**PRESIDENCY SCHOOL BANGALORE SOUTH**

****

**PROGRAMMING**

**IN**

**PYTHON & SQL**

**Subject: COMPUTER SCIENCE**

DONE BY: ***Saba Najam***

**XII ‘A’**

**2023-2024**

**CERTIFICATE**

Name: Class: 12th‘A’

Exam No: \_\_\_\_

This is certified to be the bonafide work of the student in the computer science laboratory during the academic year 2023-2024.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

TEACHER INCHARGE

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_ EXAMINER’S SIGNATURE PRINCIPAL

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Institution Rubber Stamp

**INDEX**

|  |  |
| --- | --- |
| **S.NO.** | **PROGRAM** |
| 1] | **Menu driven program:**  ➢ Factorial of a number.  ➢ Fibonacci series up to nth term.  ➢ Check a given number is Palindrome or not. |
| 2] | **Menu driven program:**  ➢ Check a given number is Armstrong number or not.  ➢ To display Floyd's triangle. |
| 3] | **Menu driven program:**  ➢ Function to display the names starting with the alphabet ‘M’.  ➢ Function to remove indicated letter from a string.  ➢ Count number of words in a string. |
| 4] | **Menu driven program (no built-in functions):**  ➢ Maximum of the element  ➢ Minimum of the element  ➢ Sum of the elements. |
| 5] | **Menu driven program (Text file):**  ➢ Display the content line by line with each word separated by “$”.  ➢ Remove the line which has ‘a’ and copy that line to another file. |
| 6] | **Menu driven program (Text file):**  ➢ Create a file with several lines of text.  ➢ Create another file which will store all the words starting with vowels.  ➢ Read & display the contents of both the files.  ➢ Display the total number of words starting with vowels. |
| 7] | **Menu driven program (Text file):**  ➢ Create a file with a few lines of text.  ➢ count number of lines, consonants, digits, spaces & words.  ➢ Create another file which will contain the text after replacing all the blank spaces with '#'. ➢ Read & display the contents of both the files. |
| 8] | **Menu driven program (Text file):**  ➢ To read lines from text file and display those words which are less than 4 characters. ➢ to search a word and its frequency in a text file. |
| 9] | **Menu driven program (Binary file):**  ➢ Create a file with the given structure.  ➢ Read contents of the file and display the details of those students whose percentage is above 75. Also display the number of students scoring above 75%. |
| 10] | **Menu driven program (Binary file):** |

|  |  |
| --- | --- |
|  | ➢ Create a file with the given structure.  ➢ Display the contents of the binary file.  ➢ Display the Company whose turnover is above user given value.  ➢ Search a Company by Company ID given by the user. |
| 11] | **Menu driven program (Binary file):**  ➢ Create a file with the given structure.  ➢ Append data to the file and update a record based on travelid.  ➢ Display the contents of the binary file. |
| 12] | **Menu driven program (CSV file):**  ➢ Create a file with the given structure.  ➢ read the file and calculate the total and percentage for each student.  ➢ display the name of student if in any subject marks are greater than 80%. |
| 13] | **Menu driven program (CSV file):**  ➢ Create a file with the given structure.  ➢ to search the record and display the record, otherwise display appropriate messages. |
| 14] | **Menu driven program:**  ➢ To perform Push, Pop operations on the stack named ‘Hostel’ with the given constraints. |
| 15] | **Menu driven program:**  ➢ To perform Push, Pop operations on the stack named ‘Book’ with the given constraints. |
| 16]  –  20] | **SQL Queries** |
| 21]  –  24] | **Python – SQL Connectivity** |

**PROGRAMS:**

1. **Write a menu driven program using functions, to find:**
2. **Factorial of a number**

**The factorial of a number is the product of all the integers from 1 to n.**

**Example,**

**The factorial of 6 is 1\* 2\* 3\* 4\* 5\* 6 = 720.**

**Factorial is not defined for negative numbers, and the factorial of zero is one, 0! is 1.**

1. **Fibonacci series up to nth term**

**It’s a unique sequence where the next number is the sum of the previous two numbers.  
Where the first two terms are always 0 and 1.**

**The series Looks like : 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144 …**

1. **Check if a given number is Palindrome or not.**

**A palindrome is nothing but any number or a string which remains unaltered when reversed.**

**Example: 12321  
Output: Yes, a Palindrome number**

def factorial(n):

if n < 0:

return "negative"

elif n == 0:

return 1

else:

f = 1

for i in range(1, n + 1):

f \*= i

return f

def fibonacci(n):

if n==0:

return []

elif n == 1:

return [0]

else:

f\_series = [0, 1]

while len(f\_series) <n:

m = f\_series[-1] + f\_series[-2]

f\_series.append(m)

return f\_series

def palindrome(n):

if str(n) == str(n)[::-1]:

return True

else:

return False

ans='y'

while ans=='y':

print("Main Menu:")

print("1. Find factorial of a number")

print("2. Generate Fibonacci series up to n terms")

print("3. Check if a number is palindrome")

c = int(input("Enter your choice: "))

if c == 1:

r = int(input("Enter a number: "))

f= factorial(r)

if f =='negative':

print("Factorial is not defined for negative numbers.")

else:

print("The factorial of",r,"is" ,f)

elif c== 2:

r = int(input("Enter the number of terms: "))

s= fibonacci(r)

for i in s:

print(i, end=" ")

print(' ')

elif c == 3:

r = int(input("Enter a number: "))

if palindrome(r)== True:

print(r," is a palindrome number")

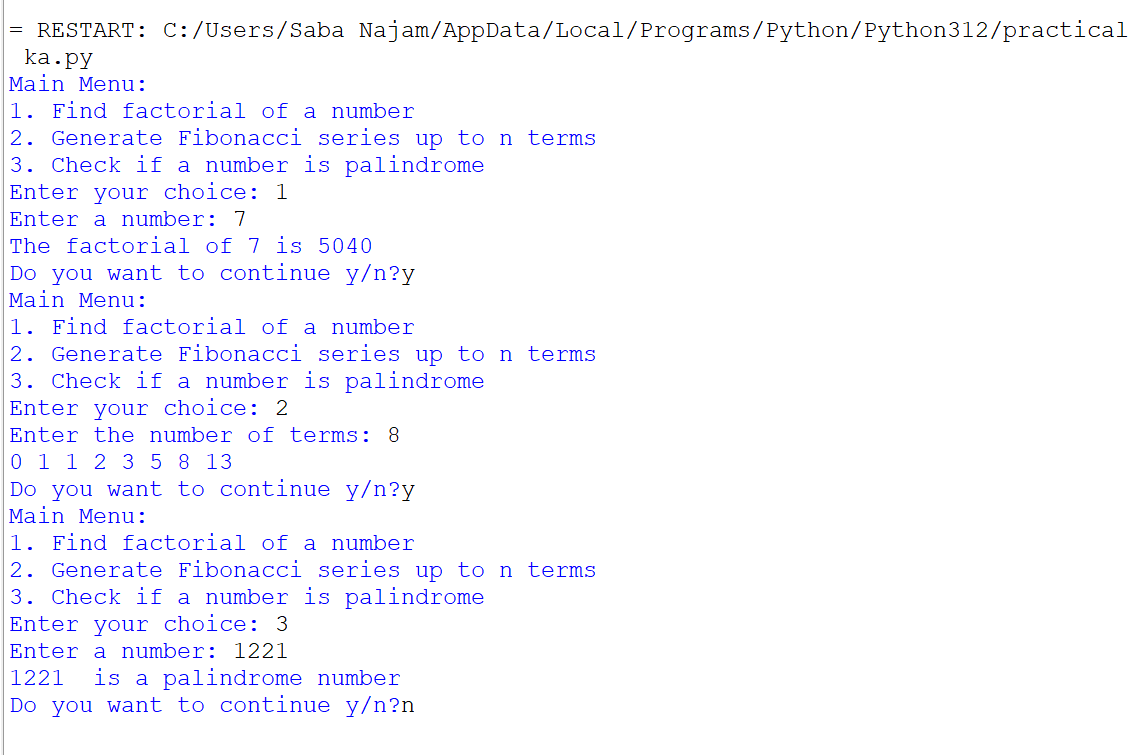
else:

print(r," is not a palindrome number")

else:

print("Invalid choice. Please try again.")

ans=input('Do you want to continue y/n?')

****

**2.Write a menu driven program using functions, to find:**

1. **Check a given number is Armstrong number or not**

**A positive integer is called an Armstrong number of order n if**

**abcd….. = an + bn + cn + dn …………**

**In case of an Armstrong number of 3 digits, the sum of cubes of each digit is equal to the number itself.**

**Example: 153 = 1\*1\*1 + 5\*5\*5 + 3\*3\*3**

**153 is an Armstrong number.**

1. **To display Floyd's triangle**

**Eg:**

**Enter the number of Rows: 5**

**Floyd's Triangle:**

**1**

**2        3**

**4        5         6**

**7        8         9         10**

1. **12       13       14       15**

def armstrong(n):

p = n

l = len(str(p))

a= 0

while p > 0:

q= p % 10

a += q\*\*l

p //= 10

if a== n:

return True

else:

return False

def floydstriangle(n):

r= 1

for i in range(1, n + 1):

for j in range(1, i + 1):

print(r, end=" ")

r += 1

print()

print(" ")

ans='y'

while ans=='y':

print("Menu:")

print("1. Check if a number is an Armstrong number")

print("2. Display Floyd's triangle")

c = int(input("Enter your choice: "))

if c== 1:

s= int(input("Enter a number: "))

if armstrong(s):

print(s, "is an Armstrong number")

else:

print(s, "is not an Armstrong number")

elif c == 2:

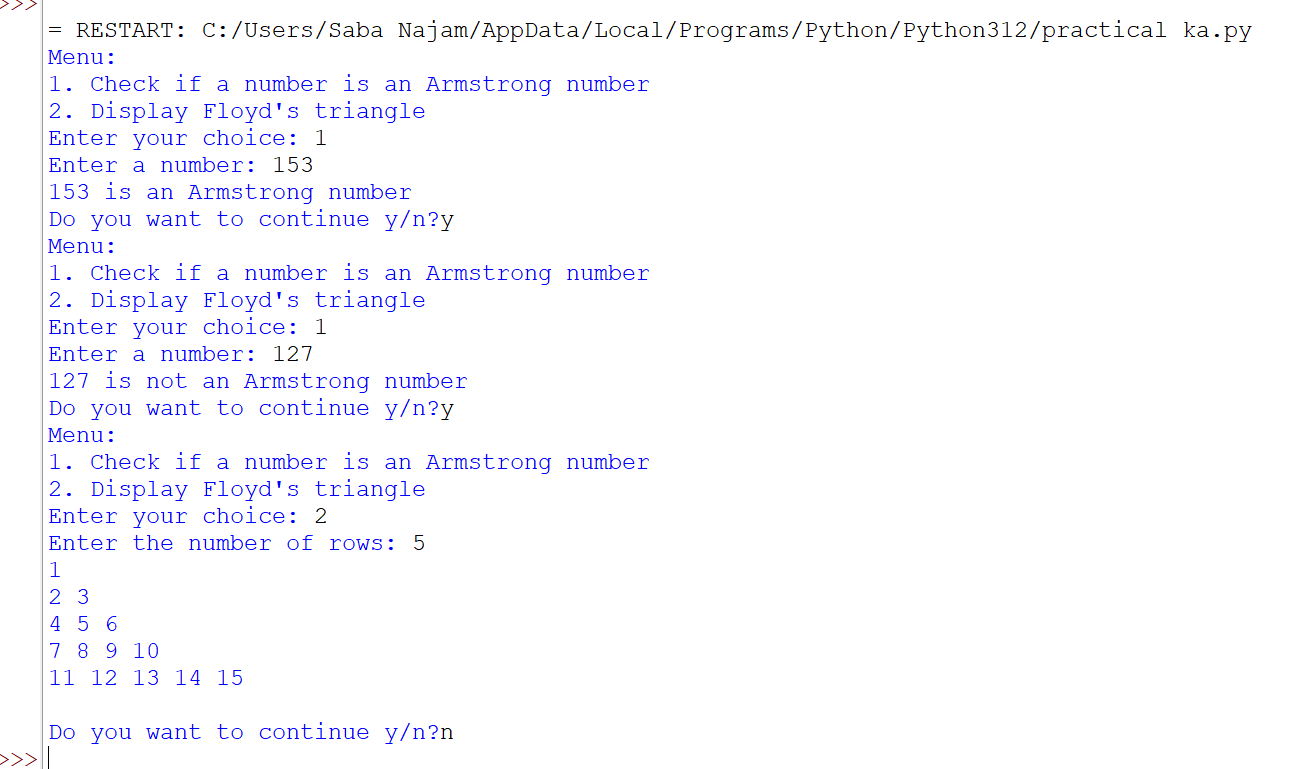
s= int(input("Enter the number of rows: "))

floydstriangle(s)

else:

print("Invalid choice. Please enter a valid choice.")

ans=input('Do you want to continue y/n?')



