**PYTHON**

**#1.Accepting 3 marks for 5 students using a nested tuple. Display the total andaverage marks ofeach student.**

**Code:**

#%%

l=()

for i in range(5):

t=()

print('STUDENT',i+1)

for i in range(3):

s=int(input("enter marks of the student = "))

t+=(s,)

l+=(t,)

print(l)

for i in range(5):

s=l[i][0]+l[i][1]+l[i][2]

avg=(s/300)\*100

print("TOTAL OF THE STUDENT",i+1,"=",s)

print("AVERAGE OF THE STUDENT",i+1,"=",avg)

#%%

**Output:**

**STUDENT 1**

**enter marks of the student = 88**

**enter marks of the student = 88**

**enter marks of the student = 88**

**STUDENT 2**

**enter marks of the student = 89**

**enter marks of the student = 89**

**enter marks of the student = 89**

**STUDENT 3**

**enter marks of the student = 90**

**enter marks of the student = 90**

**enter marks of the student = 90**

**STUDENT 4**

**enter marks of the student = 67**

**enter marks of the student = 68**

**enter marks of the student = 79**

**STUDENT 5**

**enter marks of the student = 78**

**enter marks of the student = 90**

**enter marks of the student = 88**

**((88, 88, 88), (89, 89, 89), (90, 90, 90), (67, 68, 79), (78, 90, 88))**

**TOTAL OF THE STUDENT 1 = 264**

**AVERAGE OF THE STUDENT 1 = 88.0**

**TOTAL OF THE STUDENT 2 = 267**

**AVERAGE OF THE STUDENT 2 = 89.0**

**TOTAL OF THE STUDENT 3 = 270**

**AVERAGE OF THE STUDENT 3 = 90.0**

**TOTAL OF THE STUDENT 4 = 214**

**AVERAGE OF THE STUDENT 4 = 71.33333333333334**

**TOTAL OF THE STUDENT 5 = 256**

**AVERAGE OF THE STUDENT 5 = 85.33333333333334**

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

**#2. Accept a list of numbers as per the users choice. Reverse the original list. Display the list before and after the changes.**

**Code:**

#%%

l=[]

o=int(input("Enter number of elements in list:"))

for k in range(o):

b=int(input("Enter element:"))

l.append(b)

print("Original list",l)

n=len(l)-1

for i in range(len(l)//2):

l[i],l[n-i]=l[n-i],l[i]

print("Reversed list",l)

#%%

**Output:**

**Enter number of elements in list:5**

**Enter element:1**

**Enter element:2**

**Enter element:3**

**Enter element:4**

**Enter element:5**

**Original list [1, 2, 3, 4, 5]**

**Reversed list [5, 4, 3, 2, 1]**

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

**#3. Create a dictionary to hold lower case alphabets as keys and their mirror character as value. Accept a string from the user( only having lowercase alphabets) and display it in its mirror form using the dictionary.**

**Code:**

#%%

l=[]

d={}

for i in range(26):

d[chr(97+i)]=chr(122-i)

a=input("enter a lower case string...")

st=""

for j in a:

st+=d[j]

print(st)

#%%

**Output:**

**enter a lower case string...prachi**

**kizxsr**

***PROGRAM 4***

***#4‘’’Write a program to generate a random number with exactly ‘n’ digits. Use***

***a function generate( ) that takes an integer ‘n’ as argument ,generates a random number havingexactly ‘n’ digits(not starting from zero) and returns it .’’’***

***Code:***

#%%

import random

def generate(n):

r=random.randint(1,n)

print(r)

a=0

a=r\*(10\*\*(n-1))+random.randint(0,9)

return a

n=int(input("ENTER THE NUMBER OF DIGITS TO BE PRINTED"))

generate(n)

#%%

***Output:***

***ENTER THE NUMBER OF DIGITS TO BE PRINTED : 5***

***30003***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***PROGRAM 5***

**#5 ‘’’Calculate the sum of the following series for n terms. Accept x and n from the user and**

**pass it to the function sum.**

**x - x 3 /3! + x 5 /5! - x 7 /7! + x 9 /9! ........**

**Create functions for the following:**

** power which will calculate the power of x and returns it(not to use math.pow())**

** factorial which will calculate the factorial of the given number and returns it**

** Sum which will call the functions power and factorial , calculates the sum of the given**

**series and returns it.’’’**

***Code:***

#%%

def power(x,n):

a=x\*\*n

return a

def fact(x):

sum1=1

for i in range(x,1,-1):

sum1\*=i

return sum1

def sum2(x,n):

s=0

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

if i%2==0:

l=i+l

s-=(x\*\*1)/(1)

else:

s+=(x\*\*i)/(i)

return s

x1=int(input("ENTER NUMBER : "))

n1=int(input("ENTER NUMBER: "))

ans=power(x1,n1)

ans1=fact(x1)

ans2=sum2(x1,n1)

print("power of a number",ans)

print("factorial of

a number",ans1)

print("sum of series",ans2)

#%%

***Output:***

***ENTER NUMBER : 3***

***ENTER POWER : 4***

***power of a number 81***

***factorial ofa number 6***

***sum of series 3.0***

***PROGRAM 6***

***#6 ‘’’Write functions to do the following:***

***Declare a dictionary d globally.***

***CREATE( )***

***Accept values from the user to store the noon temperature of each day of the week for a month.***

***The key for the dictionary would be the week number(1,2,3,4) and values would be a list of 7***

***temperatures for each week.***

***CHECK( )***

***Checks the dictionary and prints each weeks average temperature. It also should print thehottest and coolest day of the month.’’’***

***Code:***

#%%

d={}

for i in range(1,5):

print("week : ",i)

l=[]

for j in range(1,8):

print("day : ",j)

t=int(input("enter temperature : "))

l.append(t)

d[i]=l

print(d)

def check(d):

mi=[]

ma=[]

for i in d:

avg=0

for j in d[i]:

avg+=j

print("average of week",i,"=",avg/7)

mi.append(min(d[i]))

ma.append(max(d[i]))

for i in d:

if min(mi) in d[i]:

a=d[i].index(min(mi))

c=((i-1)\*7)+a+1

if max(ma) in d[i]:

b=d[i].index(max(ma))

h=((i-1)\*7)+b+1

print("day",c,"is the coldest day of the month with temperature",min(mi))

print("day",h,"is the hottest day of the month with temperature",max(ma))

check(d)

***Output:***

***week : 1***

***day : 1***

***enter temperature : 15***

***day : 2***

***enter temperature : 16***

***day : 3***

***enter temperature : 17***

***day : 4***

***enter temperature : 18***

***day : 5***

***enter temperature : 19***

***day : 6***

***enter temperature : 20***

***day : 7***

***enter temperature : 21***

***week : 2***

***day : 1***

***enter temperature : 22***

***day : 2***

***enter temperature : 33***

***day : 3***

***enter temperature : 44***

***day : 4***

***enter temperature : 23***

***day : 5***

***enter temperature : 24***

***day : 6***

***enter temperature : 25***

***day : 7***

***enter temperature : 26***

***week : 3***

***day : 1***

***enter temperature : 27***

***day : 2***

***enter temperature : 17***

***day : 3***

***enter temperature : 32***

***day : 4***

***enter temperature : 34***

***day : 5***

***enter temperature : 21***

***day : 6***

***enter temperature : 20***

***day : 7***

***enter temperature : 31***

***week : 4***

***day : 1***

***enter temperature : 32***

***day : 2***

***enter temperature : 35***

***day : 3***

***enter temperature : 36***

***day : 4***

***enter temperature : 35***

***day : 5***

***enter temperature : 23***

***day : 6***

***enter temperature : 14***

***day : 7***

***enter temperature : 26***

***{1: [15, 16, 17, 18, 19, 20, 21], 2: [22, 33, 44, 23, 24, 25, 26], 3: [27, 17, 32, 34, 21, 20, 31], 4: [32, 35, 36, 35, 23, 14, 26]}***

***average of week 1 = 18.0***

***average of week 2 = 28.142857142857142***

***average of week 3 = 26.0***

***average of week 4 = 28.714285714285715***

***day 27 is the coldest day of the month with temperature 14***

***day 10 is the hottest day of the month with temperature 44***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***PROGRAM 7***

***#7’’’Accept a list of five numbers from the user. Pass the list to a function CONVERT( ). If thenumber is a non prime number convert it to the next higher prime number and if it is a primenumber convert it to the next higher composite number. All changes are to be done on theoriginal list. CONVERT( ) should call another function PRIME( ) which will take one elementof the list at a time and returns 1 if it is prime or 0 if it is not. Display the list before and after thechange.***

***Ex: If p = [2,4,9,13,21] then after conversion p=[4,5,11,14,23]’’’***

***Code:***

#%%

def convert(n):

global pc

pc=True

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

if n%i==0:

pc=False

break

l=[]

for k in range(5):

a=int(input("enter elements"))

l.append(a)

for j in range(5):

k=l[j]

convert(k)

if pc==True:

while pc!=False:

k=k+1

convert(k)

l[j]=k

elif pc==False:

while pc!=True:

k=k+1

convert(k)

l[j]=k

print("List after conversion : ",l)

#%%

***Output:***

***enter elements2***

***enter elements4***

***enter elements9***

***enter elements13***

***enter elements21***

***List after conversion : [4, 5, 11, 14, 23]***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***PROGRAM 8***

***#8’’’Write a program to get http request information from the url www.google.com and open itwithin your program.’’’***

***Code:***

#%%

import urllib as u

import webbrowser as w

url1=u.request.urlopen("https://www.google.com")

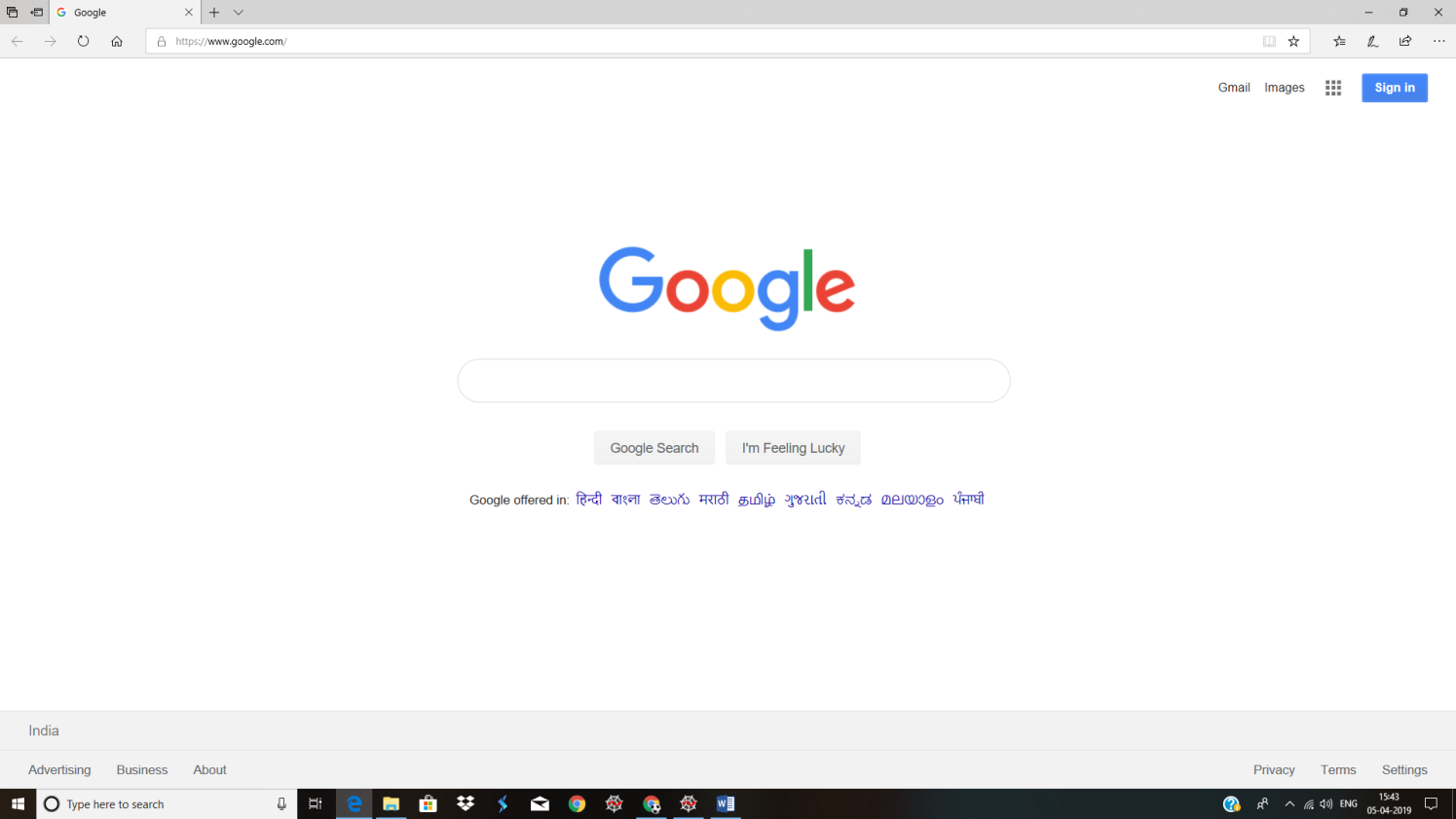
url1=url1.geturl()

w.open\_new(url1)

