Computer

Science

Practical

File

By: Sarthak Singhania

Class: XII-D

Index

1. A website requires the users to input username and password to register. You have to write a program to check the validity of password input by users based on the following criteria. Take in a list of passwords.
2. Write a program to input a number and then call the functions:
   1. count(n) which returns the number of digits
   2. reverse(n) which returns the reverse of a number
   3. hasdigit(n) which returns True if the number has a digit else False
   4. show(n) to show the number as sum of place values of the digits of the number.
3. A Number is a perfect number if the sum of all the factors of the number (including 1) excluding itself is equal to number.Write functions
   1. Generatefactors() to populate a list of factors
   2. isPrimeNo() to check whether the number is prime number or not
   3. isPerfectNo() to check whether the number is perfect number or not
4. Pascal’s triangle is a number triangle with numbers arranged in staggered rows such that anr=n!/r!(n−r)!

This equation is the equation for a binomial coefficient.

Write a UDF and a Recursive function in Python to print the Pascal Triangle

1. Input number in decimal and desired type (Specify B for Binary, O for Octal, H for Hexadecimal) for output. Write a program using UDF to perform the conversions.
2. Take in the population of 10 states in India and plot a population density pie graph showing each state with different colour.
3. Use matplotlib.pyplot.plot to produce a plot of the functions f(x) = e−x/10 sin(πx) and g(x) = xe−x/3 over the interval [0, 10]. Include labels for the x- and y-axes, and a legend explaining which line is which plot. Save the plot as a .jpg (“Jpeg”) file
4. Write a program to take two lists from a user and write functions to do the following:
5. Return a sorted merged list from the above list
6. Prints sum of common elements from both the lists.
7. Returns True if the two lists provided by user are circularly identical or False if not.
8. Create a graphical application for Simple Interest Calculator that accepts user inputs for P, R and T.Calculate Simple Interest writes the output using a message box on the screen. Use the tkinter library.
9. Write a program to input a list and write the function for the following:
10. To sort list using bubble sort and find efficiency
11. To search an element using binary search and find efficiency
12. To search an element using linear search and find efficiency
13. Write a function to create a text file containing following data:

*Neither apple nor pine are in pineapple. Boxing rings are square.*

*Writers write, but fingers don’t fing. Overlook and oversee are opposites.*

*A house can burn up as it burns down. An alarm goes off by going on.*

1. Read back the entire file content using read() or readlines() and display on the screen.
2. Append more text of your choice in the file and display the content of file with line numbers prefixed to line.
3. Display last line of file.
4. Display first line from 10th character onwards.
5. Read and display a line from the file. Ask user to provide the line number to be read.
6. Find the frequency of words beginning with every letter.
7. Assume that a text file named file1.txt contains some text, write a function named isvowel( ) that reads the file file1.txt and creates a new file named file2.txt, which shall contain only those words from the file file1.txt which don’t start with a vowel
8. A file containing data about a collection of students has the following format. Each line contains a first name, a second name, a registration number, no of years and a department separated by tabs.
9. Write a Python program that will copy the contents of the file into a list of tuples
10. Display full details of the student sorted by registration number
    1. The names of all students with no of year less than 3
    2. The number of people in each department
11. Write is a program that reads a file “myfile.txt” and builds a histogram (a dictionary having key value pair as word: occurrence) of the words in the file.
    1. Now use histogram to print
       * + 1. Total number of words
           2. Number of different words
           3. The most common words
    2. Using above text file “myfile.txt”, write a program that maps a list of words read from the file to an integer representing the length of the corresponding words. Now using above dictionary design a function find\_longest\_word() to display a list of longest words from file. Define a function filter\_long\_words(n) that takes an integer n and returns the list of words that are longer than n from file. Using above function create another file “newfile.txt” by filtering out all words returned by function filter\_long\_words(8)
12. In cryptography, a *Caesar cipher* is a very simple encryption techniques in which each letter in the plain text is replaced by a letter some fixed number of positions down the alphabet. Select suitable option from user whether he wishes to encode/decode a text file. Read input file and create output file after encoding / decoding it using the dictionary created.
13. Create a stack to take in stack of numbers and then simulate a ring game.
14. Create a program to take in a list **reg\_no, Name,admission\_to\_class (Nursery, KG, I) and add member functions to** 
    1. Add data to the queue.
    2. Display length of the queue.
    3. Print a report showing number of applications received for admission to each class.
15. MySQL
16. Django