**3. Write a menu driven program for the following:**

1. **Define a function MSEARCH(STATES) to display all the names from the list STATES, which are starting with the alphabet M.**

**For eg. If list STATES contains**

**L = ["MP", "UP", "DL", "UK", "KA", “MH”]**

**The output should be:**

**[“MP”, “MH”]**

1. **A function with the following signature:**

**remove\_letter(sentence, letter)**

**This function should take a string & a letter as arguments, returning a copy of that string with**

**every instance of the indicated letter removed.**

def M\_states(l):

s = []

for i in l:

if i[0] == 'M':

s.append(i)

return s

def removeletter(s, l):

n = s.replace(l, "")

return n

def count(s):

r= s.split()

return len(r)

ans='y'

while ans=='y':

print("Menu:")

print("1. Search for states starting with 'M'")

print("2. Remove a letter from a string")

print("3. Count words in a string")

c = int(input("Enter your choice : "))

if c == 1:

l = eval(input("Enter a list of states: "))

s = M\_states(l)

print("States starting with 'M' are:", s)

elif c == 2:

s = input("Enter a sentence: ")

l = input("Enter a letter to remove: ")

n= removeletter(s, l)

print("New sentence:",n)

elif c == 3:

s = input("Enter a sentence: ")

r = count(s)

print("Number of words in the given sentence:", r)

else:

print("Invalid choice, please try again.")

ans=input('Do you want to continue y/n?')



**4. Write a menu driven program in Python using user defined functions that take a list as parameter and return (Do not use built-in functions):**

1. **Maximum of the element**
2. **Minimum of the element**
3. **Sum of the elements.**

def maximum(l):

max = l[0]

for i in l:

if i > max:

max = i

return max

def minimum(l):

min = l[0]

for i in l:

if i < min:

min =i

return min

def sum(l):

total = 0

for i in l:

total+=i

return total

ans='y'

while ans=='y':

print("Main Menu:")

print("1. Find the maximum element")

print("2. Find the minimum element")

print("3. Find the sum of elements")

c = int(input("Enter your choice: "))

if c == 1:

l= eval(input("Enter a list of numbers: "))

max = maximum(l)

print("Maximum value:", max)

elif c == 2:

l = eval(input("Enter a list of numbers: "))

min = minimum(l)

print("Minimum value:", min)

elif c == 3:

l = eval(input("Enter a list of numbers: "))

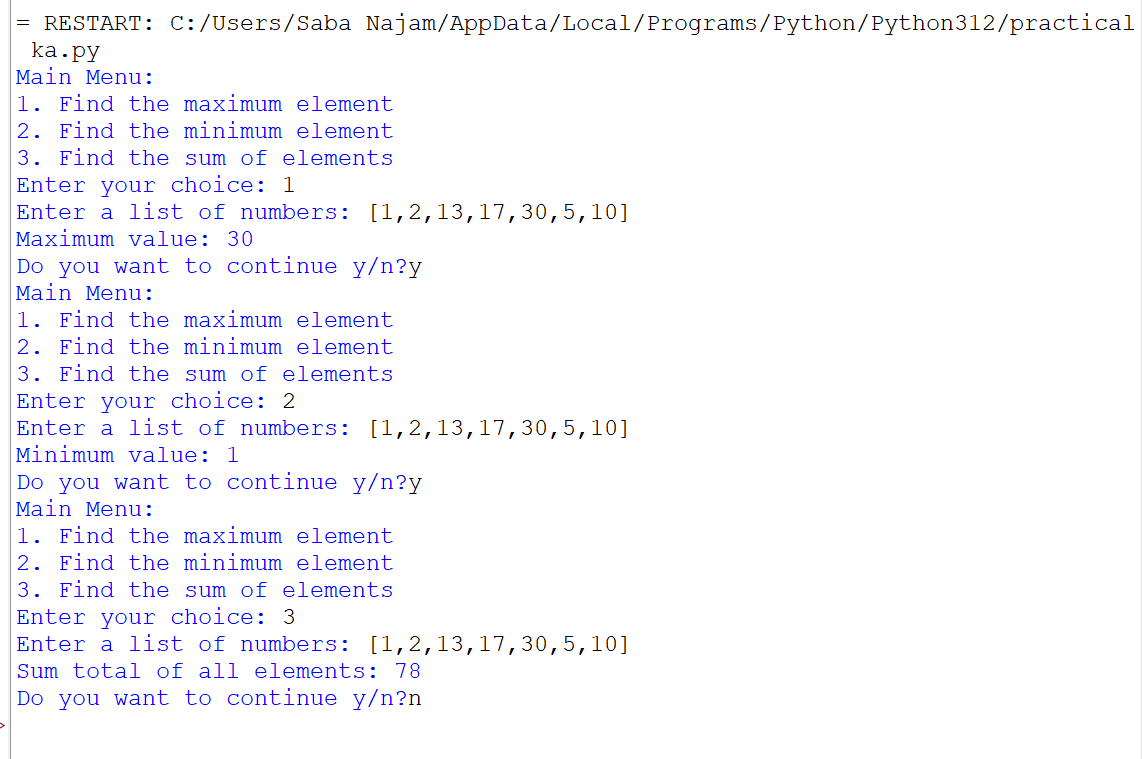
s = sum(l)

print("Sum total of all elements:", s)

else:

print("Invalid choice, please try again.")

ans=input('Do you want to continue y/n?')



**5.Write a menu driven program in Python using user defined functions for the following:**

1. **To read and display the text file content line by line with each word separated by “$”.**
2. **To remove all the lines that contain the character ‘a’ in a file and write it to another file.**

def display(fn):

try:

f = open(fn, 'r')

for i in f.readlines():

for j in i.split():

print(j, end='$')

print()

f.close()

except:

print('File not found')

def rl(f1, f2):

try:

a=open(f1,'r')

b=open(f2,'w')

for i in f1:

if 'a' not in i:

f2.write(i)

print('Successfully copied')

except FileNotFoundError:

print("File", f1, "not found.")

ans='y'

while ans=='y':

print("Main Menu:")

print("1. Read and display the file content with words separated by '$'")

print("2. Remove lines that contain 'a' and write to another file")

c = int(input("Enter your choice: "))

if c == 1:

fn = input("Enter the file name: ")

display(fn)

elif c == 2:

f1= input("Enter the file name to read from: ")

f2= input("Enter the file name to write on: ")

rl(f1,f2)

else:

print("Invalid choice, please try again.")

ans=input('do you want to continue y/n?')



**6.Write a menu driven program with the following functions:**

1. **CreateTextFile( ) - create a text file "data.txt" with several lines of text**
2. **CopyVowelWord( ) - create another text file "vowel.txt" which will store all the words starting with vowel from "data.txt".**
3. **Read & display the contents of both the files.**
4. **Display the total number of words starting with vowels.**

def createtextfile():

try:

with open("data.txt", "w") as f:

t=input('Enter text to write onto file')

f.write(t)

print("File created successfully.")

except:

print("Error creating file.")

def copyvowelword():

l = []

try:

with open("data.txt", "r") as f1, open("vowel.txt", "w") as f2:

for i in f1:

n = i.split()

for j in n:

if j[0].lower() in "aeiou":

f2.write(j+ "\n")

l.append(j)

print("Vowel words copied successfully.")

return l

except FileNotFoundError:

print("File not found.")

return False

def display():

try:

with open("data.txt", "r") as f1, open("vowel.txt", "r") as f2:

print("Contents of 'data.txt':")

print(f1.read())

print("Contents of 'vowel.txt':")

print(f2.read())

except FileNotFoundError:

print("File not found.")

def vowel(l):

print("Total number of words starting with a vowel:", len(l))

l = []

ans='y'

while ans=='y':

print("main menu:")

print("1. Create a text file with several lines of text")

print("2. Copy words starting with vowels to another file")

print("3. Read and display the contents of both files")

print("4. Display the total number of words starting with a vowel")

c= int(input("Enter your choice: "))

if c == 1:

createtextfile()

elif c == 2:

l= copyvowelword()

elif c == 3:

display()

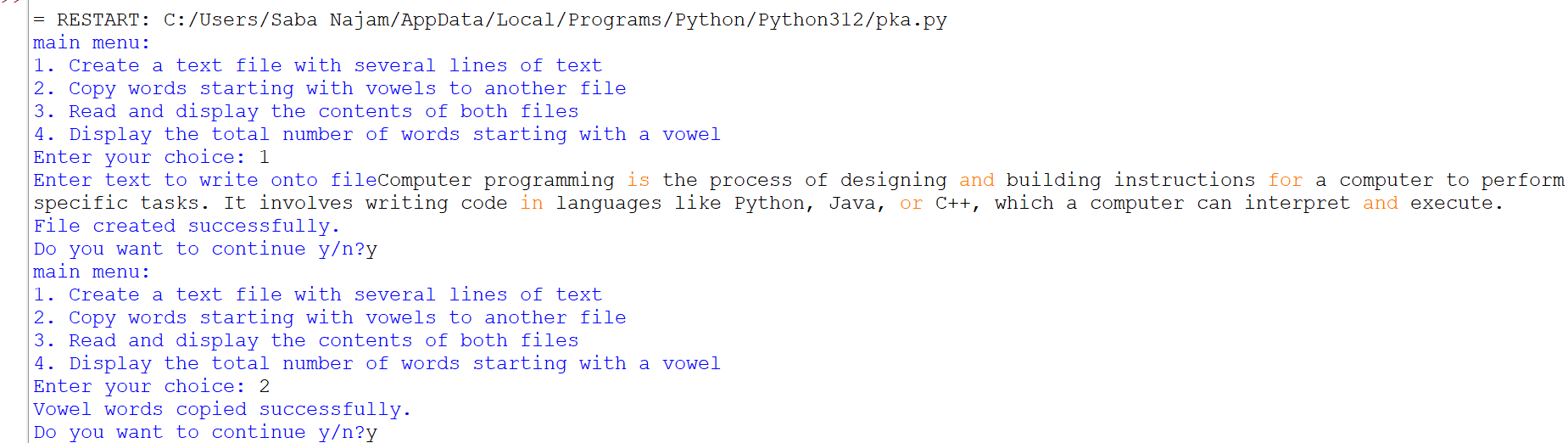
elif c == 4:

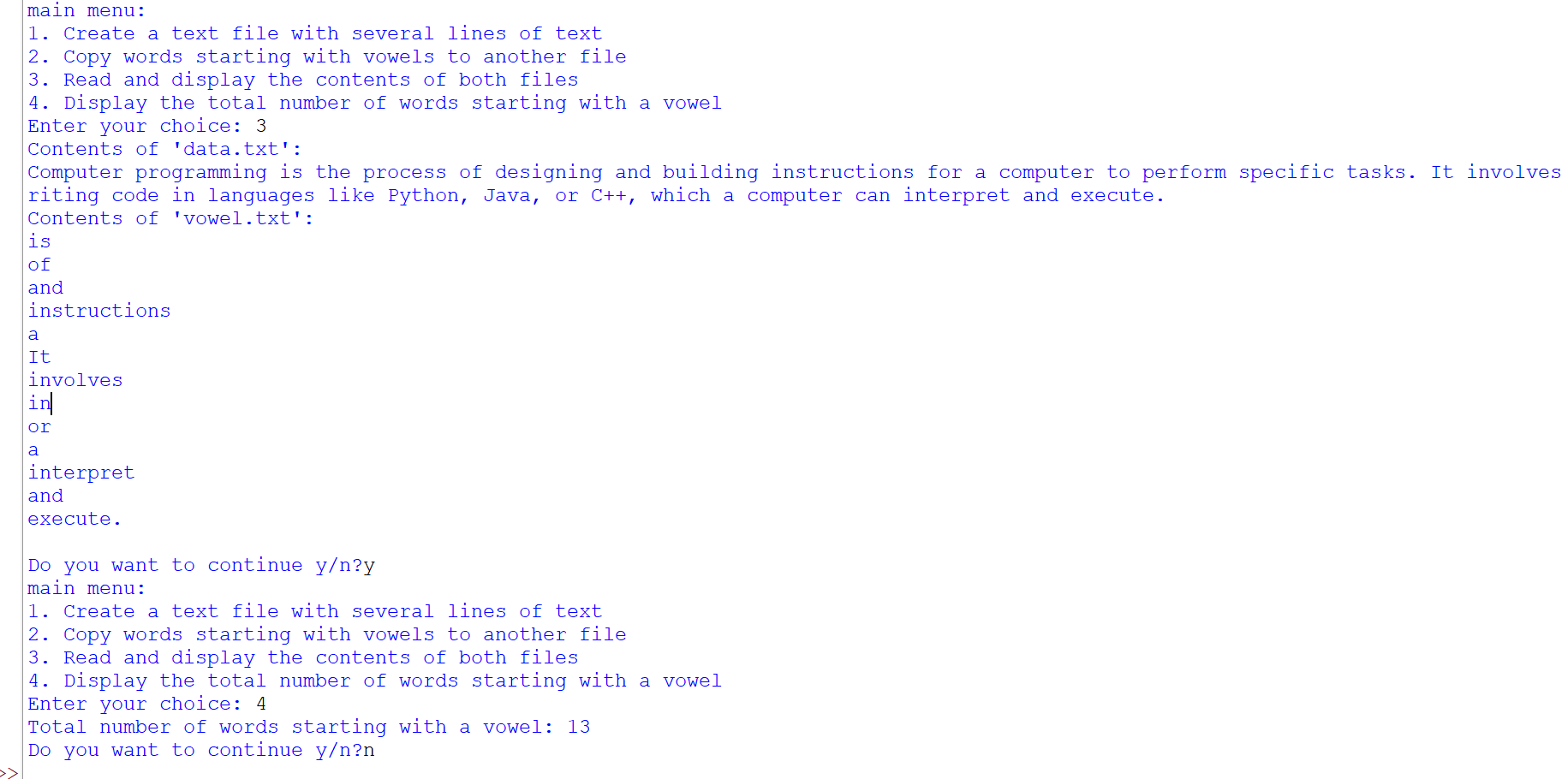
vowel(l)

else:

print("Invalid choice, please try again.")

ans=input('Do you want to continue y/n?')

****

****

**7.Write a menu driven program with the following functions:**

1. **CreateTextFile( ) - create a text file " content.txt" with few lines.**
2. **CountAll( ) - count number of lines, consonants, digits, spaces & words.**
3. **ReplaceSpace( ) - create another file called "wspace.txt" using the original which will contain the text after replacing all the blank spaces with '#'.**
4. **Read & display the contents of both the files.**

def createtextfile():

with open('content.txt', 'w') as f:

t=input('Enter text to write onto file')

f.write(t)

print('text successfully written onto file')

def countAll():

with open('content.txt', 'r') as f:

n = f.readlines()

r = len(n)

c = 0

d = 0

s = 0

w = 0

for i in n:

for j in i:

if j.isalpha() and j.lower() not in 'aeiou':

c+= 1

elif j.isdigit():

d += 1

elif j.isspace():

s+= 1

w+= len(i.split())

print('Number of lines:', r)

print('Number of consonants:', c)

print('Number of digits:', d)

print('Number of spaces:', s)

print('Number of words:', w)

def replacespace():

with open('content.txt', 'r') as f1, open('wspace.txt', 'w') as f2:

for i in f1:

s=i.replace(' ', '#')

f2.write(s)

print('replaced successfully')

def rd():

with open('content.txt', 'r') as f1, open('wspace.txt', 'r') as f2:

print('Contents of content.txt:')

print(f1.read())

print('Contents of wspace.txt:')

print(f2.read())

ans='y'

while ans=='y':

print('Main Menu:')

print('1. Create text file')

print('2. Count all')

print('3. Replace spaces')

print('4. Read files')

c = input('Enter your choice: ')

if c == '1':

createtextfile()

elif c == '2':

countAll()

elif c == '3':

replacespace()

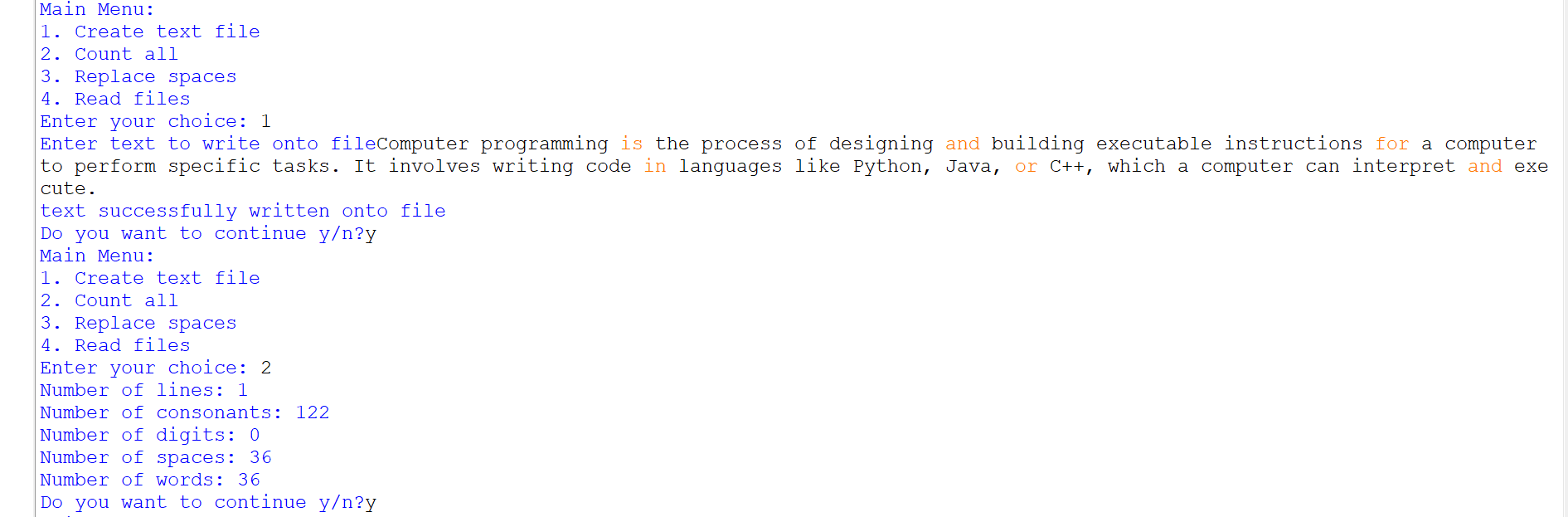
elif c == '4':

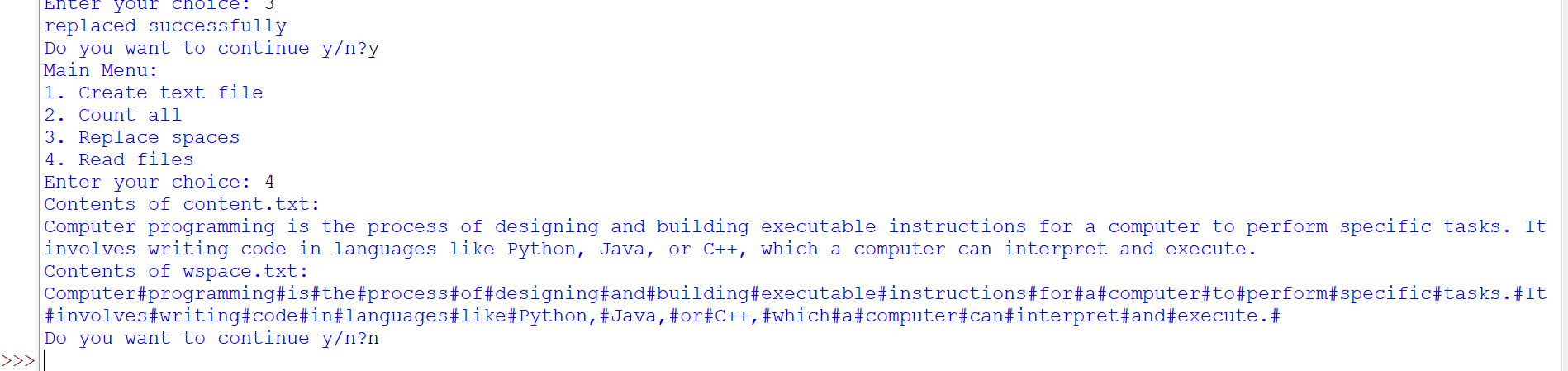
rd()

else:

print('Invalid choice. Please try again.')

ans=input('Do you want to continue y/n?')





**8.Write a menu driven program with the following functions:**

1. **DISPLAYWORDS( ) to read lines from text file STORY.TXT , and display those words which are less than 4 characters.**
2. **SEARCHWORD( ) to search a word and its frequency in a text file.**

def displaywords():

with open('STORY.txt', 'r') as f:

l = f.readlines()

for i in l:

for j in i.split():

if len(j)<4:

print(j)

def searchword():

fn=input('enter file name to search')

w= input('Enter a word to search: ')

n= 0

with open(fn, 'r') as f:

r = f.read()

for i in r.split():

if i.lower() == w:

n+=1

if n == 0:

print("Word not found")

else:

print("Word found", n, "times")

ans='y'

while ans=='y':

print('Main Menu:')

print('1. Display words less than 4 characters')

print('2. Search word frequency')

c = input('Enter your choice: ')

if c == '1':

displaywords()

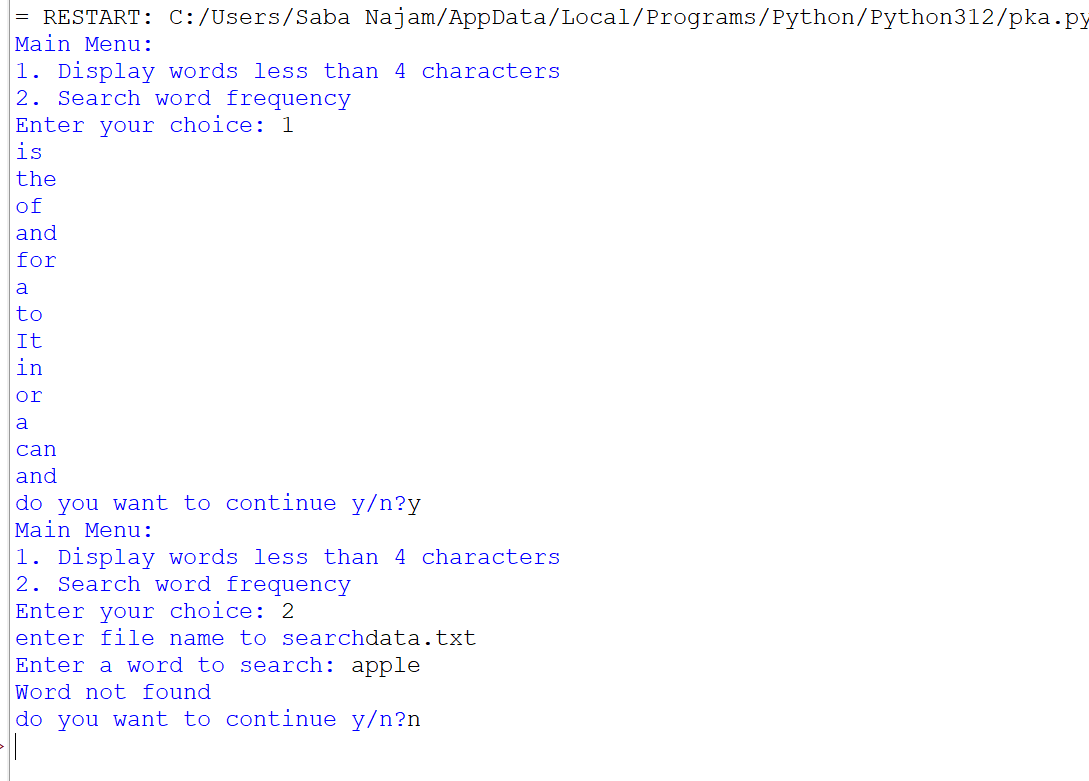
elif c == '2':

searchword()

else:

print('Invalid choice. Please try again.')

ans=input('do you want to continue y/n?')



