU	BAGMATI	NEPAL	7.4

Total no. of records came out = 2

Problem 3

Case study 2: Movie Review Dataset Case Study

The dataset is a simple text (movies_data.csv) file which has details like serial number, movie name, release year, rating and runtime (in seconds). It's a CSV file.

The file has a total of 49590 records.

Write python script to answer the following queries:

- (1) List the movie names and rating of all movies having a rating greater than 4.3.
- (2) List the movie names and release year of all movies released before 1990 and do not have a null in the rating
- (3) List the movie names and rating of all movies in which movie name contains the word "boys" or the word "wild".
- (4) List all movie names and release year for the movie names beginning with "A" and rating is null and release year is not between 1980 and 1995.
- (5) Count the number of movies released in the year 1989 and duration is more than 1½ hours.

Source Code of Problem 3:

```
from os import system
import csv
import colorama
colorama.init()

def query1():
    system('cls')
    f=open("Dataset-movies.csv","r")
    csvreader=csv.reader(f)
    count=0
    print("(1) List of movies having a rating greater than 4.4\n")
    for i in csvreader:
        if i[3] not in (None,''):
            if float(i[3])>4.4:
                count+=1
                print(i[1].ljust(40),i[3])
    if (count ==0):
        print('Sorry Data not Found')
    else:
        print('- '*45)
        print('Total no. of records came out =',count)
```

```

f.close()
input()

def query2():
    system('cls')
    f=open("Dataset-movies.csv","r")
    csvreader=csv.reader(f)
    count=0
    for i in csvreader:
        if i[3] not in (None,''):
            if int(i[2])<1990 :
                count+=1
                print(i[1].ljust(45),i[2])
    if (count ==0):
        print('Sorry Data not Found')
    else:
        print('- '*50)
        print('Total no. of records came out =',count)
    f.close()
    input()

def query3():
    system('cls')
    f=open("Dataset-movies.csv","r")
    csvreader=csv.reader(f)
    count=0
    for i in csvreader:
        if (i[1].lower().find("boys") != -1) or (i[1].lower().find("wild") != -1):
            count+=1
            print(i[1].ljust(90),i[3])
    if (count ==0):
        print('Sorry Data not Found')
    else:
        print('- '*100)
        print('Total no. of records came out =',count)
    f.close()
    input()

def query4():
    system('cls')
    f=open("Dataset-movies.csv","r")
    csvreader=csv.reader(f)
    count=0
    for i in csvreader:
        if i[3] in (None,''):
            if i[1].startswith("A") and (int(i[2])<1980 or int(i[2])>1995):
                count+=1
                print(i[1].ljust(90),i[2])

```

```

    if (count == 0):
        print('Sorry Data not Found')
    else:
        print('- '*100)
        print('Total no. of records came out =',count)
    f.close()
    input()

def query5():
    #system('cls')
    f=open("Dataset-movies.csv","r")
    csvreader=csv.reader(f)
    count=0
    for i in csvreader:
        if i[2] not in (None,'') and i[4] not in (None,''):
            if int(i[2])==1989 and int(i[4])>5400:
                count+=1
    if (count ==0):
        print('Sorry Data not Found')
    else:
        print('- '*50)
        print('The number of movies released in the year 1989 and duration is more than 1
1/2 hour =',count)
    f.close()
    input()

while(True):
    system('cls')
    print(colorama.Fore.YELLOW ,""*10+"Movie Review Dataset Case Study"+"*10)
    print()
    print("1.List of movies having a rating greater than 4.4")
    print("2.List of movies released before 1990 and do not have a null in the rating")
    print("3.List of movies in which movie name contains the word "boys" or the word "wil
d")
    print("4.List of movie names beginning with "A" and rating is null and release year i
s not between 1980 and 1995")
    print("5.Count the number of movies released in the year 1989 and duration is more th
an 1 1/2 hours.")
    print("press 0 for Exit")

    try:
        ch=int(input('\nEnter Your Choice : '))
        if ch==1:
            query1()
        elif ch==2:
            query2()
        elif ch==3:

```

```

        query5()
    elif ch==4:
        query4()
    elif ch==5:
        query5()
    elif ch==0:
        print('BYE!!')
        exit()
    else:
        print('Error!! Incorrect value entered')
        input()
except ValueError:
    print('Error!! Incorrect value entered')
    input()

```

Output Snapshots of Problem 3:

***** Movie Review Dataset Case Study *****

- 1.List of movies having a rating greater than 4.4
 - 2.List of movies released before 1990 and do not have a null in the rating
 - 3.List of movies in which movie name contains the word "boys" or the word "wild"
 - 4.List of movie names beginning with "A" and rating is null and release year is not between 1980 and 1995
 - 5.Count the number of movies released in the year 1989 and duration is more than 1½ hours.
- press 0 for Exit

Enter Your Choice : █

(1st part):

(1) List of movies having a rating greater than 4.4

Breaking Bad: Season 1	4.5
Breaking Bad: Season 2	4.5
Breaking Bad: Season 3	4.5
Breaking Bad	4.5
The Walking Dead: Season 1	4.5
Breaking Bad: Season 4	4.5
Sherlock: Series 1	4.5
The Walking Dead: Season 2	4.5
The Walking Dead	4.5
Sherlock	4.5
The Avengers	4.5
Sherlock: Series 2	4.5
Breaking Bad: Season 5	4.5
Orange Is the New Black: Season 1	4.5
Orange Is the New Black	4.5
The Walking Dead: Season 3	4.5
Fairy Tail: Season 1	4.5
Fairy Tail	4.5
Blackfish	4.5
Arrested Development (Trailer)	4.5
The Fosters	4.5
The Fosters: Season 1	4.5
Lilyhammer: Season 2 (Trailer)	4.5

Total no. of records came out = 23

(2nd part): This query results in 1463 rows. Showing below some part of it only.