#%%

***Output:***

***True***



***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***PROGRAM 9***

***#9 ‘’’Createthe module length.py that has various functions for the following conversionsMiletokm( ) , Kmtomile(), Feettoinches() and inchestofeet( )***

***the module mass.py that has various functions for the following conversions***

***kgtotonne() , tonnetokg(), kgtopound() and poundtokg()***

***Write a menu driven program help.py that will import the following modules and do theconversions as per the users requirement.***

***[Note : 1 mile = 1.609km , 1 feet = 12 inches, 1kg = .001 tonne , 1 kg = 2.204 pound]’’’***

***Code:***

***lengthconversion.py--***

#%%

def miletokm(m):

km=m\*100/62

return km

def kmtomile(km):

m=km\*0.62

return m

def feettoinches(f):

i=f\*12

return i

def inchestofeet(i):

f=i\*1/12

return f

#%%

***massconversion.py—***

#%%

def kgtotonne(kg):

t=kg\*0.001

return t

def tonnetokg(t):

kg=t\*1/0.001

return kg

def kgtopound(kg):

p=kg\*2.204

return p

def poundtokg(kg):

p=kg\*2.204

return p

#%%

***CONVERSIONS***

* ***Length***

1. ***\_\_init\_\_.py***
2. ***lengthconversion.py***

* ***Mass***

1. ***\_\_init\_\_.py***
2. ***massconversion.py***

* ***\_\_init\_\_.py***

***help.py—***

#%%

import conversions

from conversions.length import lengthconversions

from conversions.mass import massconversion

print("AVAILABLE CONVERSIONS ARE AS FOLLOWS :")

print("1.length ")

print("2. mass")

n=int(input("choose number from 1 and 2 : "))

if n==1:

print("From length,")

print("Available conversions are : ")

print("1. km to mile")

print("2. mile to km")

print("3. feet to inches")

print("4.inches to feet")

m=int(input("choose between between 1 to 4 : "))

if m==1:

km=int(input("enter in terms of km"))

print(lengthconversions.kmtomile(km))

elif m==2:

m=int(input("enter in terms of miles"))

print(lengthconversions.miletokm(m))

elif m==3:

f=int(input("enter in terms of feet"))

print(lengthconversions.feettoinches(f))

elif m==4:

i=int(input("enter in terms of inches"))

print(lengthconversions.feettoinches(i))

elif n==2:

print("From mass,")

print("Available conversions are : ")

print("1. kg to tonne")

print("2. tonne to kg")

print("3. kg to pound")

print("4. pound to kg")

m=int(input("choose between between 1 to 4 : "))

if m==1:

kg=int(input("enter in terms of kg"))

print(massconversion.kgtotonne(kg))

elif m==2:

t=int(input("enter in terms of t"))

print(massconversion.tonnetokg(t))

elif m==3:

kg=int(input("enter in terms of kg"))

print(massconversion.kgtopound(kg))

elif m==4:

kg=int(input("enter in terms of kg"))

print(massconversion.kgtopound(kg))

#%%

***Output:***

***AVAILABLE CONVERSIONS ARE AS FOLLOWS :***

***1.length***

***2. mass***

***choose number from 1 and 2 : 1***

***From length,***

***Available conversions are :***

***1. km to mile***

***2. mile to km***

***3. feet to inches***

***4.inches to feet***

***choose between between 1 to 4 : 2***

***enter in terms of miles23***

***37.096774193548384***

***File handling***

***PROGRAM 10***

***10. ‘’’Write a method in Python to create a text file ‘notes’(in python) . Read the lines and countthe number of ‘.’ or ‘,’ in the file.Display the original file and the Total count of thesecharacters.’’’***

***Code:***

def notes():

f=open("notes.txt",'w')

n=int(input("enter no of lines : "))

for i in range(n):

a=input("enter lines : ")

f.write(a+'\n')

f.close()

def count():

f=open("notes.txt")

d=f.read()

print("original file : ")

print(d)

c=0

f=open("notes.txt",'r')

for i in f:

for j in i:

for k in j:

if k=="." or k==",":

c+=1

print("COUNT OF CHARACTERS(. and ,) : ",c)

notes()

count()

***Output:***

***enter no of lines : 4***

***enter lines : Hi,my name is prachi.***

***enter lines : how are you dear.....???***

***enter lines : how is mom,dad and sista...***

***enter lines : i hope,all are fine....***

***original file :***

***Hi,my name is prachi.***

***how are you dear.....???***

***how is mom,dad amnd sista...***

***i hope,all are fine....***

***COUNT OF CHARACTERS(. and ,) : 16***

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

***PROGRAM 11***

***11. ‘’’A file ‘sports’ contains information in following format : Event-participant.(create innotepad) Write a method that would read the contents and creates a file named ‘athletics’ ,copying only those records from sports where the event name is ‘ATHLETICS’.***

***Display the contents of original file sports and then new file ‘ATHLETICS’.’’’***

***Code:***

#%%

def copy():

f1=open('C:\\Users\\HP\\Documents\\Event-participant.txt','r+')

d=f1.read()

print(" Event-participant : ")

print(d)

f2=open('atheletics','w+')

f1.seek(0)

l=f1.readlines()

for i in l:

if i[0:10]=='Atheletics':

f2.write(i)

f2.seek(0)

s=f2.read()

print(" ATHELETICS : ")

print(s)

copy()

#%%

***Output:***

***Event-participant :***

***Atheletics : ANUJ***

***Badminton : PRACHI***

***Atheletics : NIDHI***

***Hokey : VINAY***

***Football : SHREHA***

***Atheletics : KUSUM***

***Cricket : RANI***

***ATHELETICS :***

***Atheletics : ANUJ***

***Atheletics : NIDHI***

***Atheletics : KUSUM***

***PROGRAM 12***

***12. ‘’’Write a method to create a textfile ‘original’ with 2 lines in it. Make a copy of it into anothertextfile ‘copy’ which will after replacing extra blank spaces with a single blank space.***

***Display the contents of original and copy file.’’’***

***CODE:***

#%%

def original():

original=open('original.txt','w+')

for i in range(2):

line=input("enter line: ")

original.write(line+'\n')

original.seek(0)

return original

original=original()

copy=open('copy.txt','w+')

l=original.readlines()

for i in range(len(l)):

st=''

x=l[i].split()

for j in range(len(x)):

st+=x[j]+' '

copy.write(st+'\n')

copy.seek(0)

print(" COPY:\n",copy.read())

#%%

***Output:***

***enter line: prachi pathak***

***enter line: PrAcHi PaThAk***

***COPY:***

***prachi pathak***

***PrAcHi PaThAk***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***PROGRAM 13***

***13.’’’Write a method to read characters(accept from user how many characters) from the keyboardone by one. All lowercase alphabets get stored in a file ‘lower’, all upper case alphabets getstored in a file ‘upper’ and all other characters get stored in ‘others’. Display the contents of allthe three files.’’’***

***CODE:***

def char(a):

f1=open('lower.txt','a+')

f2=open('upper.txt','a+')

f3=open('others.txt','a+')

for i in f:

for j in i:

if j.islower():

f1.write(j)

elif j.isupper():

f2.write(j)

else:

f3.write(j)

f1.seek(0)

print(" LOWERCASE FILE :\n",f1.read())

f2.seek(0)

print(" UPPERCASE FILE :\n",f2.read())

f3.seek(0)

print(" OTHERS :\n",f3.read())

f=open('characters.txt','w+')

x=input("enter string in lower,upper and special characters : ")

f.write(x)

f.seek(0)

char(x)

***Output:***

***enter string in lower,upper and special characters : PrAcHi#%%PaThAk@@@***

***LOWERCASE FILE :***

***rciahk***

***UPPERCASE FILE :***

***PAHPTA***

***OTHERS :***

***##%%@@@***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***PROGRAM 14***

***14.’’’Write a method that takes 2 file names from the user(create two files testA and TestB in notepad). It should append the contents of the second file onto the first file.’’’***

***CODE:***

#%%

def file(x,y):

    a=open('1.txt','w+')

    c=input("enter content foe file 1:")

    a.write(c)

    b=open('2.txt','w+')

    d=input("enter content for file 2 :")

    b.write(d)

    b.seek(0)

    for i in b:

        for j in i:

            a.write(j)

    a.seek(0)

    print("content of file 2 in 1 is shown as:\n",a.read())

x=input("enter name for file 1: ")

y=input("enetr name for file 2 :")

file(x,y)

#%%

***OUTPUT:***

***enter name for file 1: TEST A***

***enter name for file 2 :TEST B***

***enter content foe file 1:I LOVE TO READ***

***enter content for file 2 :STORY BOOKS***

***content of file 2 in 1 is shown as:***

***I LOVE TO READSTORY BOOKS***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***PROGRAM 15***

***15.’’’ Write a program that takes a user id and password from the user and appends it to a file ‘security’, provided the given userid does not exist in the file. If it does , then display error message “ User id already exists” and prompts the user to reenter the userid. Also make sure thatthe password is atleast 8 characters long, has a digit and a special character (only $ , @, %) in it.’’’***

***CODE:***

s=open(r'C:\Users\HP\Documents\security.txt','a+')

s.seek(0)

r=s.read()

while True:

u=input("enter id")

if u not in r:

s.write('\n')

s.write(u)

break

else:

print('id exists')

cd=0

cs=0

while True:

p=input('password')

if len(p)<7:

print("password is too short")

for i in p:

if i.isdigit():

cd+=1

elif i in'@$%':

cs+=1

if cd==0:

print('add digit to password')

if cs==0:

print('add special char')

else:

s.write(p)

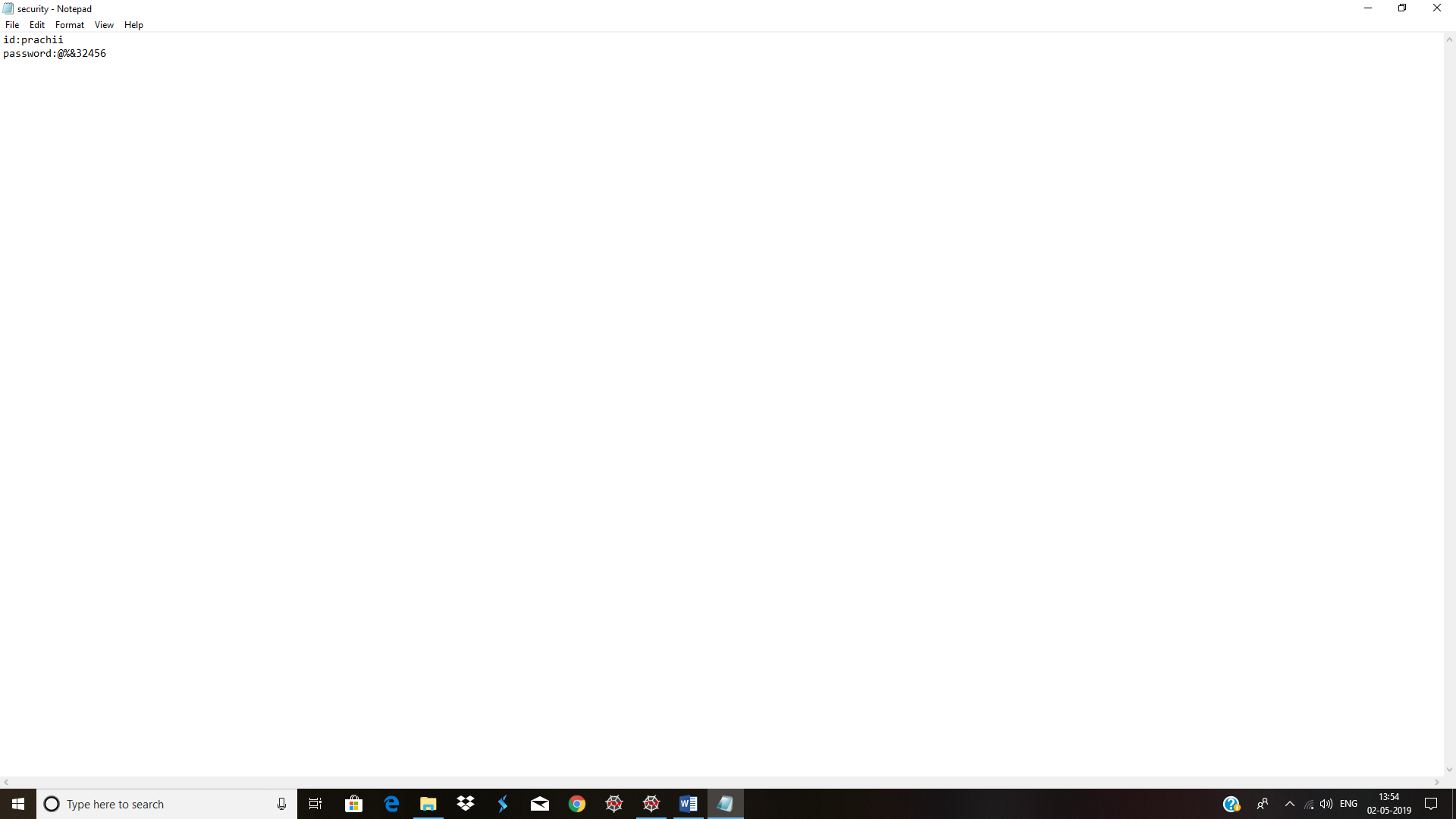
break

s.close()