**9.Write a menu driven program in Python using Pickle library and**

1. **Create a binary file “STUDENT.DAT” with following structure:**

* **Admission\_number**
* **Student Name**
* **Percentage**

1. **Write a function countrec( ) in Python that would read contents of the file “STUDENT.DAT” and display the details of those students whose percentage is above 75. Also display the number of students scoring above 75%.**

import pickle

def create():

ans='y'

while ans=='y':

an = input('Enter admission number: ')

name = input('Enter student name: ')

p= float(input('Enter percentage: '))

with open('STUDENT.DAT', 'ab') as f:

r = [an,name, p]

pickle.dump(r, f)

ans=input('do you want to add more records? y/n')

def countrec():

c = 0

with open('STUDENT.DAT', 'rb') as f:

while True:

try:

r = pickle.load(f)

if r[2] > 75:

print('Admission number:', r[0])

print('Name:', r[1])

print('Percentage:', r[2])

c+= 1

except EOFError:

break

print('Number of students scoring above 75%:', c)

ans='y'

while ans=='y':

print('Main Menu:')

print('1. Create student record')

print('2. Count records with percentage above 75')

c = input('Enter your choice: ')

if c == '1':

create()

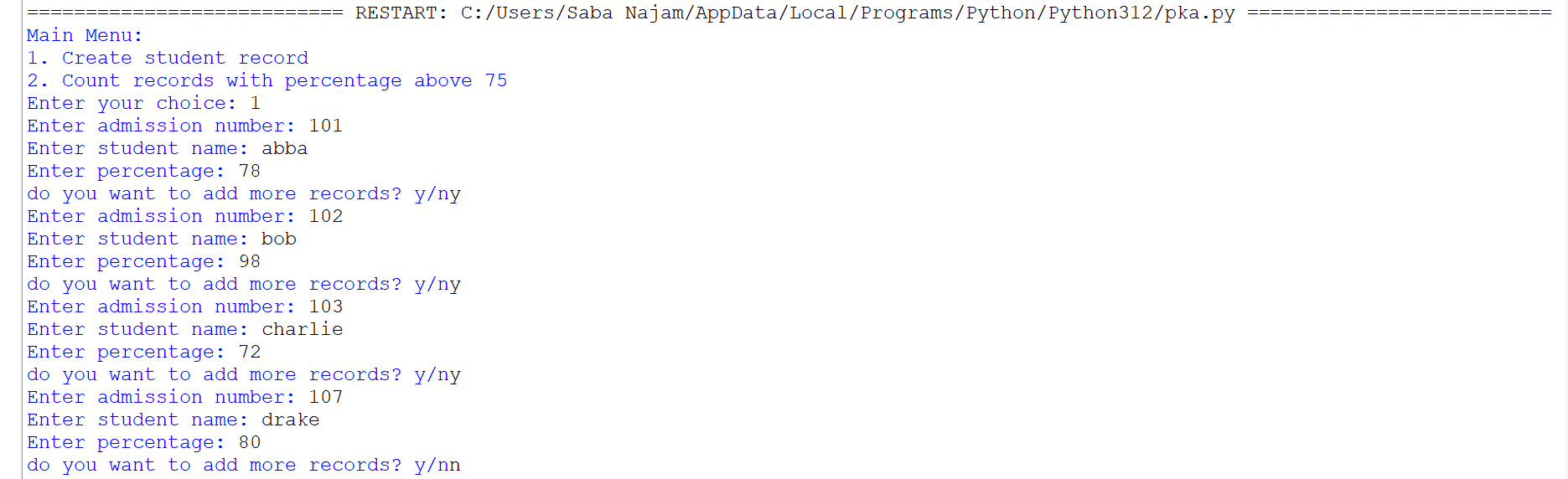
elif c == '2':

countrec()

else:

print('Invalid choice.')

ans=input('Do you want to continue')





**10.Write a menu driven program in Python using Pickle library and**

1. **Create a binary file with following structure:**

* **CompanyID**
* **Company name**
* **Turnover**

1. **Display the contents of the binary file.**
2. **Display the Company whose turnover is above user given value.**
3. **Search a Company by Company ID given by the user.**

import pickle

def create():

ans='y'

while ans=='y':

id = input('Enter company ID: ')

name = input('Enter company name: ')

to = float(input('Enter turnover: '))

with open('COMPANY.DAT', 'ab') as f:

r = [id, name,to]

pickle.dump(r, f)

ans=input('Do you want to add more records?y/n')

def display():

f=open('COMPANY.DAT', 'rb')

while True:

try:

r= pickle.load(f)

print('Company ID:', r[0])

print('Name:', r[1])

print('Turnover:', r[2])

print()

except:

f.close()

break

def to():

x = float(input('Enter turnover value '))

found = False

f=open('COMPANY.DAT', 'rb')

while True:

try:

r = pickle.load(f)

if r[2] > x:

print('Company ID:', r[0])

print('Name:', r[1])

print('Turnover:', r[2])

print()

found = True

except:

f.close()

break

def search():

s = input('Enter company ID: ')

found = False

f= open('COMPANY.DAT', 'rb')

while True:

try:

r = pickle.load(f)

if r[0] == s:

print('Company ID:', r[0])

print('Name:', r[1])

print('Turnover:', r[2])

print()

found = True

except:

f.close()

break

ans='y'

while ans=='y':

print('Main Menu:')

print('1. Create company record')

print('2. Display all records')

print('3. Search company by turnover')

print('4. Search company by ID')

c= input('Enter your choice: ')

if c == '1':

create()

elif c== '2':

display()

elif c == '3':

to()

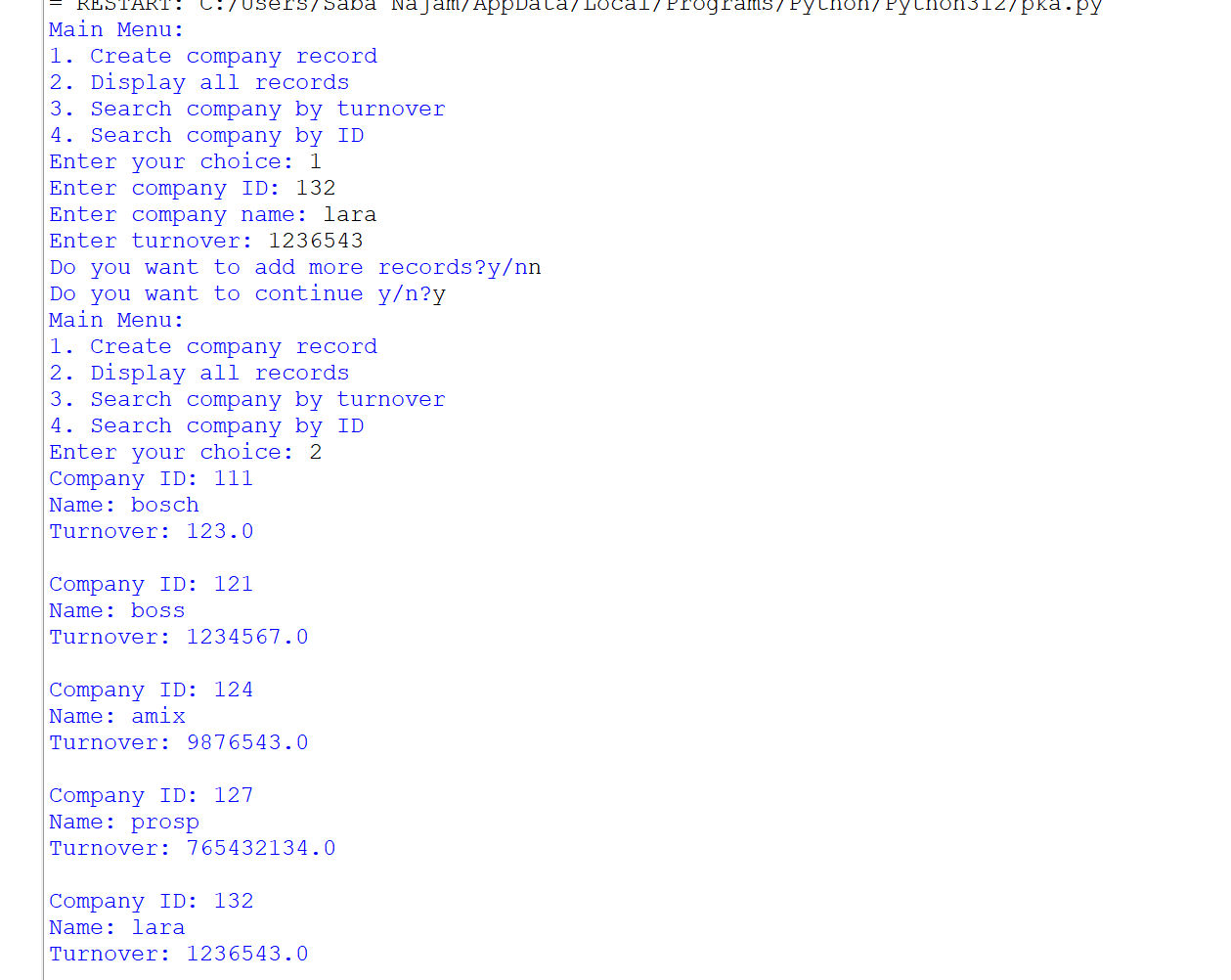
elif c== '4':