Kung-fu Gold	1975
Carry On Cleo	1964
Carry On Cowboy	1966
Kit Carson	1940
The Lion of Thebes	1964
Let 'Em Have It	1935
Capone	1989
The Crystal Ball	1943
Confessions of Tom Harris	1969
The Club	1980
The Corsican Brothers	1941
Deadly Hero	1975
Danny	1977
Escape from Hell Island	1963
The Diamond Hunters	1975
The Duke of West Point	1938
Heavenly Pursuits	1986
Brewster's Millions	1945
Black Klansman	1966
Black Beauty	1946
The Christmas That Almost Wasn't	1966
Chinese Boxes	1984
Buffalo Soldiers	1970
Brotherhood of Death	1976
8 Million Ways to Die	1986
Two Roses and a Golden Rod	1969
The Telephone Book	1971
Night of the Cobra Woman	1972
Ecstasies of Women	1969
Linda and Abilene	1969
Blue Summer	1973
Crime Zone	1988
Barracuda	1978
Prison Girls	1972

Total no. of records came out = 1463

(3rd part): This query results in 799 rows. Showing below some part of it only.

Gator Boys: Season 2: Alligator Face-Off	
Gator Boys: Season 2: Gators Gone Wild	
Gator Boys: Season 2	4.1
Sister Wives: Season 4: Brown Boys Do Vegas	
Alaska: The Last Frontier: Season 1: The River Wild	
Wild Animal Baby Explorers	2.9
Glee: Season 4: Girls (and Boys) on Film	
Life With Boys	4.1
Life With Boys: Season 1: Driven Crazy with Boys	
Life With Boys: Season 1: Blah Blah Blah with Boys	
Life With Boys: Season 1: Smokin' with Boys	
Life With Boys: Season 1: Nightmares with Boys	
Life With Boys: Season 1: Fashion Faux Pas with Boys	
Life With Boys: Season 1: Birthdays with Boys	
Life With Boys: Season 1: When Something Better Comes Along with Boys	
Life With Boys: Season 1: Hitting the Breaks with Boys	
Life With Boys: Season 1: Double Trouble with Boys	
Life With Boys: Season 1: Trouble with Boys	
Life With Boys: Season 1: Bathroom Battles with Boys	
Life With Boys: Season 1: Misguided Motives with Boys	
Life With Boys: Season 1: Chrisbus with Boys	
Life With Boys: Season 1: Disarmed with Boys	
Life With Boys: Season 1: Set Up's with Boys	
Life With Boys: Season 1: The Big Kiss Off with Boys	
Life With Boys: Season 1: Social Death with Boys	
Life With Boys: Season 1: This Time the Problem Is with Dad and Not with Boys	
Life With Boys: Season 1: In the Principal's Office with Boys	
Life With Boys: Season 1: Monkey Talk with Boys	
Life With Boys: Season 1: A Perfect Life with Boys	
Life With Boys: Season 1: Wrestling with Boys	
Life With Boys: Season 1	4.1

Total no. of records came out = 799

(4th part): This query results in 1463 rows. Showing below some part of it only.

```
Auction Kings: Season 3: Mountain Man Slot Machine Wild West BMW Motorcycle      2012
Auction Kings: Season 3: John Wayne Guns Omnibot 2000                          2012
Auction Kings: Season 3: Harley Bike Lincoln Document                          2012
Auction Kings: Season 3: Ship's Cannon Vitapulser                             2012
Auction Kings: Season 3: Edison Phonograph Ali-Frazier Boxing Gloves            2012
Auction Kings: Season 3: Elvis Bracelet Silver Columbus Ships                  2012
Auction Kings: Season 3: Revisited: Rick Fairless Chopper Gas-Powered Blender    2012
Auction Kings: Season 3: Beatles Butcher Album Fire Grenade                   2012
Auction Kings: Season 3: Antique Doctor's Buggy Cleopatra Folio                2012
Auction Kings: Season 3: Skee-Ball Antique Fishing Gear                        2012
Auction Kings: Season 3: Jackie Robinson Memorabilia 1875 NYC Map               2012
Auction Kings: Season 3: Redneck Chariot Japanese Antique Daggers              2012
Auction Kings: Season 3: Wild West Memorabilia NFL Helmet                      2012
Alaska: The Last Frontier: Season 1: Something's Fishy                        2011
Alaska: The Last Frontier: Season 1: Fall Feast                               2011
Alaska: The Last Frontier: Season 1: Spring Has Sprung                         2011
Alaska: The Last Frontier: Season 1: Cattle Drive                             2011
Alaska: The Last Frontier: Season 1: Range Riding                             2011
Alaska: The Last Frontier: Season 1: The River Wild                           2011
Alaska: The Last Frontier: Season 1: Before the Freeze                         2011
Alaska: The Last Frontier: Season 1: Fueling the Fire                         2011
Alaska: The Last Frontier: Season 1: Snow Cold and Darkness                    2011
Alaska: The Last Frontier: Season 1: Dead of Winter                           2011
Alaska: Ice Cold Killers: Season 1: Frozen Terror                             2012
Alaska: Ice Cold Killers: Season 1: Mountain Man                             2012
Alaska: Ice Cold Killers: Season 1: Hunting Humans                             2012
```

Total no. of records came out = 1787

(5th part):

```
1.List of movies having a rating greater than 4.4
2.List of movies released before 1990 and do not have a null in the rating
3.List of movies in which movie name contains the word "boys" or the word "wild"
4.List of movie names beginning with "A" and rating is null and release year is not between 1980 and 1995
5.Count the number of movies released in the year 1989 and duration is more than 1½ hours.
press 0 for Exit
```

Enter Your Choice : 5

The number of movies released in the year 1989 and duration is more than 1½ hour = 53

Problem 4

Q. Choose a dataset (CSV file) of your choice with at least 300 records. Generate and write solutions for at least 10 queries based on this dataset. Suggested Datasets include: Health datasets, Weather datasets, Agriculture datasets, Soil datasets, Rainfall datasets, Retail stores dataset etc.