**output:**

**enter id: prachii**

**password: @%&32456**



***MYSQL***

***PROGRAM 16***

***16. CREATE TABLE EMPLOYEE WITH FOLLOWING FIELDS AND CONSTRAINTS***

***ENO  INT PK***

***ENAME VARCHAR(20) NOT NULL***

***SAL DECIMAL(10,2) CAN BE &gt;5000 AND &lt;=100000***

***COMM  DECIMAL(10,2)***

***DEPTNO INT CAN BE 10,50,80***

***JOB VARCHAR(10) CAN BE MGR,CLERK,FIN,MKT***

***DOJ DATE DEFAULT 2000-10-10***

***INSERT 5 RECORDS***

***Syntax:***

create table employee

-> (eno int primary key,

-> ename varchar(20) not null,

-> sal decimal(10,2),

-> dept int check(deptno in(50,10,80)),

-> job varchar(10) check (job in('MGR','CLERK','FIN','MKT')),

-> doj DATE default '2000-10-10'

-> );

**Query OK, 0 rows affected (0.28 sec)**

insert into employee

-> values(01,'A',6000,100,'MGR','2000-10-10'),

-> (02,'B',2000,100,'MGR','2000-10-10'),

-> (03,'C',8000,110,'FIN','2000-10-11'),

-> (04,'D',8000,200,'MKT','2000-10-12'),

-> (05,'E',10000,200,'CLERK','2000-10-12');

**Query OK, 5 rows affected (0.13 sec)**

**Records: 5 Duplicates: 0 Warnings: 0**

select \* from employee;

**+-----+-------+----------+------+-------+------------+-----+**

**|eno| ename | sal | dept | job | doj |**

**+-----+-------+----------+------+-------+------------+**

**| 1 | A | 6000.00 | 100 | MGR | 2000-10-10 |**

**| 2 | B | 2000.00 | 100 | MGR | 2000-10-10 |**

**| 3 | C | 8000.00 | 110 | FIN | 2000-10-11 |**

**| 4 | D | 8000.00 | 200 | MKT | 2000-10-12 |**

**| 5 | E | 10000.00 | 200 | CLERK | 2000-10-12 |**

**+-----+-------+----------+------+-------+-----------------+**

**5 rows in set (0.03 sec)**

***Write the following queries***

1. ***Delete the records where job is clerk.***

DELETE from employee

-> WHERE job='CLERK';

***Query OK, 1 row affected (0.18 sec)***

1. ***Update all the records by increasing salary by 10%.***

*UPDATE employee*

*-> set sal=sal+sal\*10;*

***Query OK, 4 rows affected (0.22 sec)***

***Rows matched: 4 Changed: 4 Warnings: 0***

1. ***Update the job of empno 1 to mkt.***

*Update employee*

*->set job=’MKT’*

*->WHERE eno=1;*

***Query OK, 4 rows affected (0.22 sec)***

***Rows matched: 1 Changed: 1 Warnings: 0***

***4. Modify the comm of employees in FIN job to 1000.***

UPDATE employee

-> set sal = 1000

-> WHERE job = 'FIN';

***Query OK, 1 row affected (0.39 sec)***

***Rows matched: 1 Changed: 1 Warnings: 0***

***5. Display all the records.***

> SELECT \* FROM EMPLOYEE;

***+-----+-------+----------+------+------+------------+------+----+***

***| eno | ename | sal | dept | job | doj |***

***+-----+-------+----------+------+------+------------+-----+----+***

***| 1 | A | 66000.00 | 100 | MKT | 2000-10-10 |***

***| 2 | B | 22000.00 | 100 | MGR | 2000-10-10 |***

***| 3 | C | 1000.00 | 110 | FIN | 2000-10-11 |***

***| 4 | D | 88000.00 | 200 | MKT | 2000-10-12 |***

***+-----+-------+----------+------+------+------------+----------+--+***

***4 rows in set (0.12 sec)***

***6. Change the deptno to 50 and job as MKT  for the employees where comm is null***

*Update employee*

*Set deptno=50 and job =’mkt’*

*Where comm is null;*

***Query OK, 0 rows affected (0.26 sec)***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***PROGRAM 17***

***17.Create table member***

***mid varchar(5),***

***fname varchar(10)***

***contractduration char(1)***

***SYNTAX:***

create table member

-> (mid varchar(5),

-> fname varchar(10),

-> contractduration char(1)

-> );

***Query OK, 0 rows affected (0.26 sec)***

***Do the following after creating above table:***

1. ***add primary key to mid column***

*alter table member*

*-> add primary key(mid);*

***Query OK, 0 rows affected (0.43 sec)***

***Records: 0 Duplicates: 0 Warnings: 0***

1. ***add not null to fname column***

*alter table member*

*-> modify fname varchar(10) not null;*

***Query OK, 0 rows affected (0.40 sec)***

***Records: 0 Duplicates: 0 Warnings: 0***

***3.add new columns as lname,caddress***

*alter table member*

*-> add(lname char(10),caddress varchar(15):*

***Query OK, 0 rows affected (0.35 sec)***

***Records: 0 Duplicates: 0 Warnings: 0***

***4. change the size of fname column to varchar(20)***

alter table member

-> modify fname varchar(20);