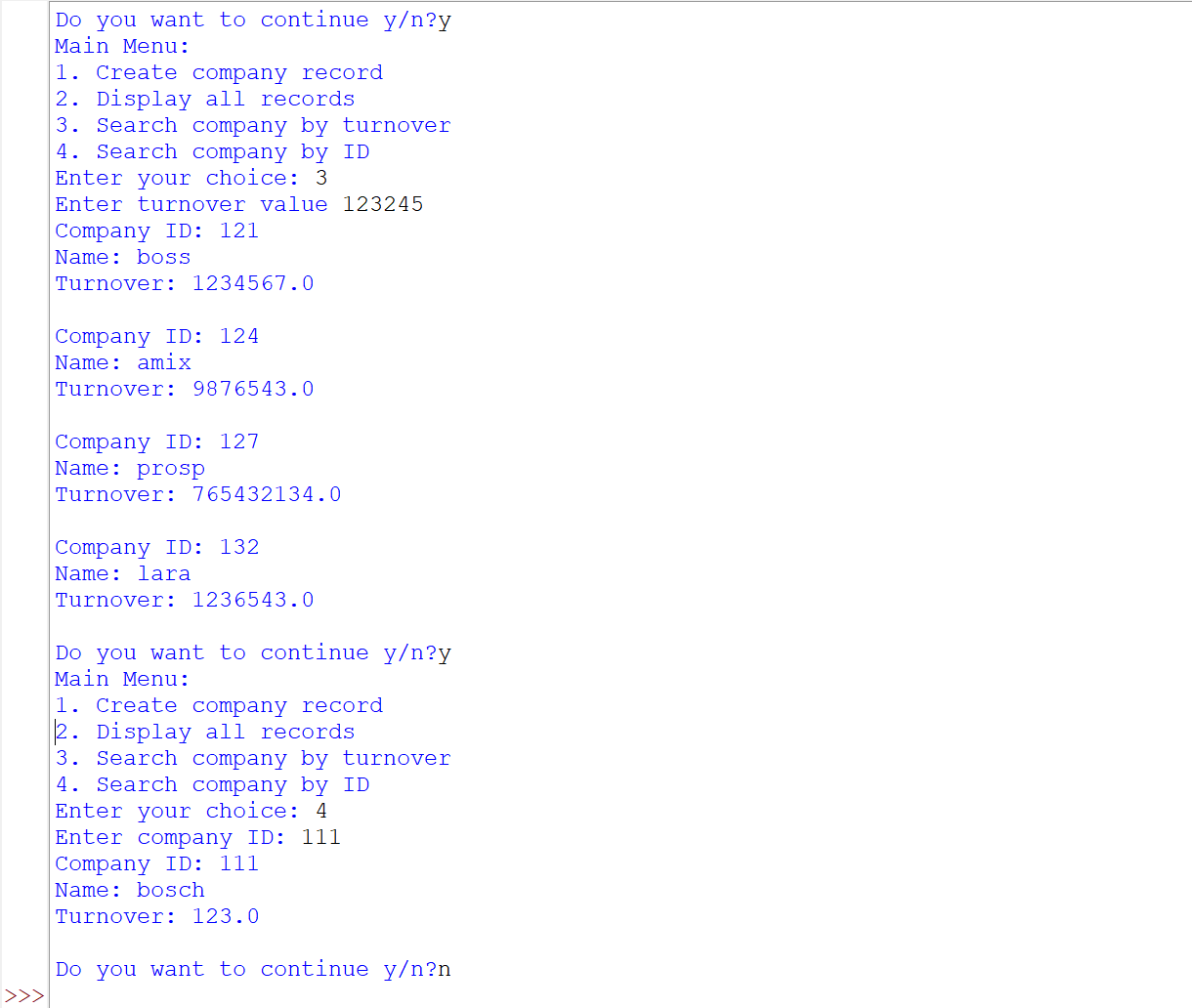
search()

else:

print('Invalid choice')

ans=input('Do you want to continue y/n?')





**11.Write a menu driven program in Python using Pickle library and**

1. **Create binary file with following structure:**

* **BookId**
* **Book Name**
* **Price**

1. **Append data to the file.**
2. **Update the price based on BookId.**
3. **Display the contents of the binary file.**

import pickle

def Create():

n= input("Enter name of file : ")

f = open(n, 'wb')

pickle. dump(['Book ID','BookName','price'],f)

print('file created successfully')

f.close()

def Append():

n= input("Enter name of file : ")

f= open(n, 'ab')

ans= 'y'

while ans=='y':

l=[]

id = int(input("Enter Book ID : "))

name = input("Enter name of book : ")

pr = int(input("Enter price of book : "))

r=[id,name,pr]

pickle.dump(r,f)

ans= input("Do you want to add more records? y/n : ")

f.close()

def Update():

n= input("Enter name of the file : ")

f= open(n, 'rb+')

p= int(input("Enter ID of book whose price is to be updated : "))

s= float(input("Enter new price : "))

found = False

try:

while True:

pos = f.tell()

r = pickle.load(f)

if r[0] == p:

f.seek(pos)

r[2] = s

pickle.dump(r, f)

found= True

except:

if found == True:

print("Successfully updated")

else:

print("Book not found")

f.close()

def Display():

n = input("Enter name of the file : ")

f = open(n, 'rb')

try:

while True:

r= pickle.load(f)

print(r)

except :

f.close()

ans='y'

while ans=='y':

print('Main Menu:')

print('1. Create file')

print('2. Append record')

print('3. Update record')

print('4. Display records')

c = int(input("Enter your choice : "))

if c== 1:

Create()

elif c == 2:

Append()

elif c == 3:

Update()

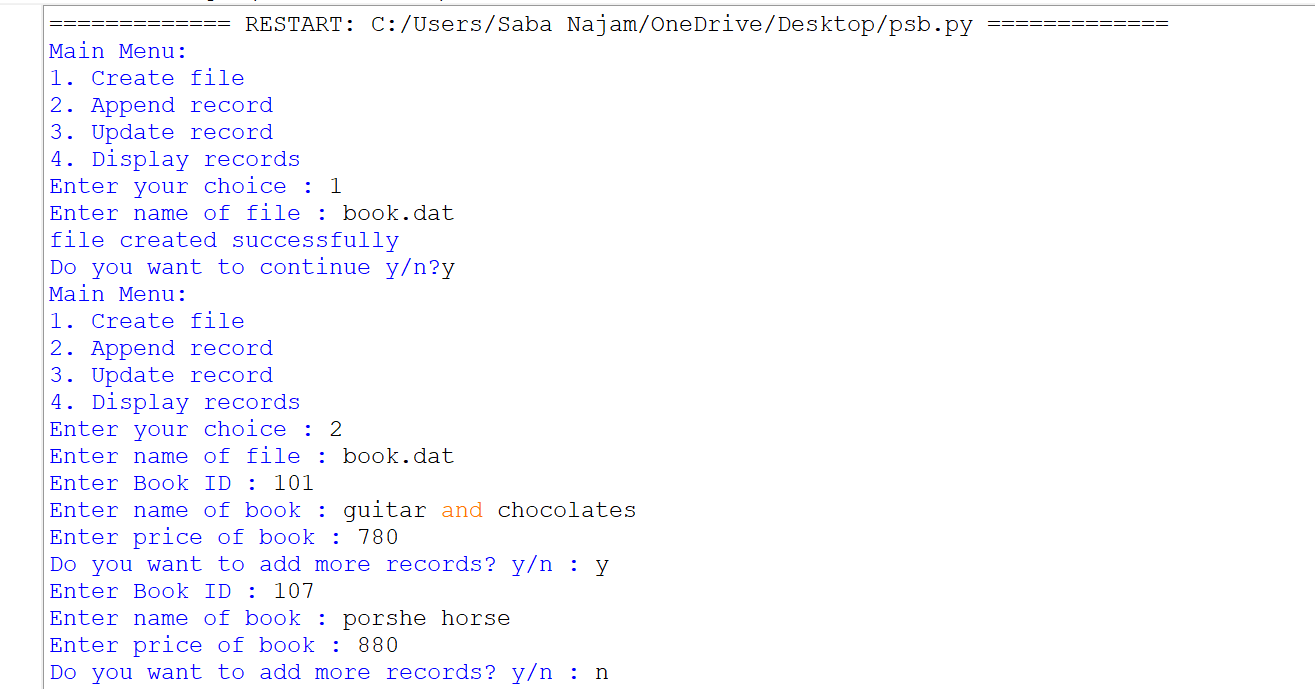
elif c == 4:

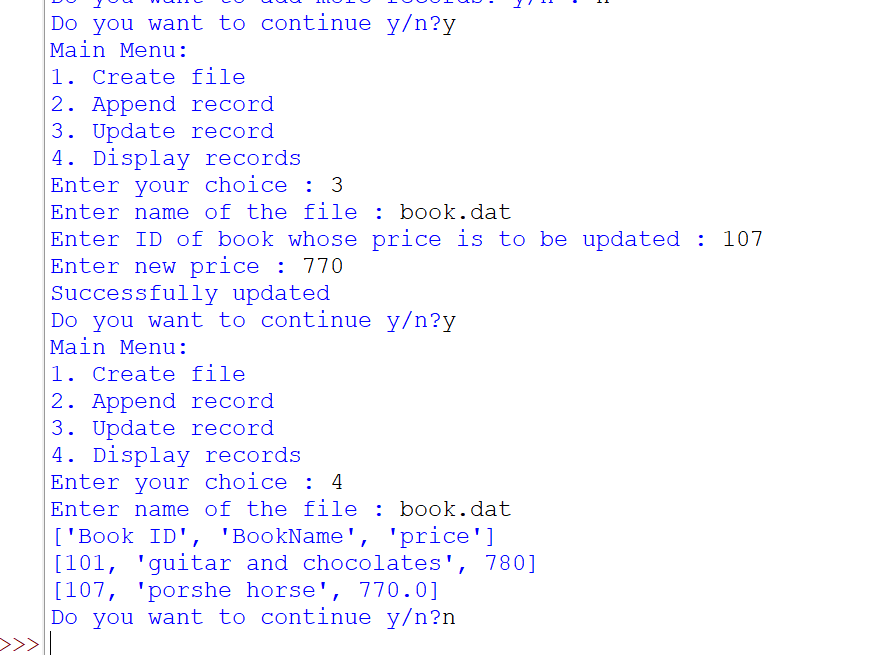
Display()

else:

print("Invalid Choice")

ans = input('Do you want to continue y/n?')

****

****

**12.Write a menu driven program in Python:**

1. **Define a function to write the following data into a CSV file:**

* **Roll no**
* **Name of student**
* **Mark in Sub1**
* **Mark in sub2**
* **Mark in sub3**
* **Mark in sub4**
* **Mark in sub5**

**Perform following operations on the CSV file:**

1. **Define a function to read the CSV file and calculate total and percentage for each student.**
2. **Define a function to display the name of student if in any subject marks are greater than 80% (Assume marks are out of 100)**

import csv

def write():

n= input("Enter name of file ")

f = open(n, 'w')

ans= 'y'

w= csv.writer(f)

while ans == 'y':

r = int(input("Enter roll number of student : "))

name = input("Enter name of student : ")

M1 = int(input("Enter marks in Subject 1 : "))

M2 = int(input("Enter marks in Subject 2 : "))

M3 = int(input("Enter marks in Subject 3 : "))

M4 = int(input("Enter marks in Subject 4 : "))

M5 = int(input("Enter marks in Subject 5 : "))

l = [r, name, M1, M2, M3, M4, M5]

w.writerow(l)

ans = input("Do you want to add more records? y/n : ")

f.close()

def read():

n=input('enter name of file to read data')

f= open(n, 'r',newline='\r\n')

r = csv.reader(f)

next(r)

for i in r:

print(i)

t = int(i[2])+int(i[3])+int(i[4])+int(i[5])+int(i[6])

print("Total marks : ", t)

p = t/5

print("Percentage : ", p)

break

f.close()

def display():

n=input('enter the name of file')

f=open(n, 'r')

r = csv.reader(f)

next(r)

for i in r:

for j in i[2:]:

if int(j)>80:

print(i[1])

break

f.close()

ans='y'

while ans=='y':

print('Main Menu:')

print('1. Write data to CSV file')

print('2. Read data from CSV file and calculate total and percentage for each student')

print('3. Display names of students who scored more than 80 in any subject')

c = input('Enter your choice: ')

if c == '1':

write()

elif c == '2':

read()