Source Code of Problem 4:

```
import csv
from os import system
import colorama
colorama.init()

def query1():
    system("cls")
    f=open("p1_4cse1_project_dataset_3_sales_record.csv","r")
    csvreader=csv.reader(f)
    heading=next(csvreader)
    #print(heading)
    count=0
    print("--Query 1--")
    br=input('\nEnter the Brand you want to see : ').upper()
    hrpw=int(input('Enter minimum horsepower you want to see : '))
    print('\nBrand      Horsepower    Torque      Model-Name ')
    for i in csvreader:
        if i[13].upper()==br and int(i[16])>=hrpw:
            count+=1
            print(i[13].ljust(15),i[16].ljust(10),i[17].ljust(8),i[14])
    if (count ==0):
        print("Sorry Data Not Found")
    else:
        print('- '*80)
        print("Total no. of records came out =",count)
    f.close()
    input()

def query2():
    system("cls")
    f=open("p1_4cse1_project_dataset_3_sales_record.csv","r")
    csvreader=csv.reader(f)
    heading=next(csvreader)
    count=0
    print("--Query 2--")
    br=input('\nEnter the Brand you want to see : ').upper()
```

```

print("\nProduct Name :{:just(35)} Height Length Breadth ")
for i in csvreader:
    if i[13].upper()=='br':
        count+=1
        print(i[14].ljust(35),i[0].ljust(6),i[1].ljust(6),i[2])
if (count ==0):
    print("Sorry Data Not Found")
else:
    print('-'*100)
    print("Total no. of records came out =",count)
f.close()
input()
def query3():
    system("cls")
    f=open("p1_4cse1_project_dataset_3_sales_record.csv","r")
    csvreader=csv.reader(f)
    heading=next(csvreader)
    count=0
    print("--Query 3--")
    print('\n      Car ID'.ljust(50)+'Transmission'.ljust(35))
    for i in csvreader:
        if i[15] in (None,''):
            count+=1
            print(i[12].ljust(45),i[11].ljust(35))
    if (count ==0):
        print("Sorry Data Not Found")
    else:
        print('-'*100)
        print("Total no. of records came out =",count)
    f.close()
    input()
def query4():
    system("cls")
    f=open("p1_4cse1_project_dataset_3_sales_record.csv","r")
    csvreader=csv.reader(f)
    heading=next(csvreader)
    print("--Query 4--\n")
    brand_list=[]
    for i in csvreader:
        brand_list.append(i[13])
    brand_list_set=list(sorted(set(brand_list)))
    count=len(brand_list_set)
    prev=brand_list_set[0][:1]
    for i in range(count):
        first=brand_list_set[i][:1]
        if first==prev:
            print(brand_list_set[i].ljust(20),end="")
        else:

```

```

        print(brand_list_set[i].ljust(20),end="")
    prev=first
print()
if (count ==0):
    print("Sorry Data Not Found")
else:
    print('- '*100)
    print("Total no. of Brands =",count)
f.close()
input()
def query5():
    system("cls")
    f=open("p1_4cse1_project_dataset_3_sales_record.csv","r")
    csvreader=csv.reader(f)
    heading=next(csvreader)
    print("--Query 5--\n")
    ent = input("Enter the first letter of brand : ").upper()
    brand_list=[]
    for i in csvreader:
        brand_list.append(i[13])
    brand_set=set(brand_list)
    count=0
    for i in brand_set:
        if i.startswith(ent):
            count+=1
            print(i)
    if (count ==0):
        print("Sorry Data Not Found")
    else:
        print('- '*30)
        print("Total no. of records came out =",count)
    f.close()
    input()
def query6():
    system("cls")
    f=open("p1_4cse1_project_dataset_3_sales_record.csv","r")
    csvreader=csv.reader(f)
    heading=next(csvreader)
    count=0
    for i in csvreader:
        if i[12].upper().find("CONVERTIBLE") != -1 :
            count+=1
            print(i[13].ljust(20),i[14].ljust(50),"      ",i[0],"x",i[1],"x",i[2])
    if (count ==0):
        print("Sorry Data Not Found")
    else:
        print('- '*100)