***Query OK, 0 rows affected (0.23 sec)***

***Records: 0 Duplicates: 0 Warnings: 0***

***5. change the datatype of contractduration column to int***

*alter table member*

*-> modify contractduration int;*

***Query OK, 0 rows affected (0.28 sec)***

***Records: 0 Duplicates: 0 Warnings: 0***

***6. change the name of contractduration to condur***

*alter table member*

*-> change contractduration condur varchar(1);*

***Query OK, 0 rows affected (0.32 sec)***

***Records: 0 Duplicates: 0 Warnings: 0***

***7. change the order of column lname after fname.***

*alter table member*

*-> modify lname char(10) after fname;*

***Query OK, 0 rows affected (0.31 sec)***

***Records: 0 Duplicates: 0 Warnings: 0***

***8. specify the default value of condur to 1***

*alter table member*

*-> modify condur int default 1;*

***Query OK, 0 rows affected (0.30 sec)***

***Records: 0 Duplicates: 0 Warnings: 0***

***9. drop the column caddress***

*alter table member*

*-> drop caddress;*

***Query OK, 0 rows affected (0.32 sec)***

***Records: 0 Duplicates: 0 Warnings: 0***

***10. drop the primary key of the table.***

*alter table member*

*-> drop primary key;*

***Query OK, 0 rows affected (0.33 sec)***

***Records: 0 Duplicates: 0 Warnings: 0***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***PROGRAM 18***

***18.Consider the following table named “GYM” with details about Fitness products being sold in the store.***

***Table Name : GYM***

***+--------+-------------------+-----------+--------------+***

***| prcode | prname | unitprice | manufacturer |***

***+--------+-------------------+-----------+--------------+***

***| p101 | cross trainer | 25000 | avon fitness |***

***| p102 | treadmill | 28800 | ag fitline |***

***| p103 | massage chair | 18000 | fit express |***

***| p104 | vibration trainer | 19800 | avon fitness |***

***| p105 | bike | 11700 | fitexpress |***

***| p106 | vibro exerciser | 20700 | NULL |***

***+--------+-------------------+-----------+--------------+***

*create table gym*

*-> (prcode varchar(10),*

*-> prname varchar(30),*

*-> unitprice int(6),*

*-> manufacturer char(30)*

*-> );*

***Query OK, 0 rows affected (0.17 sec)***

***+--------------+-------------+------+-----+---------+-------+-------***

***| Field | Type | Null | Key | Default | Extra |***

***+--------------+-------------+------+-----+---------+-------+-----***

***| prcode | varchar(10) | YES | | NULL | |***

***| prname | varchar(30) | YES | | NULL | |***

***| unitprice | int(6) | YES | | NULL | |***

***| manufacturer | char(30) | YES | | NULL | |***

***+--------------+-------------+------+-----+---------+-------+------***

*> insert into gym*

*-> values('p101','cross trainer',25000,'avon fitness'),*

*-> ('p102','treadmill',28800,'ag fitline'),*

*-> ('p103','massage chair',18000,'fit express'),*

*-> ('p104','vibration trainer',19800,'avon fitness'),*

*-> ('p105','bike',11700,'fitexpress'),*

*-> ('p106','vibro exerciser',20700,null);*

***Query OK, 6 rows affected (0.12 sec)***

***Records: 6 Duplicates: 0 Warnings: 0***

*>select \* from gym;*

***+--------+-------------------+-----------+--------------+***

***| prcode | prname | unitprice | manufacturer |***

***+--------+-------------------+-----------+--------------+***

***| p101 | cross trainer | 25000 | avon fitness |***

***| p102 | treadmill | 28800 | ag fitline |***

***| p103 | massage chair | 18000 | fit express |***

***| p104 | vibration trainer | 19800 | avon fitness |***

***| p105 | bike | 11700 | fitexpress |***

***| p106 | vibro exerciser | 20700 | NULL |***

***+--------+-------------------+-----------+--------------+***

***6 rows in set (0.00 sec)***

***Write SQL statements to do the following:***

***a) Display the names and unit price of all the products in the store***

*select prname,unitprice from gym;*

***+-------------------+-----------+***

***| prname | unitprice |***

***+-------------------+-----------+***

***| cross trainer | 25000 |***

***| treadmill | 28800 |***

***| massage chair | 18000 |***

***| vibration trainer | 19800 |***

***| bike | 11700 |***

***| vibro exerciser | 20700 |***

***+-------------------+-----------+-----***

***6 rows in set (0.10 sec)***

***b) Display the names of all the products with unit price less than Rs.20000.00***

*select prname,unitprice*

*-> from gym*

*-> where unitprice<20000;*

***+-------------------+-----------+***

***| prname | unitprice |***

***+-------------------+-----------+***

***| massage chair | 18000 |***

***| vibration trainer | 19800 |***

***| bike | 11700 |***

***+-------------------+-----------+***

***3 rows in set (0.10 sec)***

***c) Display details of all the products with unit price in the range 20000 to 30000***

*select \* from gym*

*-> where unitprice between 20000 and 30000;*

***+--------+-----------------+-----------+--------------+-----***

***| prcode | prname | unitprice | manufacturer |***

***+--------+-----------------+-----------+--------------+-----***

***| p101 | cross trainer | 25000 | avon fitness |***

***| p102 | treadmill | 28800 | ag fitline |***

***| p106 | vibro exerciser | 20700 | NULL |***

***+--------+-----------------+-----------+--------------+-----***

***3 rows in set (0.13 sec)***

***d) Display names of all products by the select manufacturer &”Fit Express”***

*select prname*

*-> from gym*

*-> where manufacturer = 'fit express';*

***+---------------+----***

***| prname |***

***+---------------+----***

***| massage chair |***

***+---------------+----***

***1 row in set (0.00 sec)***

***e) Add a new row for product with the details: “P106”,”Vibro Exerciser”, 23000,manufacturer : null.***

*insert into gym*

*-> values("P106","Vibro Exerciser", 23000,null);*

***Query OK, 1 row affected (0.18 sec)***

***f) Change the Unit Price data of all the rows by applying a 10% discountreduction on all the products.***

*select unitprice+(10/100\*unitprice) as "new price" from gym;*

***+------------+--+***

***| new price |***

***+------------+--+***

***| 27500.0000 |***

***| 31680.0000 |***

***| 19800.0000 |***

***| 21780.0000 |***

***| 12870.0000 |***

***| 22770.0000 |***

***| 25300.0000 |***

***+------------+--+***

***7 rows in set (0.09 sec)***

***g) Display details of all products with manufacturer name starting with “A”***

*select \* from gym*

*-> where manufacturer like 'a%';*

***+--------+-------------------+-----------+--------------+---+***

***| prcode | prname | unitprice | manufacturer |***

***+--------+-------------------+-----------+--------------+---+***

***| p101 | cross trainer | 25000 | avon fitness |***

***| p102 | treadmill | 28800 | ag fitline |***

***| p104 | vibration trainer | 19800 | avon fitness |***

***+--------+-------------------+-----------+--------------+---+***

***3 rows in set (0.00 sec)***

***h)Display details of all products with manufacturer name not ending  with “s”***

*select \* from gym*

*-> where manufacturer not like '%s';*

***+--------+-----------+-----------+--------------+---------+***

***| prcode | prname | unitprice | manufacturer |***

***+--------+-----------+-----------+--------------+--------+***

***| p102 | treadmill | 28800 | ag fitline |***

***+--------+-----------+-----------+--------------+----------+***

***1 row in set (0.00 sec)***

***i) Display all rows sorted in descending order of unit price.***

*select \* from gym*

*-> order by unitprice desc;*

***+--------+-------------------+-----------+--------------+-----+***

***| prcode | prname | unitprice | manufacturer |***

***+--------+-------------------+-----------+--------------+-----+***

***| p102 | treadmill | 28800 | ag fitline |***

***| p101 | cross trainer | 25000 | avon fitness |***

***| P106 | Vibro Exerciser | 23000 | NULL |***

***| p106 | vibro exerciser | 20700 | NULL |***

***| p104 | vibration trainer | 19800 | avon fitness|***

***| p103 | massage chair | 18000 | fit express|***

***| p105 | bike | 11700 | fitexpress|***

***+--------+-------------------+-----------+--------------+-----+***

***7 rows in set (0.10 sec)***

***j)Display the name and price where manufacturer is null***

*select prname,unitprice*

*from gym*

*-> where manufacturer is null;*

***+-----------------+-----------+***

***| prname | unitprice |***

***+-----------------+-----------+***

***| vibro exerciser | 20700 |***

***| Vibro Exerciser | 23000 |***

***+-----------------+-----------+***

***2 rows in set (0.00 sec)***

***k)Display the name and price where manufacturer is not null***

*select prname,unitprice*

*from gym*

*-> where manufacturer is not null;*

***+-------------------+-----------+***

***| prname | unitprice |***

***+-------------------+-----------+***

***| cross trainer | 25000 |***

***| treadmill | 28800 |***

***| massage chair | 18000 |***

***| vibration trainer | 19800 |***

***| bike | 11700 |***

***+-------------------+-----------+***

***5 rows in set (0.00 sec)***

***l) Display manufacturer, whose name has 10 characters***

*select manufacturer*

*from gym*

*-> where manufacturer like '\_\_\_\_\_\_\_\_\_\_';*

***+--------------+***

***| manufacturer |***

***+--------------+***

***| ag fitline |***

***| fitexpress |***

***+--------------+***

***2 rows in set (0.00 sec)***

***m) Display manufacturer, whose name not starting with A alphabet.***

*select manufacturer from gym*

*-> where manufacturer not like 'a%';*

***+--------------+***

***| manufacturer |***

***+--------------+***

***| fit express |***

***| fitexpress |***

***+--------------+--***

***n)Display the name and price where manufacturer is not null***

*select prname,unitprice from gym*

*-> where manufacturer is not null;*

***+-------------------+-----------+***

***| prname | unitprice |***

***+-------------------+-----------+***

***| cross trainer | 25000 |***

***| treadmill | 28800 |***

***| massage chair | 18000 |***

***| vibration trainer | 19800 |***

***| bike | 11700 |***

***+-------------------+-----------+***

***o) display the price of 10 items where manufacturer is  avon fitness . Give an alias name to price.arrange data in descending order of items price.***

*select unitprice as "price"*

*-> from gym*

*-> where manufacturer="avon fitness"*

*-> order by unitprice desc;*

***+-------+***

***| price |***

***+-------+***

***| 25000 |***

***| 19800 |***

***p)Display the report in following format for all the records:***

***Example***

***Product Name:Cross trainer is manufactured by Avon Fitness and it costs Rs. 25000/-from gym***

*select concat(prname,'is manufactured by'’*

*->manufacturer,'and its cost is Rs',unitprice,'/-')*

*-> from gym;*

***+--------------------------------------------------------------------------------------+-------------+***

***| concat(prname,'is manufactured by',manufacturer,'and its cost is Rs',unitprice,'/-') |***

***+--------------------------------------------------------------------------------------+-------------+***

***| cross traineris manufactured byavon fitnessand its cost is Rs25000/- |***

***| treadmill is manufactured byag fitlineand its cost is Rs28800/- |***

***| massage chairis manufactured byfit expressand its cost is Rs18000/- |***

***| vibration traineris manufactured byavon fitnessand its cost is Rs19800/- |***

***| bikeis manufactured byfitexpressand its cost is Rs11700/- |***

***| NULL |***

***| NULL |***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***PROGRAM 19***

***19.Create tables with following constraints***

***Table emp***

***empno int  Primary key***

***ename***

***deptno int foreign key***

***sal***

***Table dept***

***deptno int  primary key***

***dname***

***location***

***syntax:***

*create table dept*

*-> (deptno int(5) primary key,*

*-> dname varchar(29),*

*-> location varchar(50)*

*-> );*

***Query OK, 0 rows affected (0.38 sec)***

*create table emp*

*-> (empno int(5) primary key,*

*-> ename varchar(20),*

*-> deptno int, foreign key(deptno) references dept(deptno),*

*-> sal int);*

***Query OK, 0 rows affected (0.23 sec)***

*desc emp;*

***+--------+-------------+------+-----+---------+-------+--------+***

***| Field | Type | Null | Key | Default | Extra |***

***+--------+-------------+------+-----+---------+-------+--------+***

***| empno | int(5) | NO | PRI | NULL | |***

***| ename | varchar(20) | YES | | NULL | |***

***| deptno | int(11) | YES | MUL | NULL | |***

***| sal | int(11) | YES | | NULL | |***

***+--------+-------------+------+-----+---------+-------+--------+***

***4 rows in set (0.18 sec)***

***Insert datas:***

*insert into dept*

*-> values(10,'fin','nm'),*

*-> (20,'mkt','delhi'),*

*-> (30,'admin','nm'),*

*-> (40,'res','jaipur'),*

*-> (50,'sales','jaipur');*

***Query OK, 5 rows affected (0.16 sec)***

***Records: 5 Duplicates: 0 Warnings: 0***

*select \* from dept;*

***+--------+-------+----------+***

***| deptno | dname | location |***

***+--------+-------+----------+***

***| 10 | fin | nm |***

***| 20 | mkt | delhi |***

***| 30 | admin | nm |***

***| 40 | res | jaipur |***

***| 50 | sales | jaipur |***

***+--------+-------+----------+***

*insert into emp*

*-> values(101,'a',10,10000),*

*-> (102,'b',20,20000),*

*-> (103,'c',null,30000),*

*-> (104,'d',10,5000),*

*-> (105,'e',30,50000);*

***Query OK, 5 rows affected (0.24 sec)***

***Records: 5 Duplicates: 0 Warnings: 0***

*select \* from emp;*

***+-------+-------+--------+-------+***

***| empno | ename | deptno | sal |***

***+-------+-------+--------+-------+***

***| 101 | a | 10 | 10000 |***

***| 102 | b | 20 | 20000 |***

***| 103 | c | NULL | 30000 |***

***| 104 | d | 10 | 5000 |***

***| 105 | e | 30 | 50000 |***

***+-------+-------+--------+-------+***

***Write the query to:***

***1. Display the cartisean product.***

*select \**

*from emp,dept;*

***+-------+-------+--------+-------+--------+-------+----------+***

***| empno | ename | deptno | sal | deptno | dname | location |***

***+-------+-------+--------+-------+--------+-------+----------+***

***| 101 | a | 10 | 10000 | 10 | fin | nm |***

***| 102 | b | 20 | 20000 | 10 | fin | nm |***

***| 103 | c | NULL | 30000 | 10 | fin | nm |***

***| 104 | d | 10 | 5000 | 10 | fin | nm |***

***| 105 | e | 30 | 50000 | 10 | fin | nm |***

***| 101 | a | 10 | 10000 | 20 | mkt | delhi |***

***| 102 | b | 20 | 20000 | 20 | mkt | delhi |***

***| 103 | c | NULL | 30000 | 20 | mkt | delhi |***

***| 104 | d | 10 | 5000 | 20 | mkt | delhi |***

***| 105 | e | 30 | 50000 | 20 | mkt | delhi |***

***| 101 | a | 10 | 10000 | 30 | admin | nm |***

***| 102 | b | 20 | 20000 | 30 | admin | nm |***

***| 103 | c | NULL | 30000 | 30 | admin | nm |***

***| 104 | d | 10 | 5000 | 30 | admin | nm |***

***| 105 | e | 30 | 50000 | 30 | admin | nm |***

***| 101 | a | 10 | 10000 | 40 | res | jaipur |***

***| 102 | b | 20 | 20000 | 40 | res | jaipur |***

***| 103 | c | NULL | 30000 | 40 | res| jaipur |***

***| 104 | d | 10 | 5000 | 40 | res | jaipur |***

***| 105 | e | 30 | 50000 | 40 | res | jaipur |***

***| 101 | a | 10 | 10000 | 50 | sales | jaipur |***

***| 102 | b | 20 | 20000 | 50 | sales | jaipur |***

***| 103 | c | NULL | 30000 | 50 | sales | jaipur |***

***| 104 | d | 10 | 5000 | 50 | sales | jaipur |***

***| 105 | e | 30 | 50000 | 50 | sales | jaipur |***

***2. write down the number of rows and columns returned by cartiscian product.***

*Number of rows:25*

*Number of columns:7*

***3. Display ename,dname and dno .***

*select ename,dname,emp.deptno*

*-> from emp,dept****;***

***+-------+-------+--------+***

***| ename | dname | deptno |***

***+-------+-------+--------+***

***| a | fin | 10 |***

***| b | fin | 20 |***

***| c | fin | NULL |***

***| d | fin| 10 |***

***| e | fin | 30 |***

***| a | mkt | 10 |***

***| b | mkt | 20 |***

***| c | mkt | NULL |***

***| d | mkt | 10 |***

***| e | mkt | 30 |***

***| a | admin | 10 |***

***| b | admin | 20 |***

***| c | admin | NULL |***

***| d | admin | 10 |***

***| e | admin | 30 |***

***| a | res | 10 |***

***| b | res | 20 |***

***| c | res | NULL |***

***| d | res | 10 |***

***| e | res | 30 |***

***| a | sales | 10 |***

***| b | sales | 20 |***

***| c | sales | NULL |***

***| d | sales | 10 |***

***| e | sales | 30 |***

***+-------+-------+--------+***

***4. display ename,dno,location where sal is between 1000 and 40000***

*select ename,location,emp.deptno*

*-> from emp,dept*

*-> where sal>1000 and sal<4000;*

***Empty set (0.10 sec)***

***5. Display ename,dname and dno  WHERE SAL IS>30000 AND LOCATION IS NM.***

*select ename,dname,emp.deptno*

*-> from emp,dept*

*-> where sal>30000 and location='nm';*

***+-------+-------+--------+----------+***

***| ename | dname | deptno |***

***+-------+-------+--------+-----------+***

***| e | fin | 30 |***

***| e | admin | 30 |***

***+-------+-------+--------+-----------+***

***2 rows in set (0.00 sec)***

***6. display ename,dno,location where sal is between 1000 and 40000.ARRANGE SAL IN ASCENDING ORDER.***

*select ename,location,emp.deptno*

*-> from emp,dept*

*-> where sal>1000 and sal<40000 order by sal asc;*

***+-------+----------+--------+***

***| ename | location | deptno |***

***+-------+----------+--------+***

***| d | jaipur | 10 |***

***| d | nm | 10 |***

***| d | jaipur | 10 |***

***| d | delhi | 10 |***

***| d | nm | 10 |***

***| a | jaipur | 10 |***

***| a | nm | 10 |***

***| a | jaipur | 10 |***

***| a | delhi | 10 |***

***| a | nm | 10 |***

***| b | nm | 20 |***

***| b | jaipur | 20 |***

***| b | nm | 20 |***

***| b | jaipur | 20 |***

***| b | delhi | 20 |***

***| c | jaipur | NULL |***

***| c | delhi | NULL |***

***| c | nm | NULL |***

***| c | jaipur | NULL |***

***| c | nm | NULL |***

***7. Display ename,dname and dno  WHERE SAL IS <=30000 AND LOCATION IS NM  AND DEPT IS 10 OR 20.***

select ename,dname,emp.deptno

-> from emp,dept

-> where sal<=30000 and location='nm' and emp.deptno=10 or emp.deptno=20;