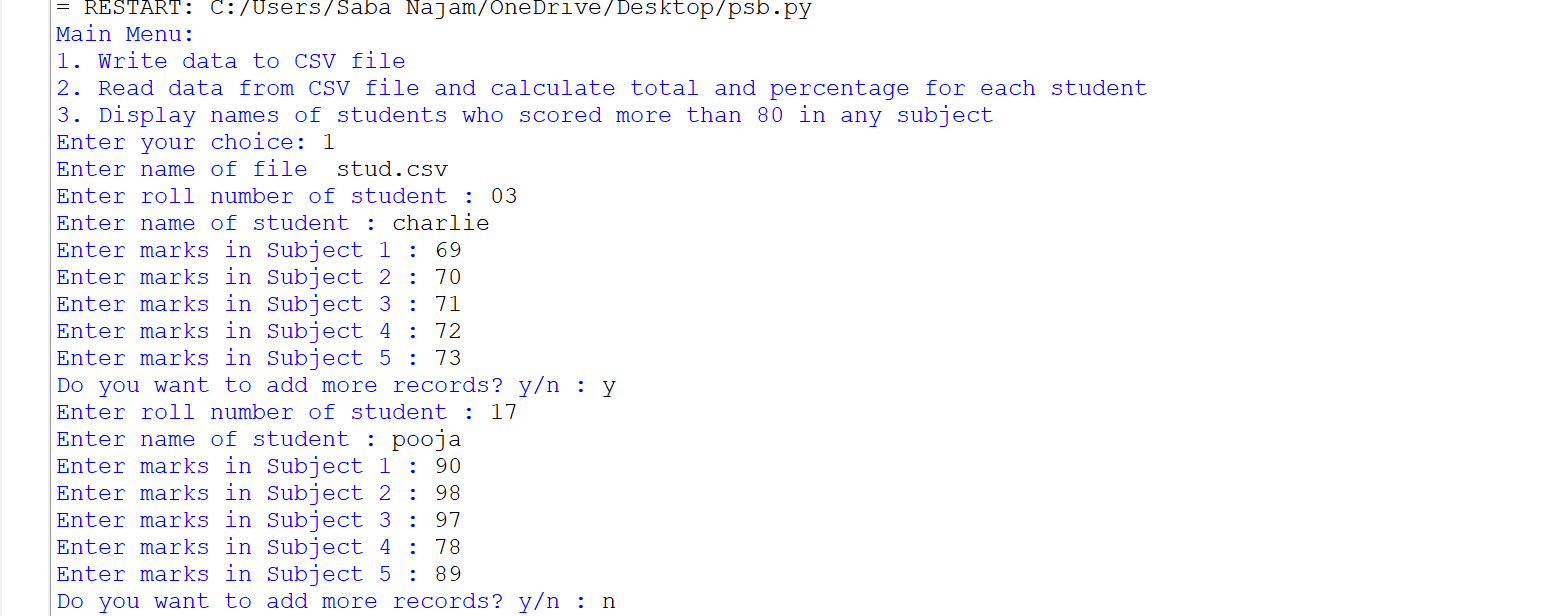
elif c == '3':

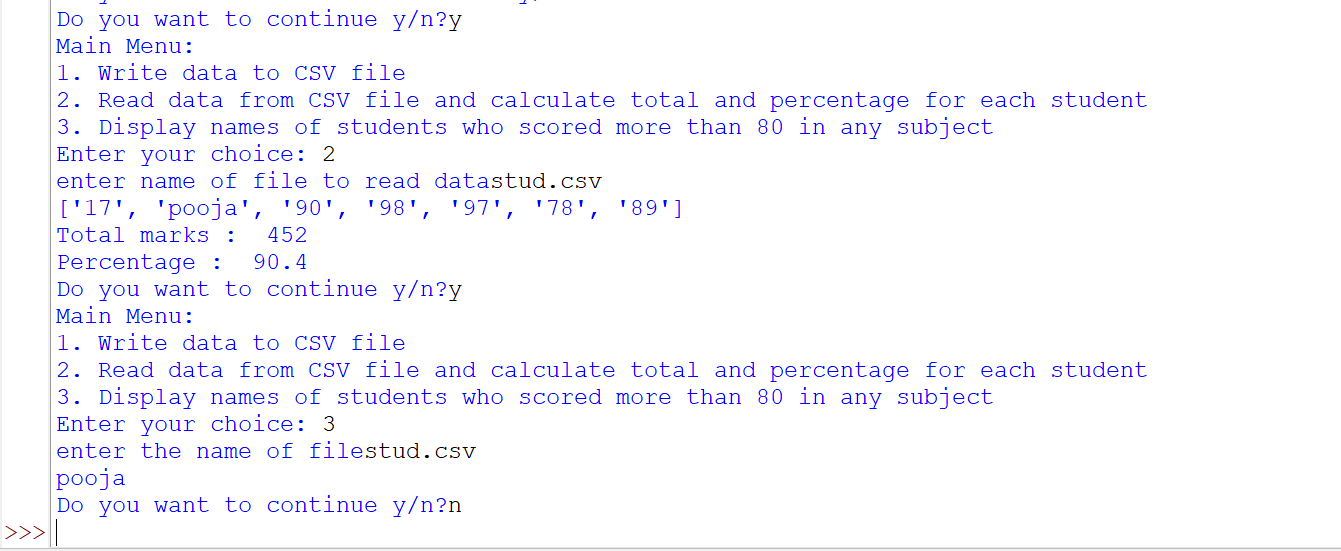
display()

else:

print('Invalid choice.')

ans=input('Do you want to continue y/n?')





**13.Write a menu driven program in Python:**

1. **Define a function to write the following data into a CSV file “emp.CSV”**

* **empid**
* **empName**
* **Salary**
* **Department**

1. **Define a function to search the record from “emp.CSV” file based on empid. If a record is found, display the record, otherwise display appropriate messages.**

import csv

def write():

f= open('emp.csv', 'w')

ans= 'y'

w = csv.writer(f)

while ans == 'y':

id= int(input("Enter employee ID : "))

name = input("Enter name of employee : ")

s= float(input("Enter salary of employee : "))

dep = input("Enter employee department : ")

l = [id, name, s, dep]

w.writerow(l)

ans = input("Do you want to add more records y/n? : ")

f.close()

def search():

n= int(input('Enter employee ID to search: '))

f=open('emp.csv', 'r')

r = csv.reader(f)

next(r)

found = False

for i in r:

if i[0] == n:

print('Employee record found:', i)

found = True

break

else:

print("Record not found")

break

f.close()

ans='y'

while ans=='y':

print('1. Write employee data to CSV file')

print('2. Search employee record by ID')

c = input('Enter your choice: ')

if c == '1':

write()

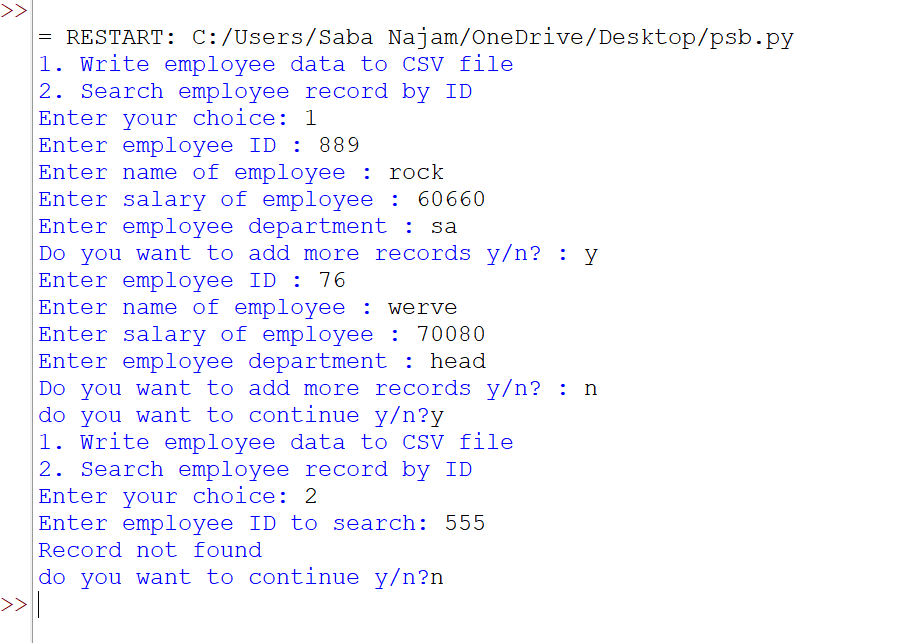
elif c == '2':

search()

else:

print('Invalid choice')

ans=input('do you want to continue y/n?')



**14.A list contains the following record of a Hostel:**

**[Hostel\_No, Total\_Students, Total Rooms]**

**Write a menu driven program, with the following user defined functions to perform given operations on the stack named ‘Hostel’:**

1. **Push\_element( ) - To push an object containing Hostel\_No and Total Students along with Total Rooms to the stack**
2. **Pop\_element( ) - To pop the objects from the stack and display them. Also, display “Stack Empty” when there are no elements in the stack.**
3. **Implement all stack operations.**

def Push(Hostel):

ans= 'y'

while ans== 'y':

hn = int(input("Enter hostel number : "))

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

r= int(input("Enter total number of rooms in hostel : "))

l = [hn, n, r]

Hostel.append(l)

ans= input("Do you want to add more records y/n?: ")

def Pop(Hostel):

if Hostel == []:

print("Stack is Empty")

else:

while len(Hostel)!=0:

print("element :", Hostel.pop())

def Display(Hostel):

print("Hostel Number \t Number Of Students \t Number Of Rooms")

for i in range(len(Hostel)-1, -1, -1):

print(Hostel[i][0], '\t', Hostel[i][1], '\t', Hostel[i][2])

def peek(Hostel):

if Hostel == []:

print("Stack is empty; Underflow")

else:

print('Peek element:',end='')

print(Hostel.pop())

Hostel = eval(input("Enter stack Hostel : "))

ans='y'

while ans=='y':

print('Main menu')

print("1. Push Element to Stack")

print("2. Pop Element from Stack")

print("3. Display Stack")

print("4. display peek element")

c= int(input("Enter your choice : "))

if c== 1:

Push(Hostel)

elif c == 2:

Pop(Hostel)

elif c == 3:

Display(Hostel)

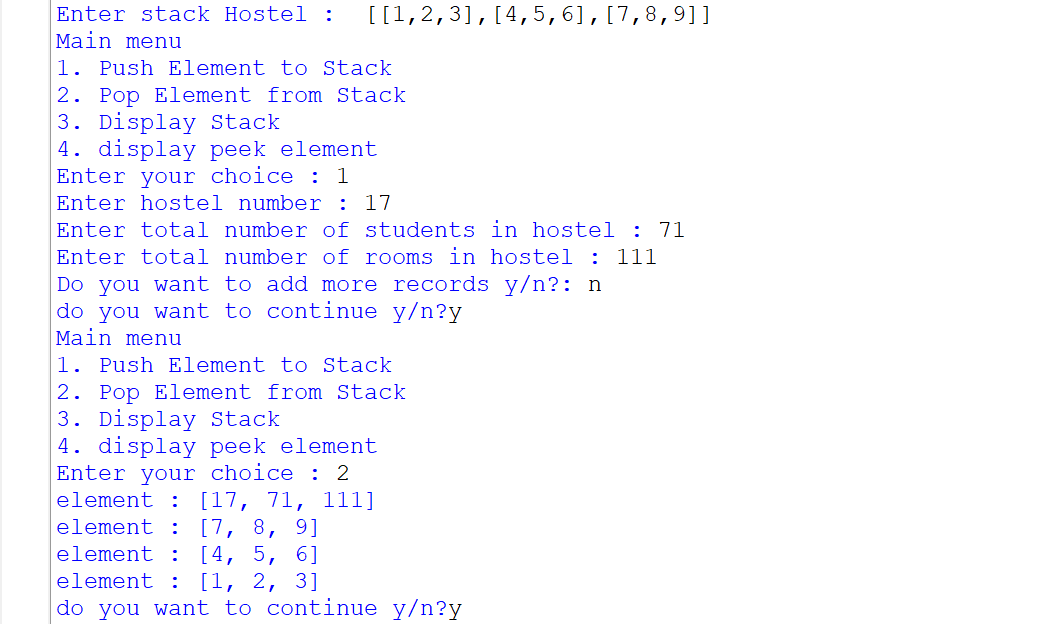
elif c == 4:

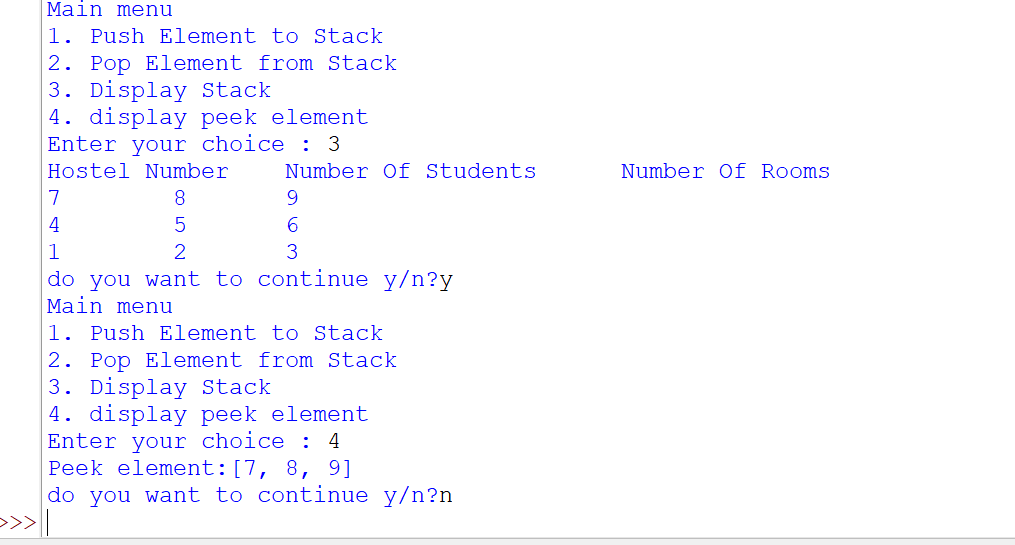
peek(Hostel)

else:

print("Invalid Choice")

ans=input('do you want to continue y/n?')