```

```

        print("Total no. of records came out =",count)
    f.close()
    input()
def query7():
    system("cls")
    f=open("p1_4cse1_project_dataset_3_sales_record.csv","r")
    csvreader=csv.reader(f)
    heading=next(csvreader)
    #print(heading)
    print('--Query 7--\n')
    y1=int(input('Enter the year 1 : '))
    y2=int(input('Enter the year 2 : '))
    print('\nBrand'.ljust(15)+'      Model'.ljust(35)+'Released Year')
    count=0
    for i in csvreader:
        if i[15] not in (None,''):
            if i[11][:6]=='Manual' and int(i[15])>y1 and int(i[15])<y2:
                count+=1
                print(i[13].ljust(15),i[14].ljust(35),i[15])
    if (count ==0):
        print("Sorry Data Not Found")
    else:
        print('- '*100)
        print("Total no. of records came out =",count)
    f.close()
    input()

def query8():
    system("cls")
    f=open("p1_4cse1_project_dataset_3_sales_record.csv","r")
    csvreader=csv.reader(f)
    heading=next(csvreader)
    #print(heading)
    count=0
    for i in csvreader:
        if i[11] == "Automatic transmission" and i[3] == "Rear-
wheel drive" and i[9]=="Gasoline" and i[8]>"12":
            count+=1
            print(i[13].ljust(25),i[14])
    if (count ==0):
        print("Sorry Data Not Found")
    else:
        print('- '*100)
        print("Total no. of records came out =",count)
    f.close()
    input()

def query9():

```



```

system("cls")
f=open("p1_4cse1_project_dataset_3_sales_record.csv","r")
csvreader=csv.reader(f)
heading=next(csvreader)
count=0
for i in csvreader:
    if i[11] == "Manual transmission" and i[16]>"255" and i[17]<"500":
        count+=1
        print(i[13].ljust(15),i[14].ljust(48),i[3].ljust(25),i[4])
if (count ==0):
    print("Sorry Data Not Found")
else:
    print('-'*100)
    print("Total no. of records came out =",count)
f.close()
input()
def query10():
    system("cls")
    f=open("p1_4cse1_project_dataset_3_sales_record.csv","r")
    csvreader=csv.reader(f)
    heading=next(csvreader)
    count=0
    for i in csvreader:
        if i[13] == "Honda" and i[15]<"2011":
            count+=1
            print(i[13].ljust(8),i[14].ljust(38),i[3].ljust(30),i[4])
    if (count ==0):
        print("Sorry Data Not Found")
    else:
        print('-'*100)
        print("Total no. of records came out =",count)
    f.close()
    input()
def query11():
    system("cls")
    f=open("p1_4cse1_project_dataset_3_sales_record.csv","r")
    csvreader=csv.reader(f)
    heading=next(csvreader)
    count=0
    for i in csvreader:
        if i[7] == "4 Speed Automatic" and i[10]<"32":
            count+=1
            print(i[13].ljust(20),i[14].ljust(45),i[16].ljust(8),i[17])
    if (count ==0):
        print("Sorry Data Not Found")
    else:
        print('-'*100)
        print("Total no. of records came out =",count)

```

```

    input()

while(True):
    system('cls')
    print(colorama.Fore.YELLOW , "***** Automobile Information Dataset Case S
tudy *****\n")
    print('1.List all the BMW cars having More than a desired Horsepower.\n')
    print('2.List the model and dimension of all the cars of a specific brand ent
ered by User.\n')
    print('3.List the Car ID and Transmission type of the cars which does not rel
eased in the market.\n')
    print('4.List all The Car Brands available in row-
wise alphabetical order.\n')
    print('5.List all The Car Brands starting with a particular letter.\n ')
    print('6.List all the models with dimension which contains word "convertible"
in their model name.\n')
    print("7.List all the manual cars released between the two specific years\n")
    print("8.Print Brand, Model of All the Automatic Cars Whose Driveline is Rear
Wheel Drive,Fuel type is Gasoline and city milage is greater than 12.\n")
    print("9.List All the Manual transmitted Cars Whose Engine Horsepower is grea
ter than 255 and Engine Torque is less than 500. Print their Maker name,Model,Dri
veline,Engine info/type.\n ")
    print("10.list All the Cars Whose Maker is Honda and releasing year is before
2011. Print their model ,Driveline,Engine information and type.\n")
    print("11.List All the Cars Whose transmission is 4 Speed Automatic and Highw
ay Milage is less than 32 print their Maker Model Engine Horsepower and Torque.\n
")
    print("press 0 for Exit")
    try:
        ch=int(input('\nEnter Your Choice : '))
        if ch==1:
            query1()
        elif ch==2:
            query2()
        elif ch==3:
            query3()
        elif ch==4:
            query4()
        elif ch==5:
            query5()
        elif ch==6:
            query6()
        elif ch==7:
            query7()
        elif ch==8:
            query8()
        elif ch==9:

```

```

        query5()
    elif ch==10:
        query10()
    elif ch==11:
        query11()
    elif ch==0:
        print('BYE!!!')
        exit()
except ValueError:
    print('Error!! Incorrect value entered')

```

Output Snapshots of Problem 4:

```

***** Automobile Information Dataset Case Study *****

1.List all the BMW cars having More than a desired Horsepower.

2.List the model and dimension of all the cars of a specific brand entered by User.

3.List the Car ID and Transmission type of the cars which does not released in the market.

4.List all The Car Brands available in row-wise alphabetical order.

5.List all The Car Brands starting with a particular letter.

6.List all the models with dimension which contains word "convertible" in their model name.

7.List all the manual cars released between the two specific years

8.Print Brand, Model of All the Automatic Cars Whose Driveline is Rear Wheel Drive,Fuel type is Gasoline and city
milage is greater than 12.

9.List All the Manual transmitted Cars Whose Engine Horsepower is greater than 255 and Engine Torque is less than
500. Print their Maker name,Model,Driveline,Engine info/type.

10.list All the Cars Whose Maker is Honda and releasing year is before 2011. Print their model ,Driveline,Engine
information and type.

11.List All the Cars Whose transmission is 4 Speed Automatic and Highway Milage is less than 32 print their Maker
Model Engine Horsepower and Torque.

press 0 for Exit

Enter Your Choice : █