***+-------+-------+--------+***

***| ename | dname | deptno |***

***+-------+-------+--------+***

***| a | fin | 10|***

***| b | fin | 20 |***

***| d | fin | 10 |***

***| b | mkt | 20 |***

***| a | admin | 10 |***

***| b | admin | 20 |***

***| d | admin | 10 |***

***| b | res | 20 |***

***| b | sales | 20 |***

***8. Display ALL DETAILS FROM BOTH THE TABLES  USING NATURAL JOIN***

*select \* from emp natural join dept;*

***+--------+-------+-------+-------+-------+----------+***

***| deptno | empno | ename | sal | dname | location |***

***+--------+-------+-------+-------+-------+----------+***

***| 10 | 101 | a | 10000 | fin | nm |***

***| 10 | 104 | d | 5000 | fin | nm |***

***| 20 | 102 | b | 20000 | mkt | delhi |***

***| 30 | 105 | e | 50000 | admin | nm |***

***+--------+-------+-------+-------+-------+----------+***

***9. create an index on column ename.***

*create index ename on emp(ename);*

***Query OK, 0 rows affected (0.49 sec)***

***10. create an index on column dname and location***

*create index dname on dept(dname);*

***Query OK, 0 rows affected (0.28 sec)***

*create index location on dept(location);*

***Query OK, 0 rows affected (0.23 sec)***

***11. drop the index on column ename.***

*alter table emp*

*-> drop index ename;*

***Query OK, 0 rows affected (0.22 sec)***

***Records: 0 Duplicates: 0 Warnings: 0***

***12. create a new table employee from the existing table emp .***

*create table employe*

*-> LIKE emp;*

***Query OK, 0 rows affected (0.24 sec)***

***13. display all the constraints of table emp***

*show create table emp;*

***+-------+--------------------------------------------------------------------------------------+***

***| Table | Create Table***

***+-------+----------------------------------------------------------------------------------------+***

***| emp | CREATE TABLE `emp` (***

***`empno` int(5) NOT NULL,***

***`ename` varchar(20) DEFAULT NULL,***

***`deptno` int(11) DEFAULT NULL,***

***`sal` int(11) DEFAULT NULL,***

***PRIMARY KEY (`empno`),***

***KEY `deptno` (`deptno`),CONSTRAINT `emp\_ibfk\_1` FOREIGN KEY (`deptno`) REFERENCES `dept` (`deptno`)***

***14. create a new table depaartment with dnalocation from already existing table dept.***

*create table department AS*

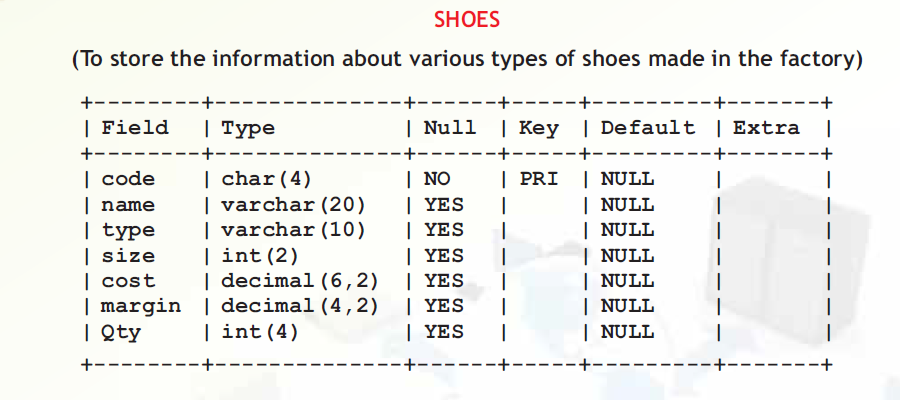
*-> SELECT location from dept****;***

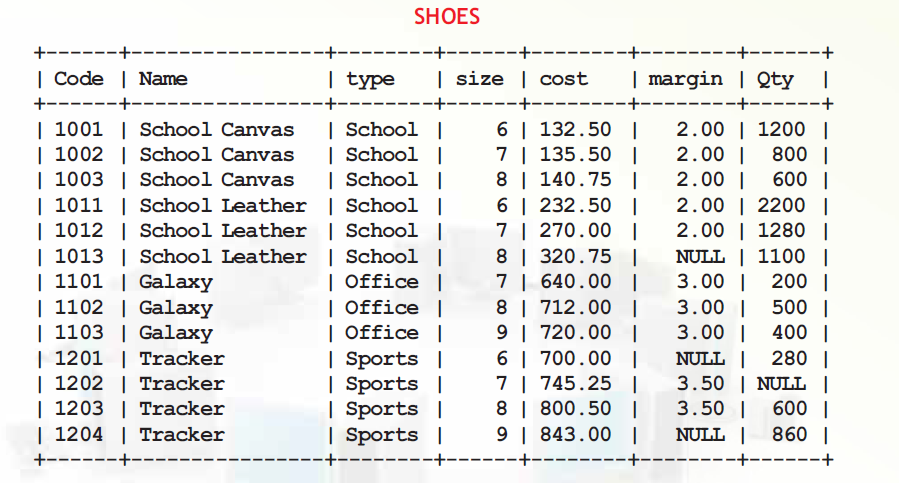
***Query OK, 5 rows affected (0.32 sec)***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***PROGRAM 20***

**Write SQL query for the following: use column alias for the expression**

****

****

mysql> create table shoes

-> (code char(4) primary key,

-> name varchar(20),

-> type varchar(10),

-> size int(2),

-> cost decimal(6,2),

-> margin decimal(4,2),

-> qty int(4));

mysql> insert into shoes values('1001','school canvas','school',6,132.50,2.00,1200);

mysql> insert into shoes values('1002','school canvas','school',7,135.50,2.00,800);

mysql> insert into shoes values('1003','school canvas','school',7,135.50,2.00,800);

mysql> insert into shoes values('1011','school leather','school',6,232.50,2.00,2200);

mysql> insert into shoes values('1012','school leather','school',7,270.00,2.00,1280);

mysql> insert into shoes values('1013','school leather','school',8,320.75,null,1100);

mysql> insert into shoes values('1101','galaxy','office',7,640.00,3.00,200);

mysql> insert into shoes values('1102','galaxy','office',8,712.00,3.00,500);

mysql> insert into shoes values('1103','galaxy','office',9,720.00,3.00,400);

mysql> insert into shoes values('1201','tracker','sports',6,700.00,null,280);

mysql> insert into shoes values('1202','tracker','sports',7,745.25,3.50,null);

mysql> insert into shoes values ('1203','tracker','sports',8,800.50,3.50,600);

mysql> insert into shoes values('1204','tracker','sports',9,843.00,null,860);

**1.Display the type, minimum,maximum,average margin of each type of shoes.**

.mysql> select type,min(margin),max(margin),avg(margin)

-> from shoes

-> group by type;

**+--------+-------------+-------------+-------------+**

**| type | min(margin) | max(margin) | avg(margin) |**

**+--------+-------------+-------------+-------------+**

**| office | 3.00 | 3.00 | 3.000000 |**

**| school | 2.00 | 2.00 | 2.000000 |**

**| sports | 3.50 | 3.50 | 3.500000 |**

**+--------+-------------+-------------+-------------+**

**2.Display type and total quantity of each type of shoes.Arrange it by total quantity in descending order.**

.mysql> select type,sum(qty)

-> from shoes

-> group by type

-> order by sum(qty) desc;

**+--------+----------+**

**| type | sum(qty) |**

**+--------+----------+**

**| school | 7380 |**

**| sports | 1740 |**

**| office | 1100 |**

**+--------+----------+**

**3. Display type and total quantity of each type of shoes.and display only those type where total qty is**

**more than 1500.**

mysql> select type,sum(qty)

-> from shoes

-> group by type

-> having sum(qty)>1500;

**+--------+----------+**

**| type | sum(qty) |**

**+--------+----------+**

**| school | 7380 |**

**| sports | 1740 |**

**+--------+----------+**

**4. Display type and total quantity of each type of shoes.and display only those type where averagemargin greater than 2.**

.mysql> select type,sum(qty)

-> from shoes

-> group by type

-> having avg(margin)>2;

**+--------+----------+**

**| type | sum(qty) |**

**+--------+----------+**

**| office | 1100 |**

**| sports | 1740 |**

**+--------+----------+**

**5. Display type and total quantity of each type of shoes where size is not equal to 6,and display only**

**those type where total quantity greter than 1500.**

mysql> select type,sum(qty)

-> from shoes

-> where size!=6

-> group by type

-> having sum(qty)>1500;

**+--------+----------+**

**| type | sum(qty) |**

**+--------+----------+**

**| school | 3980 |**

**+--------+----------+**

**6. Display the type,size,total count from shoes for each type and with that each size.(group by multiplecolumns)**

mysql> select type,size,sum(qty)

-> from shoes

-> group by type,size;

**+--------+------+----------+**

**| type | size | sum(qty) |**

**+--------+------+----------+**

**| office | 7 | 200 |**

**| office | 8 | 500 |**

**| office | 9 | 400 |**

**| school | 6 | 3400 |**

**| school | 7 | 2880 |**

**| school | 8 | 1100 |**

**| sports | 6 | 280 |**

**| sports | 7 | NULL |**

**| sports | 8 | 600 |**

**| sports | 9 | 860 |**

**+--------+------+----------+**

**7.Display the total stock value(cost\*qty) of each type of shoes. And arrange it in ascending order of Total**

**Value**

mysql> select type,cost\*qty as 'stock value'

-> from shoes

-> group by type

-> order by cost\*qty desc;

**+--------+-------------+**

**| type | stock value |**

**+--------+-------------+**

**| sports | 196000.00 |**

**| school | 159000.00 |**

**| office | 128000.00 |**

**+--------+-------------+**

**8.Display the count of total number of different types of shoes available in shoes table.**mysql> select type,sum(qty)

-> from shoes

-> group by type;

**+--------+----------+**

**| type | sum(qty) |**

**+--------+----------+**

**| office | 1100 |**

**| school | 7380 |**

**| sports | 1740 |**

**+--------+----------+**

***PROGRAM 21***

**MYSQL and PYTHON Interface:Quiz Application**

**Use Python to do the following:**

1. **Connect Python and MYSQL**
2. **create database Quiz**
3. **Activate it**
4. **Show all the databases**
5. **Show all the tables in database**
6. **Create table Question with following fields and constraint**

**Qno int Primary key**

**Ques varchar(100) not null**

**Type varchar(20) can be GK,Maths,CS**

**Opt1 varchar(30) not null**

**Opt2 varchar(30) not null**

**Opt3 varchar(30) not null**

**Opt4 varchar(30) not null**

**Ans varchar(30)**

**7.show all the tables**

**8.create a menu and accept user choice**

**1.Add**

**2.Display**

**If user enters 1 then do the following:**

**Add(): To Add questions in the program.Ask user how many questions to enter. And insert records**

**If user enters 2 then do the following:**

**Display(): Display all the data in the question table.Call the functions.**

**CODE:**

#%%

import mysql.connector as msq  
import random as r  
try:  
    conn=msq.connect(host='localhost',user='root',password='',database='')  
    v=conn.cursor()  
    v.execute('create database if not exists quiz;')  
    v.execute('use quiz;')  
    v.execute('''create table if not exists question  
              (qno int primary key,  
              ques varchar(100) not null,  
              type varchar(20),  
              opt1 varchar(30) not null,  
              opt2 varchar(30) not null,  
              opt3 varchar(30) not null,  
              opt4 varchar(30) not null,  
              ans varchar(30));''')  
except:  
    print('Error')  
def enter():  
    try:  
        conn=msq.connect(host='localhost',user='root',password='',database='')  
        v=conn.cursor()  
        v.execute('use quiz;')  
        v.execute('select count(qno) from question;')  
        for i in v:  
            qn=i[0]+1  
        q=input('Enter a question: ')  
        t=input('Enter type: ')  
        o1=input('Enter option: ')  
        o2=input('Enter option: ')  
        o3=input('Enter option: ')  
        o4=input('Enter option: ')  
        ans=int(input('Enter option no which is correct: '))  
        if ans==1:  
            ans=o1  
        elif ans==2:  
            ans=o2  
        elif ans==3:  
            ans=o3  
        elif ans==4:  
            ans=o4  
        y='''insert into question value(%s,%s,%s,%s,%s,%s,%s,%s)'''  
        val=(qn,q,t,o1,o2,o3,o4,ans)  
        v.execute(y,val)  
        conn.commit()  
        v.execute('select \* from question;')  
        print('Qno Ques Type Opt1 Opt2 Opt3 Opt4 Ans')  
        for i in v:  
            print(i)  
    except:  
        print('Error')  
def disp():  
    try:  
        conn=msq.connect(host='localhost',user='root',password='',database='')  
        v=conn.cursor()  
        v.execute('use quiz;')  
        v.execute('select \* from question;')  
        print('Qno Ques Type Opt1 Opt2 Opt3 Opt4 Ans')  
        for i in v:  
            print(i)  
    except:  
        print('Error')  
def pla():  
    try:  
        conn=msq.connect(host='localhost',user='root',password='',database='')  
        v=conn.cursor()  
        v.execute('use quiz;')  
        v.execute('select count(qno) from question;')  
        for i in v:  
            lq=i[0]  
        qs=r.randint(1,lq)  
        v.execute('''Select \* from question where qno='''+str(qs))  
        for i in v:  
            y=i  
        print('Q.',y[1],)  
        for j in range(4):  
            print(y[j+3])  
        a=int(input('Enter the answer no.: '))  
        if y[a+2]==y[-1]:  
            print('Right Answer')  
        else:  
            print('Wrong answer')  
    except:  
        print('Error')  
print('Hello and welcome to the quiz. What would you like to do?? \n1.Add \n2.Display \n3.Play')  
k=int(input('What would you like to do: '))  
while k!=0:  
    if k==1:  
        enter()  
    elif k==2:  
        disp()  
    elif k==3:  
        pla()  
    k=int(input('What would you like to do: '))

#%%

**Output:**

Hello and welcome to the quiz. What would you like to do??   
1.Add   
2.Display   
3.Play  
  