**Code:**

passlist=[]

while True:

password=input('Enter your password:')

listpass=list(password)

caps=0

num=0

char=0

small=0

space=0

for x in listpass:

if ord(x) in range(97,123):

small+=1

elif ord(x) in range(65,91):

caps+=1

elif ord(x) in range(48,58):

num+=1

elif x in ['$','#','@']:

char+=1

elif x==' ':

space+=1

if caps>0 and num>0 and char>0 and small>0 and space==0:

passlist.append(password)

print(passlist)

**Output:**

['ABd1234@1']

**Code:**

def count(a):

cnt=0

while a!=0:

a//=10

cnt+=1

return cnt

def reverse(n):

return int(n[::-1])

def hasdigit(n):

num=input('Enter the number you want to check:')

return num in n

def show(n):

a=count(int(n))

b=list(n)

sum=n+ ' = '

for x in b:

a-=1

if a==0:

break

sum+=str(int(x)\*(10\*\*a))

sum+=' + '

sum+=x

return sum

cont='y'

while cont!='n':

option=(int(input('\n\t\t1)Count\n\t\t2)Reverse\n\t\t3)HasDigit\n\t\t4)Show\n\n\t\tEnter the option number:')))

if option==1:

print(count(int(input('Enter the number:'))))

elif option==2:

print(reverse(input('Enter the number:')))

elif option==3:

print(hasdigit(input('Enter the number:')))

elif option==4:

print(show(input('Enter the number:')))

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

cont='n'

**Output:**

1)Count

2)Reverse

3)HasDigit

4)Show

Enter the option number:

* + 1. Enter the number:1000

4

* + 1. Enter the number:1234

4321

* + 1. Enter the number:1

True

* + 1. Enter the number:12345

12345 = 10000 + 2000 + 300 + 40 + 5

**Code:**

def isprimeno(n):

for i in range(2,n):

if n%i==0:

return False

else:

return True

def factors(n):

l=[1]

for i in range(2,n):

if n%i==0:

l.append(i)

return l

def perfect(n):

l=[1]

for i in range(2,n):

if n%i==0:

l.append(i)

if sum(l)==n:

return True

else:

return False

cont='y'

while cont!='n':

option=(int(input('\n\t\t1)Prime Check\n\t\t2)Factors\n\t\t3)Perfect Number\n\n\t\tEnter the option number:')))

if option==1:

print(isprimeno(int(input('Enter a number:'))))

elif option==2:

print(factors(int(input('Enter a number:'))))

elif option==3:

print(perfect(int(input('Enter a number:'))))

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

cont='n'

**Output:**

1)Prime Check

2)Factors

3)Perfect Number

Enter the option number:

* + 1. Enter a number:13

True

* + 1. Enter a number:6

[1,2,3]

* + 1. Enter a number:6

True

**Code:**

def pascal(col,row):

if col==0 or col==row:

return 1

else:

return pascal(col-1,row-1) + pascal(col,row-1)

def PascalTriangle(num):

for r in range(num):

for c in range(r+1):

print(str(pascal(c,r)),end=' ')

print('\n')

PascalTriangle(10)

**Output:**

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

1 5 10 10 5 1

1 6 15 20 15 6 1

1 7 21 35 35 21 7 1

1 8 28 56 70 56 28 8 1

1 9 36 84 126 126 84 36 9 1

**Code:**

def binary(a):

b=''

while a!=0:

b+=str(a%2)

a//=2

return b[::-1]

def octahedral(n):

a=''

while n>0:

remainder=n%8

n//=8

a=str(remainder)+a

return a

def hexadecimal(n):

def hex1(n):

a=''

while n>0:

h=n%16

n//=16

a+=hex2(h)

print(a[::-1])

def hex2(n1):

if n1<10:

return str(n1)

if n1==10:

return "a"

if n1==11:

return "b"

if n1==12:

return "c"

if n1==13:

return "d"

if n1==14:

return "e"

if n1==15:

return "f"

hex1(n)

cont='y'

while cont!='n':