```

--Query 1--

Enter the Brand you want to see : Bentley

Enter minimum horsepower you want to see : 500

Brand	Horsepower	Torque	Model-Name
Bentley	530	774	2010 Bentley Brooklands
Bentley	621	590	2012 Bentley Continental Supersports Convertible
Bentley	621	590	2012 Bentley Continental Supersports Coupe
Bentley	621	590	2011 Bentley Continental Supersports Convertible
Bentley	621	590	2010 Bentley Continental Supersports
Bentley	552	479	2012 Bentley Continental Flying Spur
Bentley	600	553	2012 Bentley Continental Flying Spur
Bentley	600	553	2011 Bentley Continental Flying Spur
Bentley	600	553	2010 Bentley Continental Flying Spur
Bentley	552	479	2010 Bentley Continental Flying Spur
Bentley	567	516	2012 Bentley Continental GT
Bentley	552	479	2010 Bentley Continental GT
Bentley	600	553	2010 Bentley Continental GT
Bentley	600	553	2011 Bentley Continental GTC
Bentley	600	553	2010 Bentley Continental GTC
Bentley	552	479	2010 Bentley Continental GTC
Bentley	500	738	2010 Bentley Azure T
Bentley	505	752	2012 Bentley Mulsanne

Total no. of records came out = 18

Query 2's output has total 133 rows. Showing below some part of it.

--Query 2--

Enter the Brand you want to see : bmw

Model Name	Height	Length	Width
2010 BMW X5	226	243	201
2011 BMW X5	226	243	201
2010 BMW X6	148	12	191
2011 BMW X6	148	12	191
2010 BMW 1 Series Convertible	132	9	211
2010 BMW 1 Series Convertible	132	9	211
2011 BMW 1 Series Convertible	132	9	211
2010 BMW 3 Series Convertible	104	227	247
2010 BMW 3 Series Convertible	104	227	247
2011 BMW 3 Series Convertible	104	12	247
2011 BMW 3 Series Convertible	91	12	247
2011 BMW 3 Series Convertible	104	12	247
2010 BMW 3 Series Coup	114	247	247
2010 BMW 3 Series Coup	94	247	247
2010 BMW 3 Series Coup	94	247	247
2010 BMW 3 Series Coup	114	247	247
2011 BMW 3 Series Coup	114	12	247
2011 BMW 3 Series Coup	94	12	247
2011 BMW 3 Series Coup	94	12	247
2011 BMW 3 Series Coup	114	12	247
2011 BMW 3 Series Coup	94	12	247
2010 BMW 3 Series Sedan	139	174	24
2010 BMW 3 Series Sedan	144	174	24
2010 BMW 3 Series Sedan	144	174	24
2010 BMW 3 Series Sedan	144	174	24
2010 BMW 3 Series Sedan	139	174	24
2011 BMW 3 Series Sedan	144	174	24
2011 BMW 3 Series Sedan	144	174	24
2011 BMW 3 Series Sedan	139	174	24
2011 BMW 3 Series Sedan	144	174	24
2011 BMW 3 Series Sedan	139	174	24
2010 BMW 5 Series Sedan	200	245	54

--Query 3--

Car ID	Transmission	City mpg
2009 Audi Q7 3.6	Automatic transmission	14
2012 Acura TL	Automatic transmission	20
2011 BMW X6 M	Automatic transmission	12
2012 Chevrolet Colorado Work Truck	Manual transmission	18
2012 Chevrolet Express LS 1500	Automatic transmission	13
2011 Volvo C30 T5	Manual transmission	21
2010 Volvo S40 R-Design	Automatic transmission	21
2010 Toyota 4Runner SR5	Automatic transmission	18
2011 Nissan 370Z Coupe	Manual transmission	18
2011 Volvo C30 R-Design AT	Automatic transmission	21
2011 Nissan Armada SV 4x4 FFV	Automatic transmission	12
2012 Hyundai Elantra Sedan Limited PZEV	Automatic transmission	29
2011 Hyundai Elantra Sedan Limited	Automatic transmission	29
2011 Chevrolet Cruze LTZ	Automatic transmission	24
2011 Cadillac SRX	Automatic transmission	18
2010 Cadillac Escalade EXT Luxury	Automatic transmission	10
2011 Saab 9-4X 3.0i Premium	Automatic transmission	18
2010 Toyota Tundra 4.6L 4X4	Automatic transmission	14
2010 Toyota Tundra 5.7L	Automatic transmission	14
2010 Toyota Tundra Double Cab	Automatic transmission	15
2010 Toyota Tundra CrewMax	Automatic transmission	15
2011 Ford E-250 Van Commercial	Automatic transmission	13

Total no. of records came out = 22

--Query 4--

AMG	Acura	Aston Martin	Audi		
BMW	BMW Motorrad	Bentley	Buick		
Cadillac	Chevrolet	Chrysler	Chrysler Group LLC		
Dodge					
Ferrari	Ford				
GMC	Grand Cherokee				
Honda	Hyundai				
Infiniti					
Jaguar	Jeep				
Kia					
Lamborghini	Land Rover	Lexus	Lincoln	Lotus	
MINI	Maserati	Maybach	Mazda	Mercedes	Mercedes-AMG
Mercedes-Benz	Mercury	Mitsubishi			
Nissan					
Porsche					
Rolls-Royce					
Saab	Scion	Subaru	Suzuki		
Toyota					
Volkswagen	Volvo				

Total no. of Brands = 47

|

--Query 5--

Enter the first letter of brand : m

Mazda
Maybach
Mercedes
MINI
Maserati
Mercedes-Benz
Mercury
Mercedes-AMG
Mitsubishi