What would you like to do: 2  
Qno Ques Type Opt1 Opt2 Opt3 Opt4 Ans  
(1, 'Output Device', 'CS', 'speaker', 'mouse', 'keyboard', 'scanner', 'speaker')  
(2, '1+3', 'Maths', '2', '3', '4', '5', '4')  
(3, 'What is the capital of India', 'GK', 'Delhi', 'New York', 'Tokyo', 'Paris', 'Delhi')  
(4, '6/2', 'Maths', '1', '2', '3', '4', '3')  
  
What would you like to do: 1  
  
Enter a question: Input Device  
  
Enter type: Cs  
  
Enter option: Monitor  
  
Enter option: Keyboard  
  
Enter option: speaker  
  
Enter option: printer  
  
Enter option no which is correct: 2  
Qno Ques Type Opt1 Opt2 Opt3 Opt4 Ans  
(1, 'Output Device', 'CS', 'speaker', 'mouse', 'keyboard', 'scanner', 'speaker')  
(2, '1+3', 'Maths', '2', '3', '4', '5', '4')  
(3, 'What is the capital of India', 'GK', 'Delhi', 'New York', 'Tokyo', 'Paris', 'Delhi')  
(4, '6/2', 'Maths', '1', '2', '3', '4', '3')  
(5, 'Input Device', 'Cs', 'Monitor', 'Keyboard', 'speaker', 'printer', 'Keyboard')  
  
What would you like to do: 3  
Q. What is the capital of India  
Delhi  
New York  
Tokyo  
Paris  
  
Enter the answer no.: 1  
Right Answer  
  
What would you like to do: 0

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**PROGRAM 22**

**Accept a list of 10 marks. Do the result analysis and find out how many percentage of studentshave scored the following :**

**LABELS COLOR OF THE SLICE**

**&gt;=95 CYAN**

**90-94 RED**

**80-89 OLIVE**

**70-79 MAGENTA**

**60-69 RED**

**&lt; 60 YELLOW**

**Display the result as a pie chart.**

**Explode &gt;=95 and &lt; 60 slices with shadow**

**Give the corresponding labels as shown above and also display the slice percentage.**

**Give the title PERCENTAGE ANALYSIS OF SCIENCE MARKS .**

**CODE:**

**#%%**

**i**mport matplotlib.pyplot as plt

import numpy as np

l=[]

n=0

m=0

o=0

p=0

q=0

r=0

for i in range(10):

marks=int(input("enter score"))

l.append(marks)

print(l)

for i in l:

if i>95:

n+=1

elif i>90 and i<94:

m+=1

elif i>80 and i< 89:

o+=1

elif i>70 and i< 79:

p+=1

elif i>60 and i< 79:

q+=1

else:

r+=1

labels=['<90','90-94','80-89','70-79','60-69','>60']

count=[n,m,o,p,q,r]

explode=[0.1,0,0,0,0,0.1]

colors=['cyan','red','olive','magenta','red','yellow']

plt.pie(count,labels=labels,colors=colors,explode=explode,shadow=True,autopct='%01.1f%%',startangle=140)

plt.title("PERCENTAGE ANALYSIS OF SCIENCE MARKS . ")

plt.show()

**#%%**

**OUTPUT:**

**enter score98**

**enter score99**

**enter score93**

**enter score85**

**enter score83**

**enter score78**

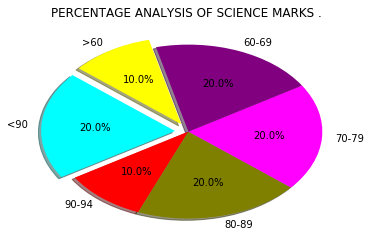
**enter score72**

**enter score65**

**enter score61**

**enter score55**

**[98, 99, 93, 85, 83, 78, 72, 65, 61, 55]**



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**PROGRAM 23**

**The following data represents number of people employed in 2010 and 2011.**

**Create a multiline chart with the following features.**

**Title – PEOPLE EMPLOYED IN THE YEAR 2010 AND 2011**

**X axis label - MONTH , Y axis label - NUMBER OF PEOPLE**

**Xtick labels should be the month names. Increase the line width, give the legend as 2010 and 2011 . Also give different line styles for each years data, put the marker ‘X’ with marker edge color black.**

|  |  |  |
| --- | --- | --- |
| **Month** | **2010** | **2011** |
| Jan | 153454 | 153250 |
| Feb | 153704 | 153302 |
| Mar | 153964 | 153392 |
| Apr | 154528 | 153420 |
| May | 154216 | 153700 |
| Jun | 153653 | 153409 |
| Jul | 153748 | 153358 |
| Aug | 154073 | 153674 |
| Sep | 153918 | 154004 |
| Oct | 153709 | 154057 |
| Nov | 154041 | 153937 |
| Dec | 153613 | 153887 |

**CODE:**

#%%

import matplotlib.pyplot as plt

import numpy as np

months=['jan','feb','march','april','may','june','july','aug','sept','oct','nov','dec']

a=[153454,153704,153964,154528,154216,153748,153748,154073,153918,153709,154041,153613]

b=[153250,153302,153392,153420,153700,153409,153358,153674,154004,154057,153937,153887]

plt.plot(months,a,color='red',linestyle='dotted',linewidth=2,marker='X',markerfacecolor='k',label="2010")

plt.plot(months,b,color='blue',linewidth=3,marker='X',markerfacecolor='black',label="2011")

plt.xlabel('MONTHS')

plt.ylabel('NUMBER OF PEOPLE')

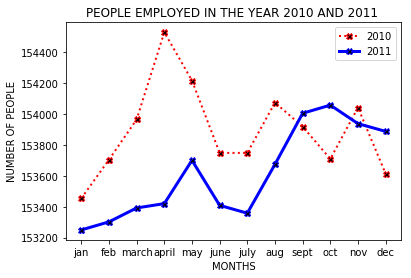
plt.title("PEOPLE EMPLOYED IN THE YEAR 2010 AND 2011 ")

plt.legend()

plt.show()

**#%%**

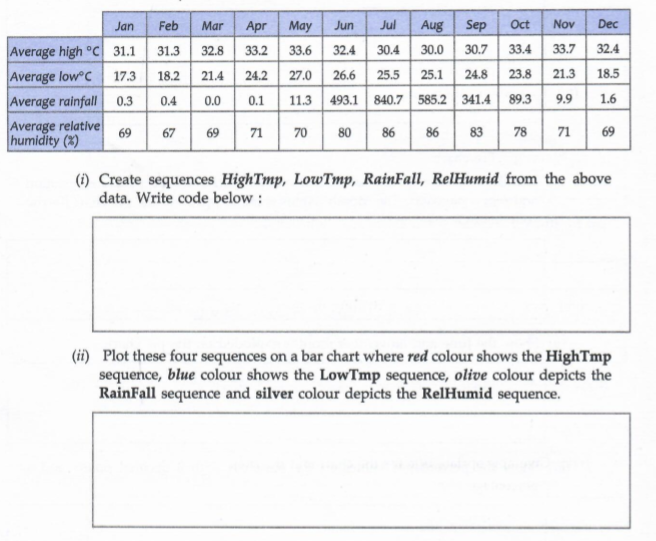
**OUTPUT:**



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**PROGRAM 24**

****

**(iii) Xticks should be the month names,   
 title should be “Average data for Mumbai city”**

**Width of the bars should be 0.1**

**(iv) The legend should show Hightemp , Lowtemp, Rainfall, Relhumid**

**CODE:**

import matplotlib.pyplot as plt

import numpy as np

months=['jan','feb','march','april','may','june','july','aug','sept','oct','nov','dec']

a=[31.1,31.3,32.8,33.2,33.6,32.8,30.4,30.0,30.7,33.4,33.7,32.4]

b=[17.3,18.2,21.4,24.2,27.0,26.6,25.5,25.1,24.8,23.8,21.3,18.5]

c=[0.3,0.4,0.0,0.1,11.3,493.1,840.7,585.2,341.4,89.3,9.9,1.6]

d=[69,97,69,71,70,80,86,86,83,78,71,69]

index=np.arange(len(months))

width=0.2

plt.bar(index,a,width,color='r',label="Hightemp")

plt.bar(index + 0.2,b,width,color='blue',label="lowtemp")

plt.bar(index + 0.4,c,width,color='olive',label="rainfall")

plt.bar(index + 0.6,d,width,color='k',label="Relhumid")

plt.xlabel('MONTHS')

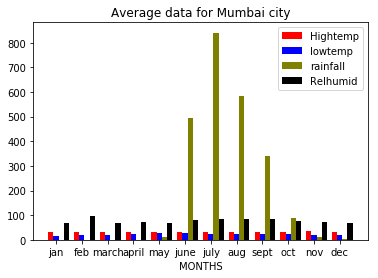
plt.title("Average data for Mumbai city")

plt.legend()

plt.xticks(index+width,('jan','feb','march','april','may','june','july','aug','sept','oct','nov','dec'))

plt.show()

***OUTPUT:***



***PROGRAM 26(DATA STUCTURE)***

***1. Write a menu driven program to do the following in a numerical list.(using Traditional Method)***

***MAIN MENU***

***1. Accept a list***

***2. Display the list***

***3. Sort the list in descending order***

***4. Insert an element***

***5. Delete an element***

***6. Search for an element***

***7. Exit***

***Note:***

***1. Write functions for all options except option 7***

***2. Insertion sort to be used for option 3***

***3. Bisect method SHOULD NOT BE USED for option 4***

***4. Binary search to be used for option 6***

***Code:***

*def inplist() :*

*n=int(input(&quot;Enter no of elements :&quot;))*

*for i in range(n) :*

*x=int(input(&quot;Enter number : &quot;))*

*lst.append(x)*

*Home()*

*def display(x) :*

*print(&quot;The list is : &quot;,x)*

*Home()*

*def sortdesc(arr) :*

*for i in range(1, len(arr)):*

*key = arr[i]*

*j = i-1*

*while j &gt;=0 and key &lt; arr[j] :*

*arr[j+1] = arr[j]*

*j -= 1*

*arr[j+1] = key*

*Home()*

*def insel(x) :*

*n=int(input(&quot;Enter position to add : &quot;))*

*el=int(input(&quot;Enter element : &quot;))*

*x.insert(n,el)*

*print(x)*

*def delel(x):*

*el=int(input(&quot;Enter element : &quot;))*

*x.remove(el)*

*print(x)*

*Home()*

*def searchel(x) :*

*y=int(input(&quot;Enter item to be searched : &quot;))*

*index=len(x)*

*for i in range(len(x)):*

*if x[i] &gt; y :*

*index = i*

*break*

*print(index)*

*Home()*

*def Home() :*

*print(&quot; MAIN MENU &quot;)*

*print(&quot;1. Accept a list&quot;)*

*print(&quot;2. Display the list&quot;)*

*print(&quot;3. Sort the list in descending order&quot;)*

*print(&quot;4. Insert an element &quot;)*

*print(&quot;5. Delete an element &quot;)*

*print(&quot;6. Search for an element&quot;)*

*print(&quot;7. Exit&quot;)*

*inp = int(input(&quot;Enter your choice : &quot;))*

*if inp==1 :*

*inplist()*

*elif inp==2 :*

*display(lst)*

*elif inp==3 :*

*sortdesc(lst)*

*elif inp==4 :*

*insel(lst)*

*elif inp==5 :*

*delel(lst)*

*elif inp==6 :*

*searchel(lst)*

*Home()*

***OUTPUT:***

***MAIN MENU***

***1. Accept a list***

***2. Display the list***

***3. Sort the list in descending order***

***4. Insert an element***

***5. Delete an element***

***6. Search for an element***

***7. Exit***

***Enter your choice : 1***

***Enter no of elements :3***

***Enter number : 2***

***Enter number : 5***

***Enter number : 1***

***MAIN MENU***

***1. Accept a list***

***2. Display the list***

***3. Sort the list in descending order***

***4. Insert an element***

***5. Delete an element***

***6. Search for an element***

***7. Exit***

***Enter your choice : 2***

***The list is : [2, 5, 1]***

***MAIN MENU***

***1. Accept a list***

***2. Display the list***

***3. Sort the list in descending order***

***4. Insert an element***

***5. Delete an element***

***6. Search for an element***

***7. Exit***

***Enter your choice : 3***

***MAIN MENU***

***1. Accept a list***

***2. Display the list***

***3. Sort the list in descending order***

***4. Insert an element***

***5. Delete an element***

***6. Search for an element***

***7. Exit***

***Enter your choice : 2***

***The list is : [1, 2, 5]***

***MAIN MENU***

***1. Accept a list***

***2. Display the list***

***3. Sort the list in descending order***

***4. Insert an element***

***5. Delete an element***

***6. Search for an element***

***7. Exit***

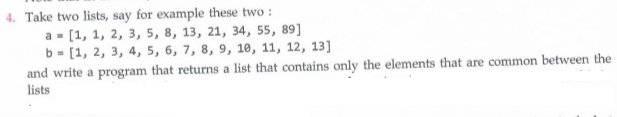
***Enter your choice : 4***

***Enter position to add : 2***

***Enter element : 8***

***[1, 2, 8, 5]***

***PROGRAM 27***

******

***CODE:***

a=[]  
b=[]  
common=[]  
num1=int(input("Enter the number of elements for the first list "))  
num2=int(input("Enter the number of elements for the 2nd list "))

for i in range(num1):     
    inp1=int(input("Enter the input for the 1st list "))  
    a.append(inp1)  
for i  in range(num2):  
      inp2=int(input("Enter the input for the 2nd list "))  
      b.append(inp2)  
for i in a:  
    if i in b:  
        common.append(i)  
print("The list containing common elements is", common)