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

option=(input('Desired type:'))

if option=='B' or option=='b':

print(binary(number))

elif option=='O' or option=='o':

print(octahedral(number))

elif option=='H' or option=='h':

hexadecimal(number)

if input('\nDo you want to continue(y/n)?')=='n':

cont='n'

**Output:**

1. Enter a number:1000

Desired type:B

1111101000

1. Enter a number:1000

Desired type:O

1750

1. Enter a number:1000

Desired type:H

3e8

**Code:**

import matplotlib.pyplot as plt

labels=[]

sizes=[]

for i in range(1,11):

print('Entry number',i)

labels.append(input('Enter the name of the country:'))

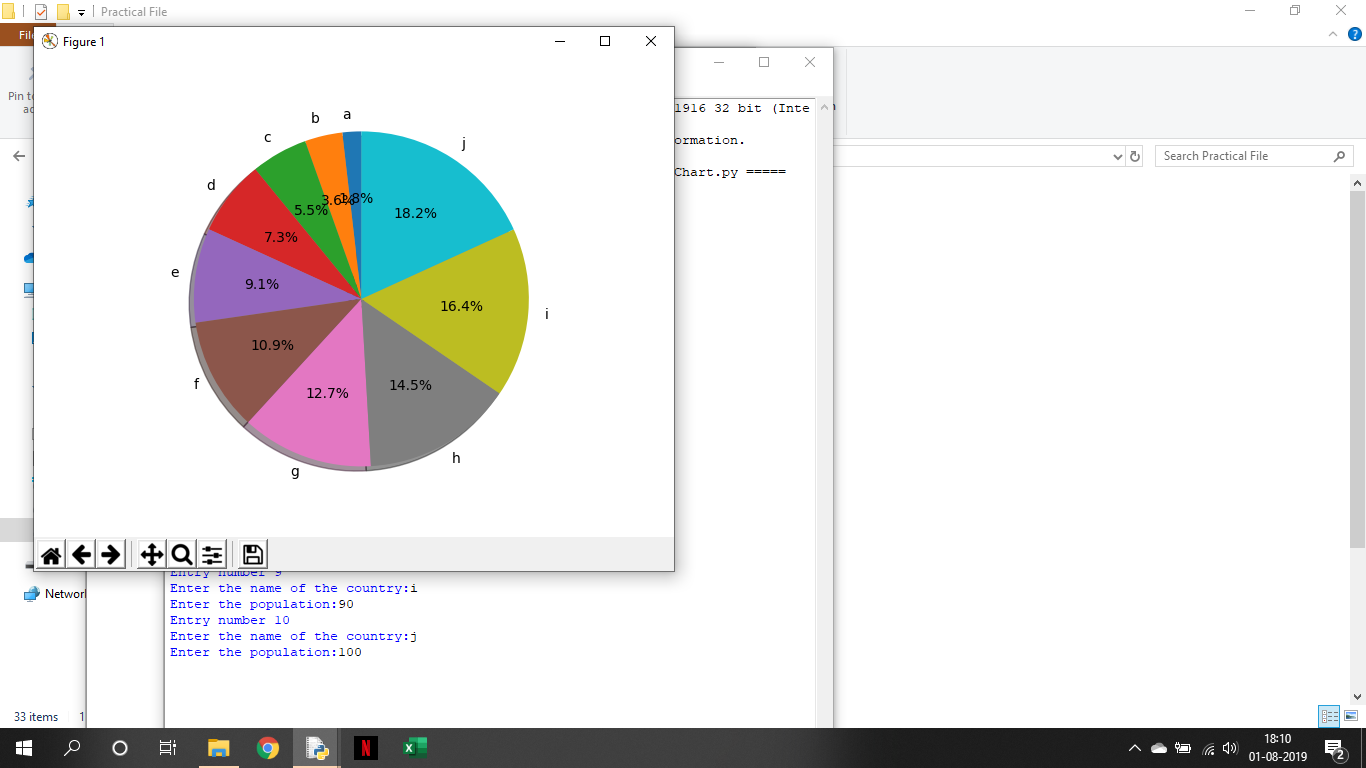
sizes.append(int(input('Enter the population:')))