Total no. of records came out = 9

|

Query 6's output has total 182 rows. Showing below some part of it

Lexus	2010 Lexus IS C	134 x 27 x 8
Lexus	2011 Lexus IS C	135 x 22 x 8
Lexus	2011 Lexus IS C	135 x 27 x 8
Lexus	2011 Lexus IS C	135 x 27 x 8
BMW Motorrad	2012 BMW 1 Series Convertible	132 x 21 x 211
BMW Motorrad	2012 BMW 1 Series Convertible	132 x 21 x 211
BMW	2012 BMW 1 Series Convertible	132 x 21 x 211
Infiniti	2012 Infiniti G37 Convertible	117 x 47 x 59
Infiniti	2012 Infiniti G37 Convertible	119 x 65 x 59
Jaguar	2012 Jaguar XK Convertible	49 x 186 x 240
Jaguar	2012 Jaguar XK Convertible	49 x 186 x 240
BMW Motorrad	2012 BMW 3 Series Convertible	104 x 12 x 247
BMW Motorrad	2012 BMW 3 Series Convertible	104 x 12 x 247
BMW	2012 BMW 3 Series Convertible	104 x 12 x 247
BMW	2012 BMW 3 Series Convertible	104 x 12 x 247
BMW	2012 BMW 3 Series Convertible	91 x 12 x 247
BMW	2012 BMW 3 Series Convertible	91 x 12 x 247
BMW	2012 BMW 3 Series Convertible	111 x 7 x 11
BMW	2012 BMW 3 Series Convertible	111 x 7 x 11
Saab	2012 Saab 9-3 Convertible	157 x 37 x 244
Saab	2012 Saab 9-3 Convertible	157 x 37 x 244
Saab	2012 Saab 9-3 Convertible	157 x 37 x 244
Saab	2012 Saab 9-3 Convertible	157 x 37 x 244
Saab	2012 Saab 9-3 Convertible	157 x 37 x 244
Saab	2012 Saab 9-3 Convertible	157 x 37 x 244

Total no. of records came out = 182

--Query 7--

Enter the year 1 : 2005

Enter the year 2 : 2010

Brand	Model	Released Year
Audi	2009 Audi A3	2009
Audi	2009 Audi A5	2009
Audi	2009 Audi A4 Sedan	2009
Nissan	2009 Nissan 370Z Coupe	2009
Nissan	2009 Nissan 370Z Coupe	2009
Nissan	2009 Nissan 370Z Coupe	2009
Nissan	2009 Nissan Cube	2009
Nissan	2009 Nissan Cube	2009
Hyundai	2009 Hyundai Elantra Touring	2009
Infiniti	2009 Infiniti G37 Convertible	2009
Lotus	2009 Lotus Elise	2009
Lotus	2009 Lotus Elise	2009
Lotus	2009 Lotus Exige	2009
Lotus	2009 Lotus Exige	2009

Total no. of records came out = 14

Query 8's output has total 1038 rows. Showing below some part of it.

Cadillac	2012 Cadillac Escalade
Cadillac	2012 Cadillac Escalade
Cadillac	2012 Cadillac Escalade
Cadillac	2012 Cadillac Escalade Hybrid
Cadillac	2012 Cadillac Escalade Hybrid
Cadillac	2012 Cadillac Escalade ESV
Cadillac	2012 Cadillac Escalade ESV
Cadillac	2012 Cadillac Escalade ESV
BMW Motorrad	2012 BMW 3 Series Convertible
BMW	2012 BMW 3 Series Convertible
BMW	2012 BMW 3 Series Convertible
BMW	2012 BMW 3 Series Convertible
BMW	2012 BMW 5 Series Sedan
BMW	2012 BMW 5 Series Sedan
BMW	2012 BMW 5 Series Sedan
Aston Martin	2012 Aston Martin DB9 Coup
Aston Martin	2012 Aston Martin DB9 Coup
Aston Martin	2012 Aston Martin DB9 Coup
Aston Martin	2012 Aston Martin Rapide
Aston Martin	2012 Aston Martin Rapide
BMW	2012 BMW 7 Series
BMW	2012 BMW 7 Series

Total no. of records came out = 1038

Query 9's output has total 360 rows. Showing below some part of it.