***OUTPUT:***

***Enter the number of elements for the first list 3***

***Enter the number of elements for the 2nd list 4***

***Enter the input for the 1st list 1***

***Enter the input for the 1st list 2***

***Enter the input for the 1st list 3***

***Enter the input for the 2nd list 4***

***Enter the input for the 2nd list 2***

***Enter the input for the 2nd list 5***

***Enter the input for the 2nd list 6***

***The list containing common elements is [2]***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***PROGRAM 28(STACKS ANS QUEUES)***

***Main menu***

***1.display()***

***2.pop()***

***3.peek()***

***4.push()***

***CODE:***

def push(l):

a=int(input('Enter element to be pushed>>>'))

l.append(a)

print('Your element '+str(a)+' has been pushed.')

print()

def popp(l):

if len(l)==0:

print('Empty list.')

a=input('Do you want to enter element(y/n)?')

if a=='n':

pass

elif a=='y':

push(l)

else:

print('Invalid input.')

print()

else:

print('Last element '+str(l[-1])+' has been popped.')

print()

l.pop()

def peek(l):

if len(l)==0:

print('Empty list.')

a=input('Do you want to enter element(y/n)?')

if a=='n':

pass

elif a=='y':

push(l)

else:

print('Invalid input.')

print()

else:

print('Element at peak:'+str(l[-1]))

print('Index:'+str(len(l)-1))

print()

def display(l):

if len(l)==0:

print(l)

print("Your list is empty.")

a=input('Do you want to enter element(y/n)?')

if a=='n':

pass

elif a=='y':

push(l)

else:

print('Invalid input.')

print()

else:

print("list",l)

print()

def stacks(l):

while True:

print('1.Push')

print('2.Pop')

print('3.Peek')

print('4.Display stack')

print('5.Exit')

c=int(input("Enter any number:"))

if c==1:

push(l)

elif c==2:

popp(l)

elif c==3:

peek(l)

elif c==4:

display(l)

elif c==5:

break

else:

print('Enter valid input')

l=[]

n2=int(input("enter range"))

for i in range(n2):

n3=int(input("enter elements"))

l.append(n3)

stacks(l)

***OUTPUT:***

***enter range5***

***enter elements1***

***enter elements2***

***enter elements3***

***enter elements4***

***enter elements5***

***1.Push***

***2.Pop***

***3.Peek***

***4.Display stack***

***5.Exit***

***Enter any number:1***

***Enter element to be pushed>>>6***

***Your element 6 has been pushed.***

***1.Push***

***2.Pop***

***3.Peek***

***4.Display stack***

***5.Exit***

***Enter any number:2***

***Last element 6 has been popped.***

***1.Push***

***2.Pop***

***3.Peek***

***4.Display stack***

***5.Exit***

***Enter any number:3***

***Element at peak:5***

***Index:4***

***1.Push***

***2.Pop***

***3.Peek***

***4.Display stack***

***5.Exit***

***Enter any number:4***

***list [1, 2, 3, 4, 5]***

***1.Push***

***2.Pop***

***3.Peek***

***4.Display stack***

***5.Exit***

***Enter any number:5***

***RECURSIONS***

***1.Write a function that takes a number and tests if it’s a prime number or not.***

***CODE:***

def prime2(a,n):

if n==1:

return print("prime")

elif a%n!=0:

return prime2(a,n-1)

elif a%n==0:

return print("not prime")

a=int(input('Enter number:'))

prime2(a,a//2)

***OUTPUT:***

***Enter number:8***

***not prime***

***2.Write a fuction that multiplies two numbers using ‘+’ and ‘-’ operator.***

***CODE:***

def product(a,b):

if b==1:

return a

return a+product(a,b-1)

product(-2,8)

***OUTPUT:***

***-16***

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

***3.HAILSTONE NUMBER:***

***CODE:***

def hails(n):

if n%2==0:

print(int(n),end=",")

return hails(n/2)

elif n==1:

print(int(n))

return

elif n%2==1:

print(int(n),end=",")

return hails((3\*n)+1)

hails(9)

***OUTPUT:***

9,28,14,7,22,11,34,17,52,26,13,40,20,10,5,16,8,4,2,1

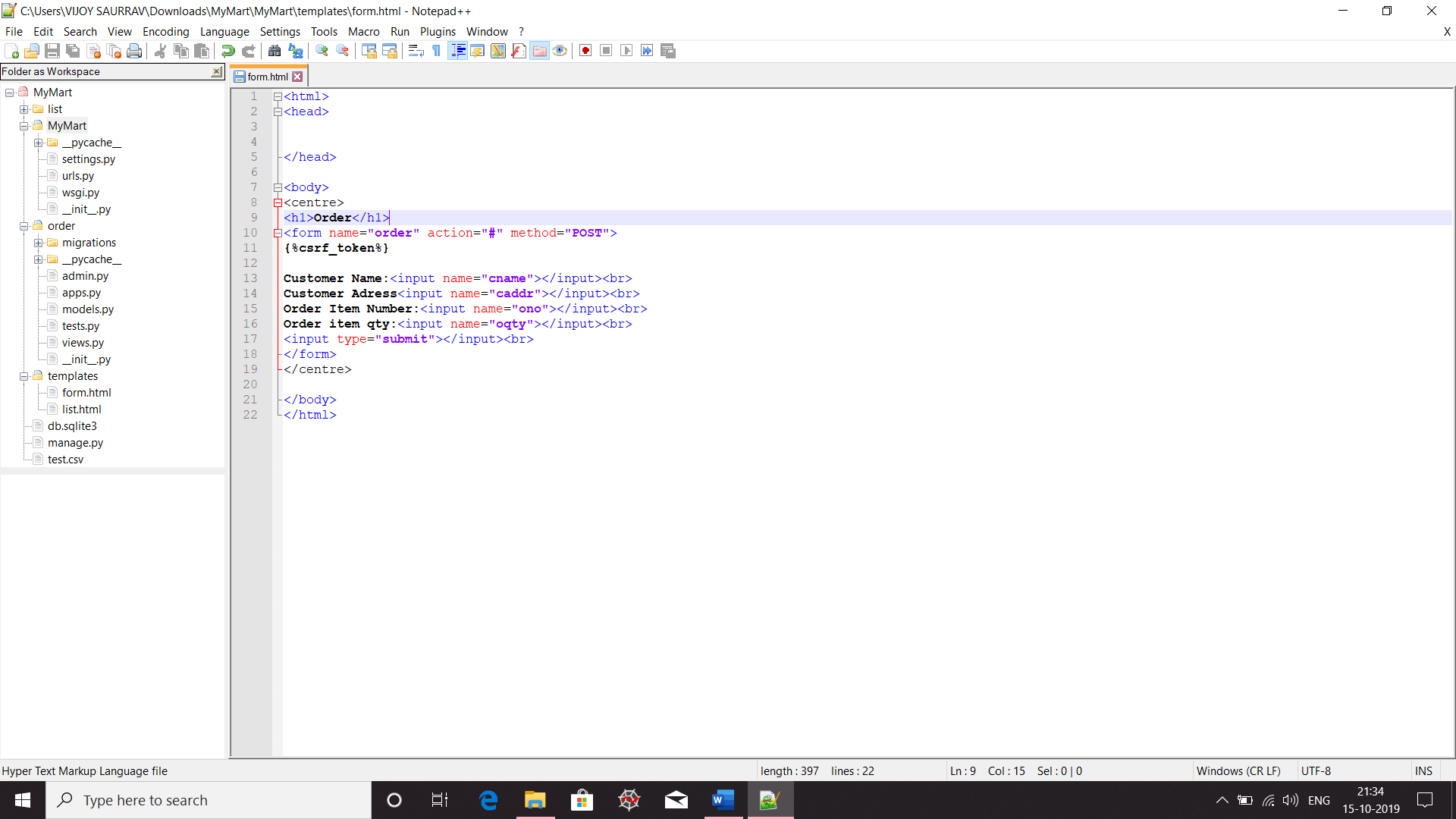
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***DJANGO PROGRAM***

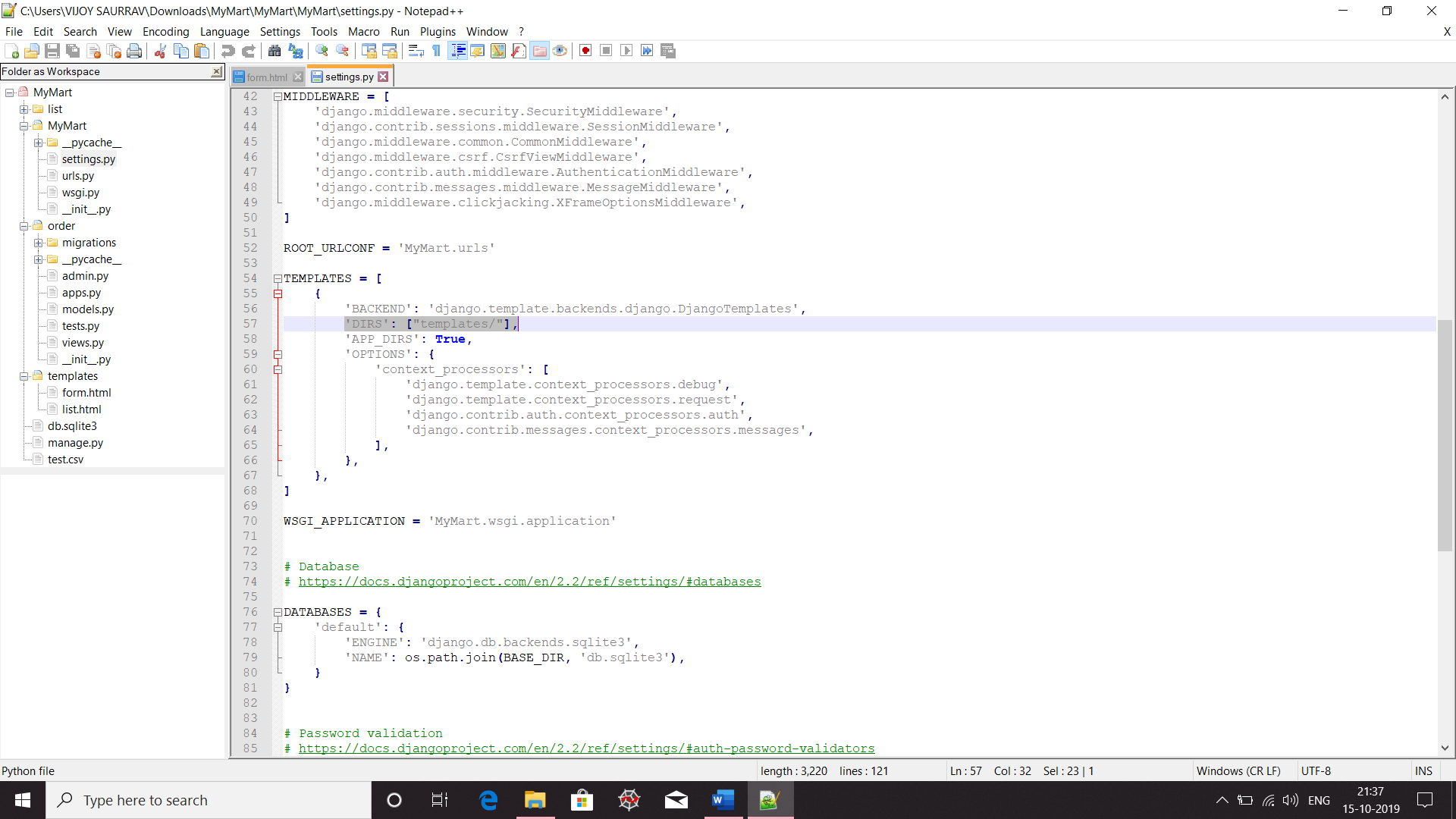
**Create a project MyMart with apps List and Order .**