fig1,ax1=plt.subplots()

ax1.pie(sizes,labels=labels,autopct='%1.1f%%',shadow=True,startangle=90)

ax1.axis('equal')

plt.show()

**Output:**

Entry number 1

Enter the name of the country:a

Enter the population:10

Entry number 2

Enter the name of the country:b

Enter the population:20

Entry number 3

Enter the name of the country:c

Enter the population:30

Entry number 4

Enter the name of the country:d

Enter the population:40

Entry number 5

Enter the name of the country:e

Enter the population:50

Entry number 6

Enter the name of the country:f

Enter the population:60

Entry number 7

Enter the name of the country:g

Enter the population:70

Entry number 8

Enter the name of the country:h

Enter the population:80

Entry number 9

Enter the name of the country:i

Enter the population:90

Entry number 10

Enter the name of the country:j

Enter the population:100

**Code:**

import math

import matplotlib.pyplot as plt

x1=[x for x in range(11)]

y1=[math.e-(x/10)\*math.sin(math.pi\*x) for x in range(11)]

plt.plot(x1,y1,color='red',label='f(x)')

x2=[x for x in range(11)]

y2=[math.e\*x-(x/3) for x in range(11)]

plt.plot(x2,y2,color='green',label='g(x)')

plt.xlabel('Values of x')

plt.ylabel('Value of function')

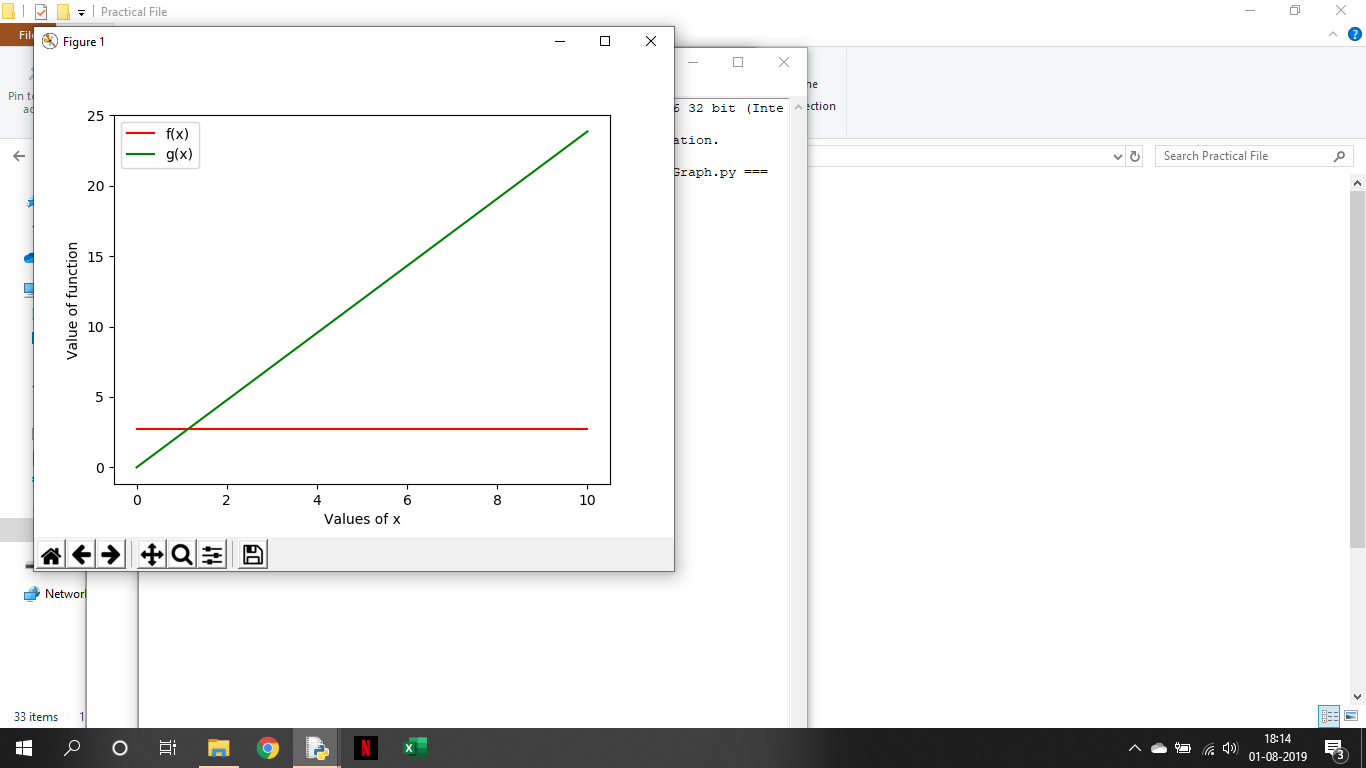
plt.legend()