****

****

**15.Mr. John wants to store book details like book ID and Book Name in a dictionary format. Write a menu driven program, with separate user defined functions to perform the following operations:**

1. **Push the values (Name of the book) of the dictionary into a stack where book names begin with A or C.**
2. **Pop and display the content of the stack.**

**For example, if the content of Dictionary is as follows:**

**Book={"B001":"C++", "B002":"Python", "B003":"Ada", "B004":"C", "B005":"Java", "B006":"Oracle", "B007":"HTMl"}**

**The output of the program should be : C Ada C++**

1. **Implement all the stack operations.**

def Push(Stack,d):

val= list(d.items())

for i in val:

if i[1][0] in ['a','c','A','C']:

Stack.append(i)

print(Stack)

print("Values added to stack")

return Stack

def Pop(Stack):

while Stack!=[]:

print(Stack.pop())

else:

print()

def Display(Stack):

print("Book ID \t Book Name")

for i in range(len(Stack)-1, -1, -1):

print(Stack[i][0], '\t', Stack[i][1])

def peek(Stack):

if Stack==[]:

print("stack is empty")

else:

print(Stack[-1])

Stack=[]

d= eval(input("Enter the dictionary : "))

ans='y'

while ans=='y':

print("""Main Menu:

(1) To push values of the dictionary into a stack where book names begin with A or C

(2) To pop and display the content of the stack

(3) To display all stack elements

(4) To display peek element""")

c = int(input("Enter your choice : "))

if c == 1:

i=Push(Stack,d)

elif c == 2:

Pop(i)

elif c == 3:

Display(Stack)

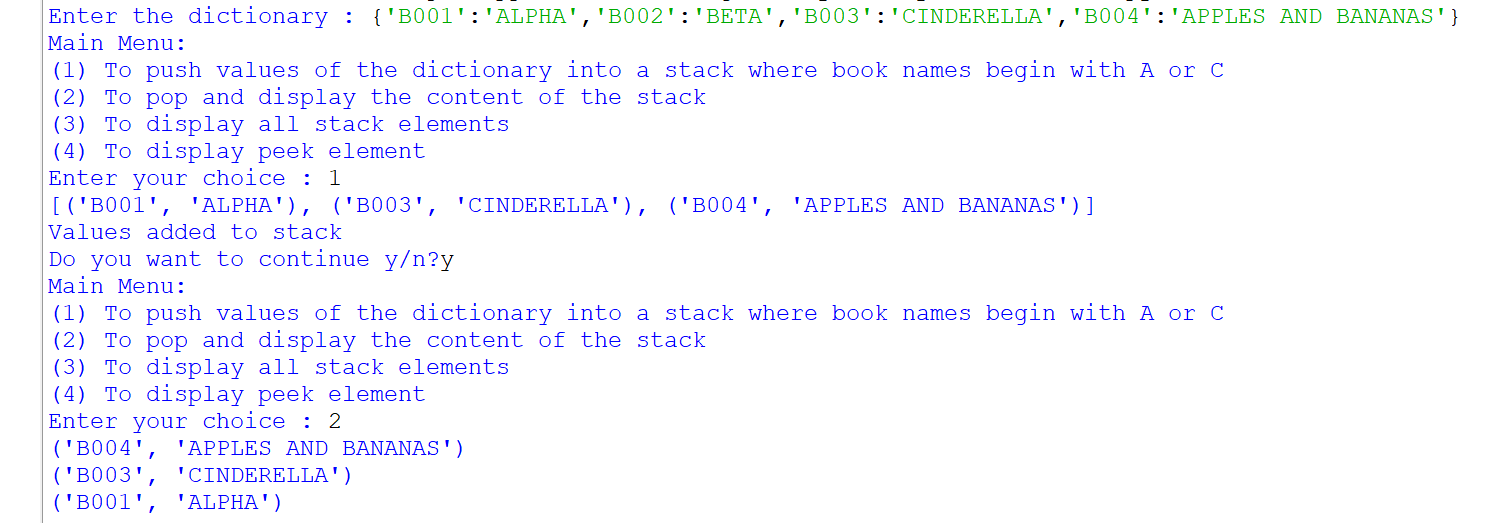
elif c == 4:

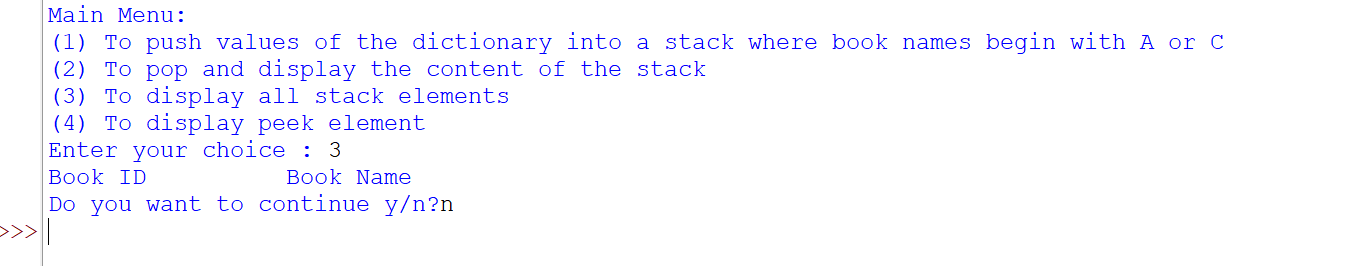
peek(Stack)

else:

print("Invalid Choice")

ans=input('Do you want to continue y/n?')





**16] Consider the following two tables: PRODUCT and CLIENT.**

Table

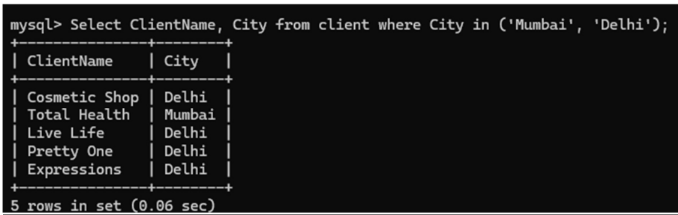
Description automatically generated

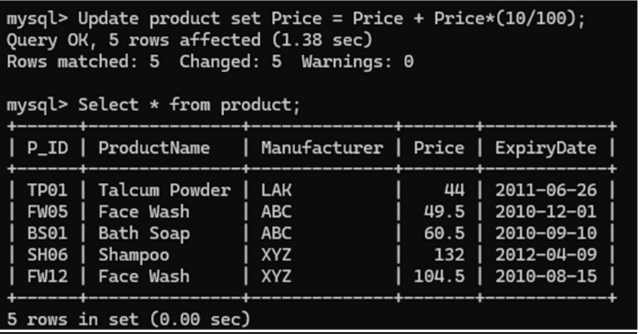
Table

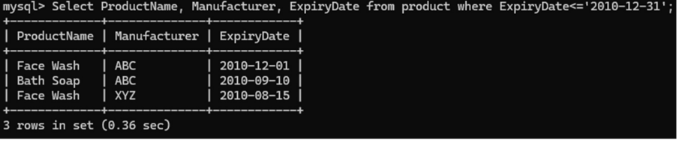
Description automatically generated

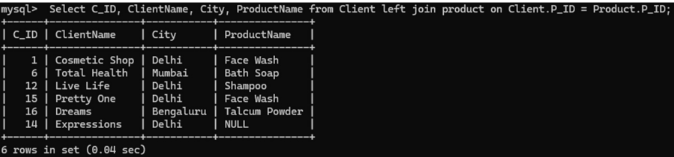
**Write SQL statements for the following:**

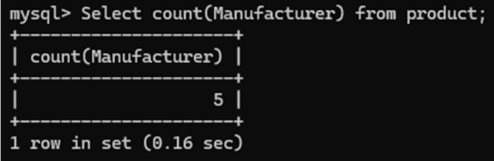
1. **To display the ClientName and City of all Mumbai and Delhi-based clients in the Client table.**
2. **Increase the price of all the products in the Product table by 10%.**
3. **To display the ProductName, Manufacturer, ExpiryDate of all the products that expired on or before ‘2010-12-31’.**
4. **To display C\_ID, ClientName, City of all the clients (including the ones that have not purchased a product) and their corresponding ProductName sold.**
5. **To display the total number of Manufacturer from the table Product.**
6. **To display C\_ID, ClientName and City starts with the letter ‘M’ from the table.**

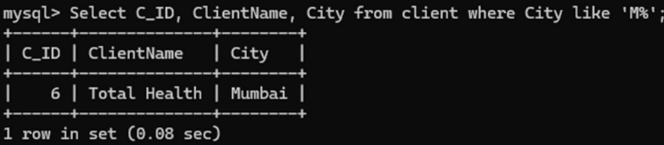
****

****

****

****

****

****

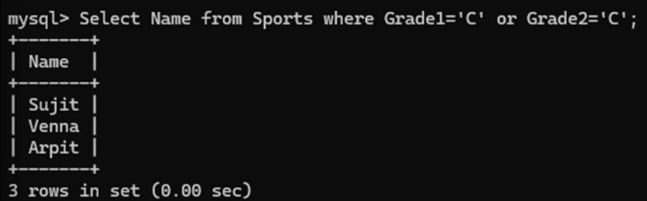
**17.Write SQL commands based on table SPORTS:**

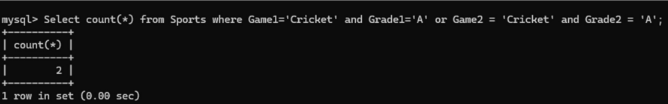
**Table: SPORTS**

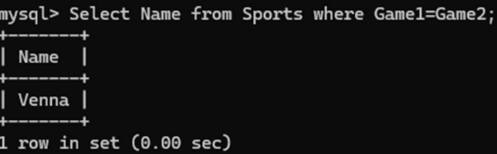
Table

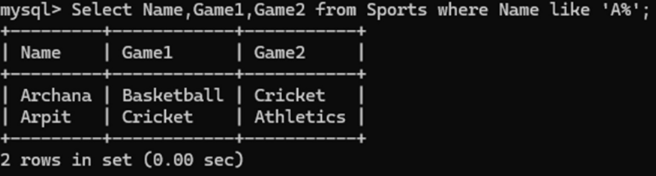
Description automatically generated

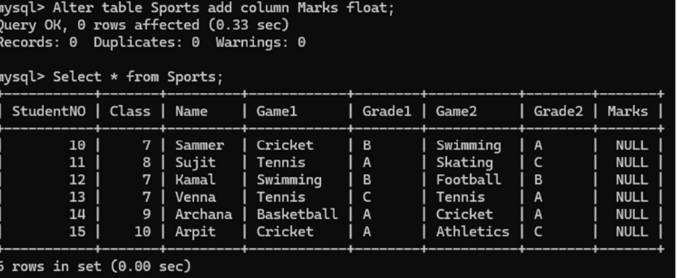
1. **Display the names of the students who have grade ‘C’ in either Game1 or Game2 or both.**
2. **Display the number of students getting grade ‘A’ in Cricket.**
3. **Display the names of the students who have the same game for both Game1 and Game2.**
4. **Display the game taken up by the students, whose name starts with ‘A’.**
5. **Add a new column named ‘Marks’.**
6. **Assign a value 200 for marks for all those who are getting grade ‘B’ or grade ‘A’ in both Game1 and Game2.**