***CODE* :**

**1.The Hierarchy of Files -**



**2.settings.py file –** addition of the templates in DIRS



**3.The Order.html File -**

<html>

<head>

<title> order </title>

</head>

<body>

<center><h1><font color =Purple> order </font></h1></centre><br>

<form action="#" method="post">

Customer name : <input name="custname" type="text" value=""> <br><br>

Customer Address : <input name="custadd" type="text" value=""><br><br>

Order Item Number : <input name="ordno" type="text" value=""><br><br>

Order Item Qty : <input name="ordqty" type="text" value=""><br><br>

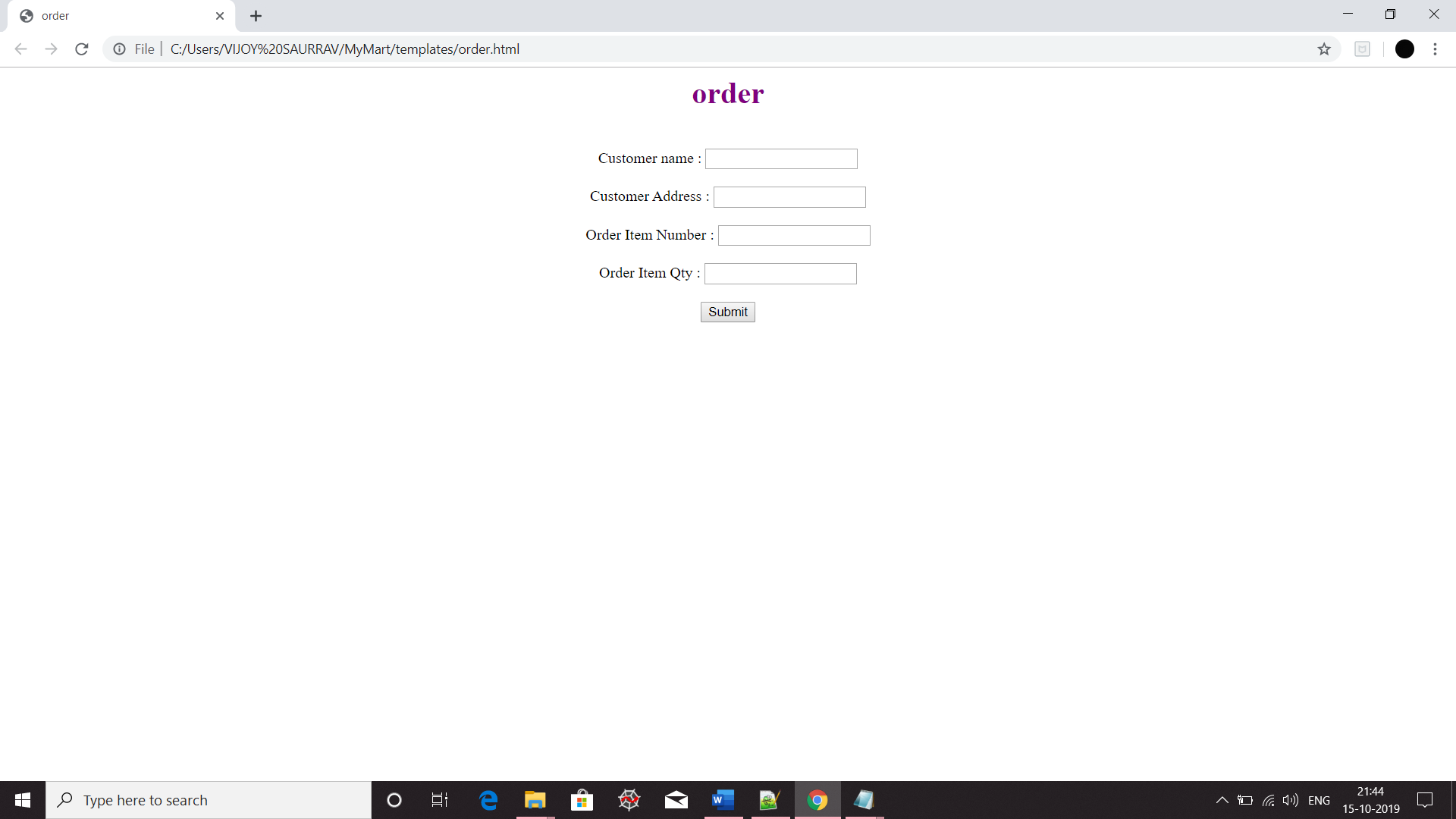
<input name="sbmt" type="submit" value="Submit" ><br><br>

</form>

</body>

</html>

**The Web-page -**



**The Views.py File –**

from django.shortcuts import render

from django.http import HttpResponse

import csv

# Create your views here.

def order(request):

d=request.POST

with open("test.csv","a") as csvfile:

wrt=csv.writer(csvfile)

for k,v in d.items():

wrt.writerow([k,v])

return render(request,"form.html")

**4.The list.html File –**

<html>

<head>

<title> list </title>

</head>

<body>

<center><h1><font color =Purple> Items List</font></h1></centre>

<h4><font color = Black> 1.Bath Tubs </font></h4>

<h4><font color = Black> 2.Battery Charger </font></h4>

<h4><font color = Black> 3.Battery Eliminator </font></h4>

<h4><font color = Black> 4.Beam Scales </font></h4>

<h4><font color = Black> 5.Belt Leather and Straps </font></h4>

<h4><font color = Black> 6.Brass Wire </font></h2>

<h4><font color = Black> 7.Bituminous Paints </font></h4>

<h4><font color = Black> 8.Blotting Paper </font></h4>

<h4><font color = Black> 9.Bolts and Nuts </font></h4>

<h4><font color = Black> 10.Bolts Sliding </font></h4>

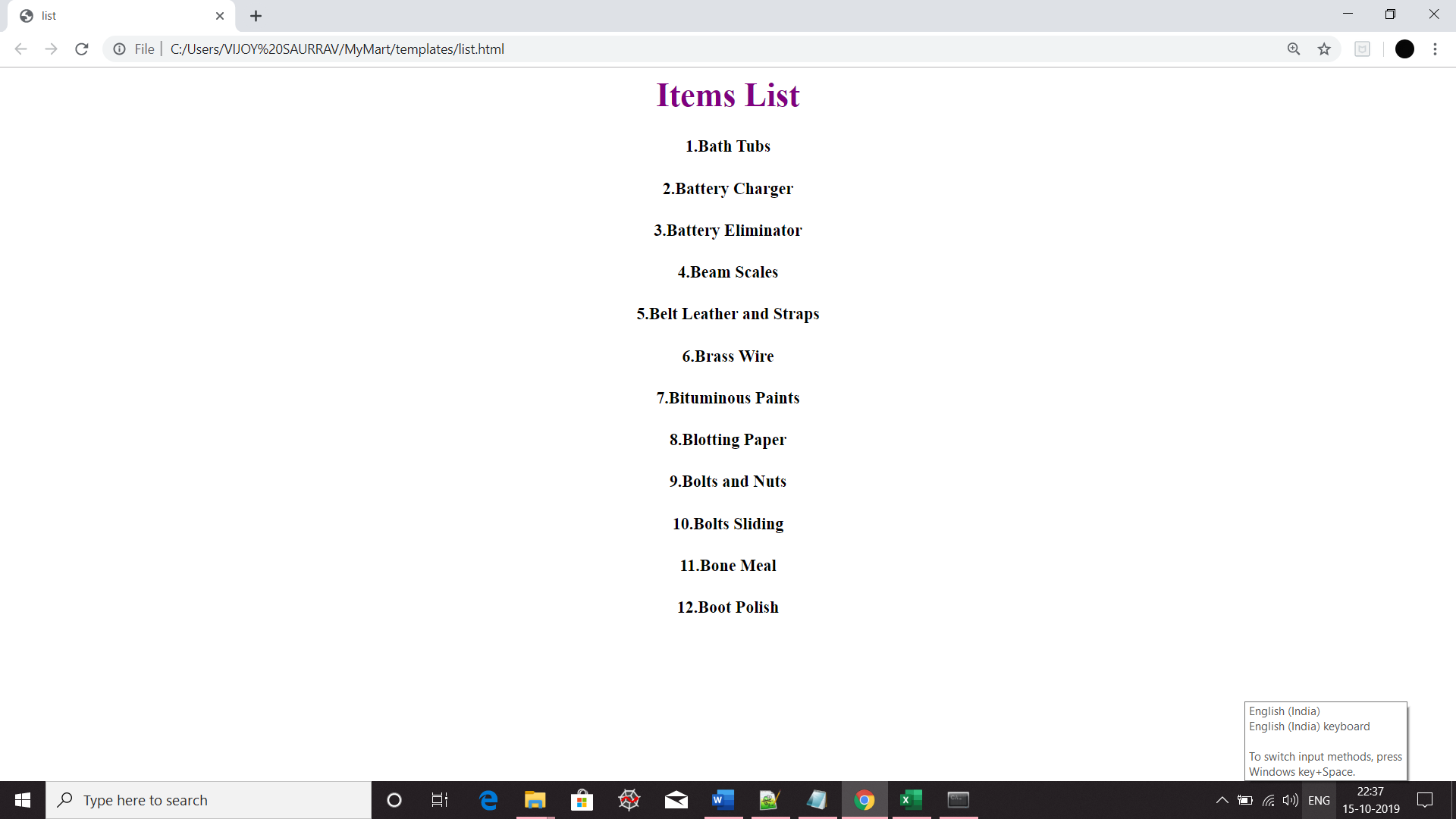
<h4><font color = Black> 11.Bone Meal </font></h4>

<h4><font color = Black> 12.Boot Polish </font></h4>

</body>

</html>

**The Web-page –**

****

**Views.py File –**

from django.shortcuts import render

# Create your views here.

def listd(request):

return render(request,"list.html")

**5.The urls.py File-**

from django.contrib import admin

from django.urls import path

from list import views

from order import views as v

urlpatterns = [

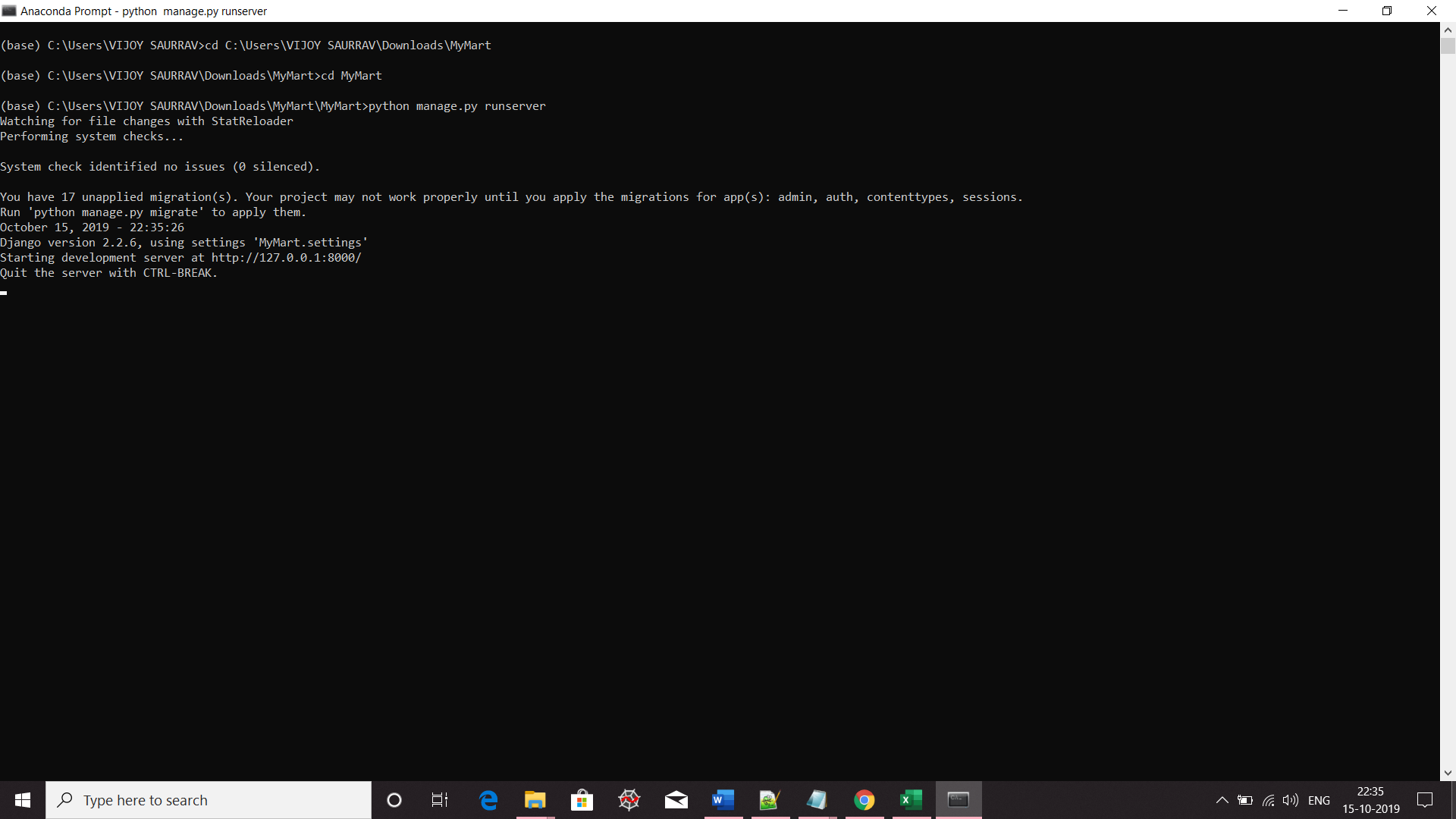
path('admin/', admin.site.urls),

path('list/',views.listd),

path('order/',v.order)

]

**6. Anaconda Prompt -**



***Output* –**