Dodge	2012 Dodge Challenger	Rear-wheel drive	Dodge 6.4L 8 Cylinder 470 hp 470 ft-lbs
Mazda	2012 Mazda Mazdaspeed3	Front-wheel drive	Mazda 2.3L 4 Cylinder 263 hp 280 ft-lbs Turbo
Volkswagen	2012 Volkswagen Golf R 2-Door	All-wheel drive	Volkswagen 2.0L 4 Cylinder 256 hp 243 ft-lbs Turbo
Cadillac	2012 Cadillac CTS Sport Sedan	Rear-wheel drive	Cadillac 3.0L 6 Cylinder 270 hp 223 ft-lbs
Cadillac	2012 Cadillac CTS Sport Sedan	Rear-wheel drive	Cadillac 3.0L 6 Cylinder 270 hp 223 ft-lbs
Volkswagen	2012 Volkswagen Golf R 4-Door	All-wheel drive	Volkswagen 2.0L 4 Cylinder 256 hp 243 ft-lbs Turbo
Jeep	2012 Jeep Wrangler	Four-wheel drive	Jeep 3.6L 6 Cylinder 280 hp 260 lb-ft
Jeep	2012 Jeep Wrangler	Four-wheel drive	Jeep 3.6L 6 Cylinder 280 hp 260 lb-ft
Jeep	2012 Jeep Wrangler	Four-wheel drive	Jeep 3.6L 6 Cylinder 280 hp 260 lb-ft
Jeep	2012 Jeep Wrangler	Four-wheel drive	Jeep 3.6L 6 Cylinder 280 hp 260 lb-ft
Jeep	2012 Jeep Wrangler	Four-wheel drive	Jeep 3.6L 6 Cylinder 280 hp 260 lb-ft
Jeep	2012 Jeep Wrangler	Four-wheel drive	Jeep 3.6L 6 Cylinder 280 hp 260 lb-ft
BMW	2012 BMW 3 Series Coup	Rear-wheel drive	BMW 3.0L 6 cylinder 300 hp 300 ft-lbs Turbo
BMW	2012 BMW 3 Series Coup	Rear-wheel drive	BMW 3.0L 6 cylinder 320 hp 332 ft-lbs Turbo
BMW	2012 BMW 3 Series Coup	All-wheel drive	BMW 3.0L 6 cylinder 300 hp 300 ft-lbs Turbo
BMW	2012 BMW 3 Series Coup	Rear-wheel drive	BMW 4.0L 8 cylinder 414 hp 295 ft-lbs
BMW	2012 BMW 3 Series Convertible	Rear-wheel drive	BMW 3.0L 6 cylinder 300 hp 300 ft-lbs Turbo
BMW	2012 BMW 3 Series Convertible	Rear-wheel drive	BMW 3.0L 6 cylinder 320 hp 332 ft-lbs Turbo
BMW	2012 BMW 3 Series Convertible	Rear-wheel drive	BMW 4.0L 8 cylinder 414 hp 295 ft-lbs
BMW	2012 BMW 5 Series Sedan	Rear-wheel drive	BMW 3.0L 6 cylinder 300 hp 300 ft-lbs Turbo
BMW	2012 BMW 5 Series Sedan	Rear-wheel drive	BMW 4.4L 8 cylinder 400 hp 450 ft-lbs Turbo
Aston Martin	2012 Aston Martin DB9 Volante	Rear-wheel drive	Aston Martin 5.9L 12 Cylinder 470 hp 443 ft-lbs
Aston Martin	2012 Aston Martin DB9 Volante	Rear-wheel drive	Aston Martin 5.9L 12 Cylinder 470 hp 443 ft-lbs
Subaru	2012 Subaru Legacy	All-wheel drive	Subaru 2.5L 4 cylinder 265 hp 258 ft-lbs Turbo
Aston Martin	2012 Aston Martin DBS Volante	Rear-wheel drive	Aston Martin 5.9L 12 Cylinder 510 hp 420 ft-lbs
Aston Martin	2012 Aston Martin DBS Volante	Rear-wheel drive	Aston Martin 5.9L 12 Cylinder 510 hp 420 ft-lbs
Aston Martin	2012 Aston Martin V12 Vantage	Rear-wheel drive	Aston Martin 6.0L 510 hp 420 ft-lbs
Aston Martin	2012 Aston Martin V12 Vantage	Rear-wheel drive	Aston Martin 6.0L 510 hp 420 ft-lbs
Aston Martin	2012 Aston Martin V8 Vantage Roadster	Rear-wheel drive	Aston Martin 4.7L 8 Cylinder 430 hp 361 ft-lbs
Aston Martin	2012 Aston Martin V8 Vantage Roadster	Rear-wheel drive	Aston Martin 4.7L 8 Cylinder 430 hp 361 ft-lbs
Aston Martin	2012 Aston Martin V8 Vantage Coup	Rear-wheel drive	Aston Martin 4.7L 8 cylinder 420 hp 346 ft-lbs
Aston Martin	2012 Aston Martin V8 Vantage Coup	Rear-wheel drive	Aston Martin 4.7L 8 cylinder 420 hp 346 ft-lbs
Lamborghini	2012 Lamborghini Gallardo Coup	All-wheel drive	Lamborghini 5.2L 10 cylinder 552 hp 398 ft-lbs
Lamborghini	2012 Lamborghini Gallardo Spyder	All-wheel drive	Lamborghini 5.2L 10 cylinder 552 hp 398 ft-lbs

Total no. of records came out = 360

Query 10's output has total 74 rows. Showing below some part of it.