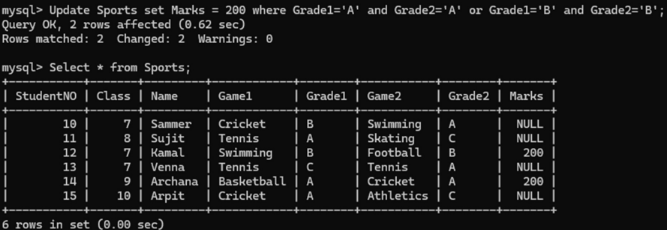
****

****

****

****

****

****

**18.Consider the following two tables: STATIONERY and CONSUMER.**

**Table: Stationery**

Table

Description automatically generated

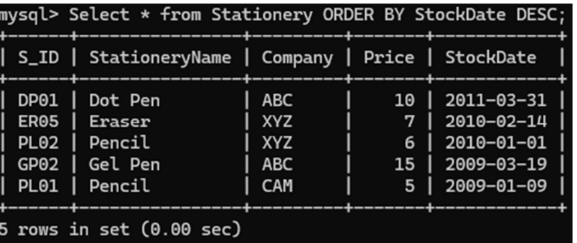
**Table: Consumer**

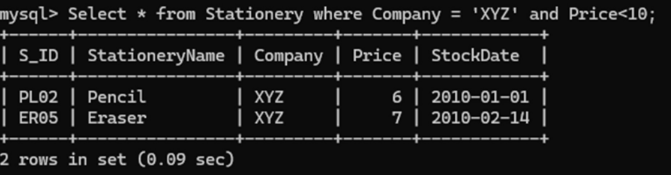
Table

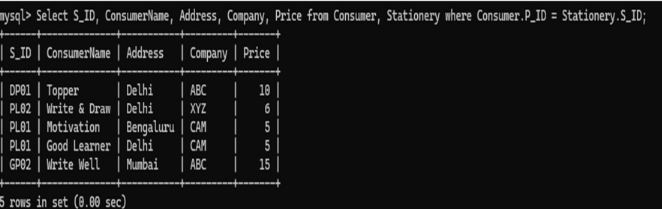
Description automatically generated

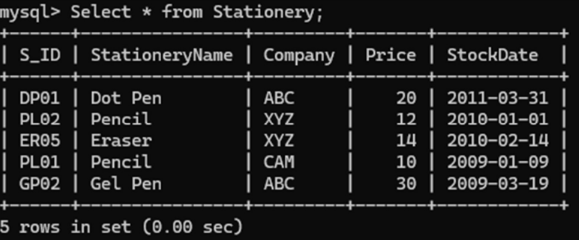
**Write SQL statements for the following:**

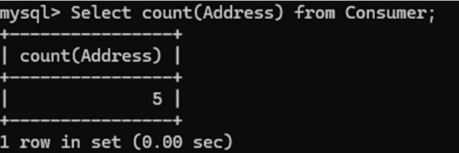
1. **To display details of all the Stationery Items in the Stationery table in descending order of StockDate.**
2. **To display details of that Stationery item whose Company is XYZ and price is below 10.**
3. **To display ConsumerName, Address from the table Consumer and Company and Price from Stationery table, with their corresponding S\_ID.**
4. **To increase the price of all the stationery items in the Stationery table by 2.**
5. **To display number of Address from the table Consumer;**
6. **Select StationeryName, price \* 3 from Stationery where Company = ‘CAM’;**

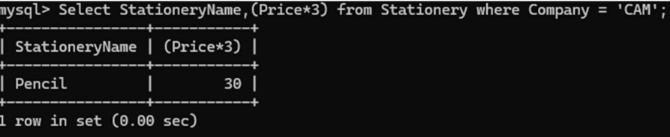
****

****

****

****

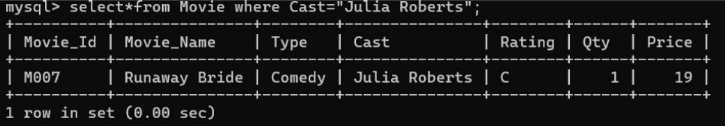
****

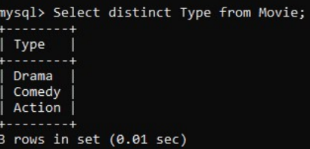
****

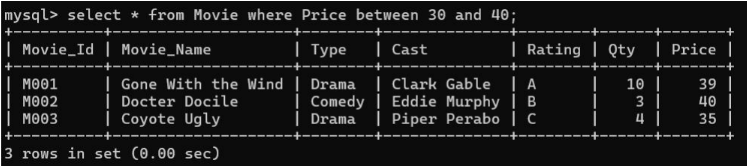
**19.Create a table with constraints and Write SQL commands for the following:**

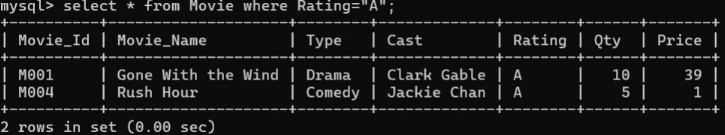
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Movie Id** | **Movie Name** | **Type** | **Cast** | **Rating** | **Qty** | **Price** |
| M001 | Gone With the Wind | Drama | Clark Gable | A | 10 | 39 |
| M002 | Doctor DoLittle | Comedy | Eddie Murphy | B | 3 | 40 |
| M003 | Coyote Ugly | Drama | Piper Perabo | C | 4 | 35 |
| M004 | Rush Hour | Comedy | Jackie Chan | A | 5 | 1 |
| M005 | Bourne Identity | Action | Matt Damon | B | 7 | 22 |
| M006 | Casino Royale | Action | Daniel Craig | B | 3 | 90 |
| M007 | Runaway Bride | Comedy | Julia Roberts | C | 1 | 19 |

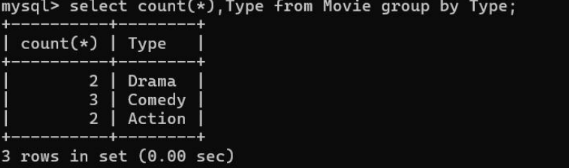
1. **Find the movies starring Julia Roberts.**
2. **To display the different types of movies available.**
3. **Display all movies which have a price range between 30 and 40.**
4. **Display all movies where the Rating is A**
5. **Display count of movies by type.**
6. **Display movie names which are starting from ‘R’.**

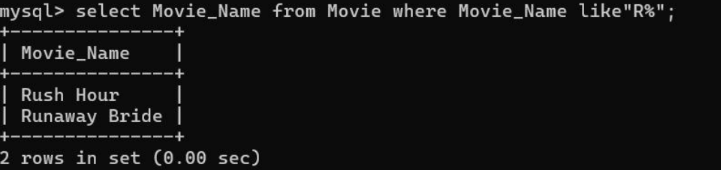












**20.Consider the following table:**

**Table: Employee**

A screenshot of a computer

Description automatically generated with medium confidence

**Table: Job**

Text

Description automatically generated

**Write SQL Queries for the following:**

1. **To display employee ids, names of employees, job ids with corresponding job titles.**
2. **To display names of employees, sales and job titles who have achieved sales more than 1300000.**
3. **To display names and job titles who have ‘SINGH’ (anywhere) in their names.**
4. **Write SQL command to change the jobid to 104 of the employee with ID as E4 in the table Employee.**
5. **Display the contents of the tables.**
6. **Display the structure of the tables.**
7. **Show the average salary for all jobtitle with 2 or more than 2 for a job.**
8. **Display only the jobs with maximum salary greater than or equal to 80000.**
9. **Find out the number of employees having “manager” as job.**
10. **List the count of employees corresponding to their jobid.**
11. **List the sum of employees’ salaries corresponding to their jobtitle.**
12. **List the maximum salary of employees corresponding to their jobid.**

**Code:**

1. Select employeeid, name,job.jobid, jobtitle from employee, job where employee.jobid = job.jobid;
2. Select name, sales, jobtitle from employee, job where sales>1300000 and employee.jobid = job.jobid;
3. Select name, jobtitle from employee, job where name like “%singh%” and employee.jobid = job.jobid;
4. Update employee set jobid=104 where employeeid = “E4”;
5. Select \* from employee, job;
6. Desc employee; desc job;
7. Select avg(salary), jobtitle from job group by jobtitle having count(\*)>=2;
8. Select jobtitle from job where salary>=80000;
9. Select jobtitle, count(\*) from job where jobtitle like”%manager%”;
10. Select jobtitle, count(\*) from job group by jobtitle;
11. Select sum(salary), jobtitle from employee , job where employee.jobid = job.jobid group by jobtitle;

l.Select max(salary), job.jobid from employee, job where employee.jobid = job.jobid group by job.jobid;

**21.  Write a menu-driven program in Python to establish connection between Python and MySQL to create a table ‘PRODUCT’ with the following description.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **P\_ID** | **ProductName** | **Manufacturer** | **Price** | **ExpiryDate** |
| **TP01** | **Talcum Powder** | **LAK** | **40** | **2011-06-26** |
| **FW05** | **Face Wash** | **ABC** | **45** | **2010-12-01** |
| **BS01** | **Bath Soap** | **ABC** | **55** | **2010-09-10** |
| **SH06** | **Shampoo** | **XYZ** | **120** | **2012-04-09** |
| **FW12** | **Face Wash** | **XYZ** | **95** | **2010-08-15** |

**Perform following operations using the connection:**

**i)    Insert at least 5 rows with appropriate data.**

**ii)   To display the records in descending order of Price Column.**

**iii)   Increase the price of all the products in Product table by 10%**

import mysql.connector as ctr

c = ctr.connect(host = 'localhost', user = 'root', password = 'mysql', database = 'psbs')

if C.is\_connected()== False:

print('unable to connect')

cob = c.cursor()

cob.execute("Create table Product (P\_ID varchar(4), ProductName varchar(30), Manufacturer varchar(3), Price float, ExpiryDate Date" )

def Insert():

ans= 'y'

while ans=='y':

id = input("Enter product ID : ")

n = input("Enter product name : ")

man = input("Enter manufacturer : ")

pr = float(input("Enter price of product : "))

edate = input("Enter expiry date (YYYY-MM-DD) : ")

s = "Insert into Product values('{}','{}','{}',{},'{}')".format(id, n,man,pr,edate)

cob.execute(s)

c.commit()

ans = input("Do you want to add more records : ")

print("Records added")

def Display():

s= "Select \* from Product ORDER BY Price DESC"

cob.execute(Q)

data = cob.fetchall()

r= cob.rowcount

print("Total number of records :", Count)

for i in data:

print(i)

def Increase():

s = "Update Product set Price = Price + Price\*(10/100)"

cob.execute(s)

c.commit()

print("Price has been updated")

ans='y'

while ans=='y':

print("""Menu:

(1) Program to insert at least 5 records

(2) Program to display the records in descending order of Price Column

(3) Program to increase the price of all products in the Product table by 10% """)

Choice = int(input("Enter your choice : "))

if Choice == 1:

Insert()

elif Choice == 2:

Display()

elif Choice == 3:

Increase()

else:

print("Invalid Choice")

C.close()

ans=input('Do you want to continuey/n?')