Honda	2010	Honda Civic Sedan	Front-wheel drive	Honda 1.8L 4 Cylinder 140hp 128 ft-lbs
Honda	2010	Honda Civic Sedan	Front-wheel drive	Honda 1.8L 4 Cylinder 140hp 128 ft-lbs
Honda	2010	Honda Civic Sedan	Front-wheel drive	Honda 2.0L 4 Cylinder 197 hp 139 ft-lbs
Honda	2010	Honda Civic Sedan	Front-wheel drive	Honda 1.8L 4 Cylinder 113 hp 109 ft-lbs CNG
Honda	2010	Honda Fit	Front-wheel drive	Honda 1.5L 4 Cylinder 117 hp 106 ft-lbs
Honda	2010	Honda Fit	Front-wheel drive	Honda 1.5L 4 Cylinder 117 hp 106 ft-lbs
Honda	2010	Honda Fit	Front-wheel drive	Honda 1.5L 4 Cylinder 117 hp 106 ft-lbs
Honda	2010	Honda Fit	Front-wheel drive	Honda 1.5L 4 Cylinder 117 hp 106 ft-lbs
Honda	2010	Honda CR-V	Front-wheel drive	Honda 2.4L 4 Cylinder 180 hp 161 ft-lbs
Honda	2010	Honda CR-V	Four-wheel drive	Honda 2.4L 4 Cylinder 180 hp 161 ft-lbs
Honda	2010	Honda CR-V	Front-wheel drive	Honda 2.4L 4 Cylinder 180 hp 161 ft-lbs
Honda	2010	Honda CR-V	Four-wheel drive	Honda 2.4L 4 Cylinder 180 hp 161 ft-lbs
Honda	2010	Honda CR-V	Front-wheel drive	Honda 2.4L 4 Cylinder 180 hp 161 ft-lbs
Honda	2010	Honda CR-V	Four-wheel drive	Honda 2.4L 4 Cylinder 180 hp 161 ft-lbs
Honda	2010	Honda Element	Front-wheel drive	Honda 2.4L 4 Cylinder 166 hp 161 ft-lbs
Honda	2010	Honda Element	Four-wheel drive	Honda 2.4L 4 Cylinder 166 hp 161 ft-lbs
Honda	2010	Honda Element	Front-wheel drive	Honda 2.4L 4 Cylinder 166 hp 161 ft-lbs
Honda	2010	Honda Element	Four-wheel drive	Honda 2.4L 4 Cylinder 166 hp 161 ft-lbs
Honda	2010	Honda Element	Four-wheel drive	Honda 2.4L 4 Cylinder 166 hp 161 ft-lbs
Honda	2010	Honda Element	Front-wheel drive	Honda 2.4L 4 Cylinder 166 hp 161 ft-lbs
Honda	2010	Honda Element	Front-wheel drive	Honda 2.4L 4 Cylinder 166 hp 161 ft-lbs
Honda	2010	Honda Pilot	Front-wheel drive	Honda 3.5L 6 Cylinder 250 hp 253 ft-lbs
Honda	2010	Honda Pilot	Front-wheel drive	Honda 3.5L 6 Cylinder 250 hp 253 ft-lbs
Honda	2010	Honda Pilot	Front-wheel drive	Honda 3.5L 6 Cylinder 250 hp 253 ft-lbs
Honda	2010	Honda Pilot	Four-wheel drive	Honda 3.5L 6 Cylinder 250 hp 253 ft-lbs
Honda	2010	Honda Pilot	Front-wheel drive	Honda 3.5L 6 Cylinder 250 hp 253 ft-lbs
Honda	2010	Honda Odyssey	Front-wheel drive	Honda 3.5L 6 Cylinder 244 hp 240 ft-lbs
Honda	2010	Honda Odyssey	Front-wheel drive	Honda 3.5L 6 Cylinder 244 hp 240 ft-lbs
Honda	2010	Honda Odyssey	Front-wheel drive	Honda 3.5L 6 Cylinder 244 hp 245 ft-lbs
Honda	2010	Honda Odyssey	Front-wheel drive	Honda 3.5L 6 Cylinder 244 hp 245 ft-lbs
Honda	2010	Honda Ridgeline	Four-wheel drive	Honda 3.5L 6 Cylinder 250 hp 247 ft-lbs
Honda	2010	Honda Ridgeline	Four-wheel drive	Honda 3.5L 6 Cylinder 250 hp 247 ft-lbs
Honda	2010	Honda Ridgeline	Four-wheel drive	Honda 3.5L 6 Cylinder 250 hp 247 ft-lbs

Total no. of records came out = 74				

Query 11"s output has total 604 rows. Showing below some part of it.

Ford	2010 Ford F-150	248	294
Ford	2010 Ford F-150	248	294
Ford	2010 Ford F-150	248	294
Ford	2010 Ford F-150	248	294
Ford	2010 Ford F-150	248	294
Ford	2010 Ford F-150	248	294
Ford	2010 Ford F-150	248	294
Ford	2010 Ford F-150	248	294
Ford	2010 Ford F-150	248	294
Ford	2010 Ford F-150	248	294
Ford	2010 Ford F-150	248	294
Ford	2010 Ford F-150	248	294
GMC	2010 GMC Sierra 1500 Hybrid	332	367
GMC	2010 GMC Sierra 1500 Hybrid	332	367
GMC	2010 GMC Sierra 1500 Hybrid	332	367
GMC	2010 GMC Sierra 1500 Hybrid	332	367
Dodge	2011 Dodge Ram 1500	215	235
Dodge	2012 Dodge Avenger	173	166
GMC	2012 GMC Yukon Hybrid	332	367
GMC	2012 GMC Yukon Hybrid	332	367
GMC	2012 GMC Savana	310	334
GMC	2012 GMC Savana	310	334
GMC	2012 GMC Savana	310	334
GMC	2012 GMC Savana	310	334
Jeep	2012 Jeep Liberty	210	235
Jeep	2012 Jeep Liberty	210	235
Jeep	2012 Jeep Liberty	210	235
Jeep	2012 Jeep Liberty	210	235
Jeep	2012 Jeep Liberty	210	235
Jeep	2012 Jeep Liberty	210	235
Jeep	2012 Jeep Liberty	210	235
GMC	2012 GMC Savana	195	260
GMC	2012 GMC Savana	310	334
GMC	2012 GMC Savana	310	334

Total no. of records came out = 604			

REFERENCES

- 1 [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))
- 2 https://matplotlib.org/3.2.2/gallery/misc/plotfile_demo_sgskip.html
- 3 <https://docs.python.org/3/library/csv.html>
- 4 <https://www.geeksforgeeks.org/working-csv-files-python/>
- 5 <https://www.javatpoint.com/python-tutorial>
- 6 <https://www.geeksforgeeks.org/oracle-database-connection-in-python/>
- 7 <http://www.youtube.com/TechXpert>