scale\_x=1

scale\_y=2

plt.show()

**Output:**



**Code:**

def sorting(l1,l2):

l=l1+l2

return sorted(l)

def commonsum(l1,l2):

sum=0

for i in l1:

for x in l2:

if i==x:

sum+=i

return sum

def circular(l1,l2):

l3=l1\*2

for x in range(len(l1)):

z=0

for y in range(x, x + len(l1)):

if l2[z]==l3[y]:

z+=1

else:

break

if z == len(l1):

return True

return False

cont='y'

while True:

option=(int(input('\n\t\t1)Sort and merge\n\t\t2)Sum of common numbers\n\t\t3)Circular Identity\n\n\t\tEnter the option number:')))

if option==1:

print(sorting(list(map(int,input('List number 1\nEnter a set of numbers(separated by spaces):').split(sep=' '))),list(map(int,input('List number 2\nEnter a set of numbers(separated by spaces):').split(sep=' ')))))

elif option==2:

print(commonsum(list(map(int,input('List number 1\nEnter a set of numbers(separated by spaces):').split(sep=' '))),list(map(int,input('List number 2\nEnter a set of numbers(separated by spaces):').split(sep=' ')))))

elif option==3:

print(circular(list(map(int,input('List number 1\nEnter a set of numbers(separated by spaces):').split(sep=' '))),list(map(int,input('List number 2\nEnter a set of numbers(separated by spaces):').split(sep=' ')))))

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

break

**Output:**

1)Sort and merge

2)Sum of common numbers

3)Circular Identity

Enter the option number:

1. List number 1

Enter a set of numbers(separated by spaces):5 4 3 2 1

List number 2

Enter a set of numbers(separated by spaces):10 9 8 7 6

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

1. List number 1

Enter a set of numbers(separated by spaces):1 1 2 2 3 3

List number 2

Enter a set of numbers(separated by spaces):1 1 2 2 4 4

12

1. List number 1

Enter a set of numbers(separated by spaces):10 10 0 0 10

List number 2

Enter a set of numbers(separated by spaces):10 10 10 0 0

True

**Code:**

from tkinter import \*

from tkinter import messagebox

tk=Tk()

tk.title('Simple Interest')

tk.geometry('300x130')

p,r,t=0,0,0

def calculate():

global p

global r

global t

p,r,t=int(e1.get()),int(e2.get()),int(e3.get())

amt=p+(p\*r\*t)/100

messagebox.showinfo('Amount','Total loan amount=%s'%amt)

Label(tk,text='Principle Amount:').grid(row=1)

Label(tk,text='Rate of Interest:').grid(row=2)

Label(tk,text='Time(in years):').grid(row=3)

e1=Entry(tk)

e2=Entry(tk)

e3=Entry(tk)

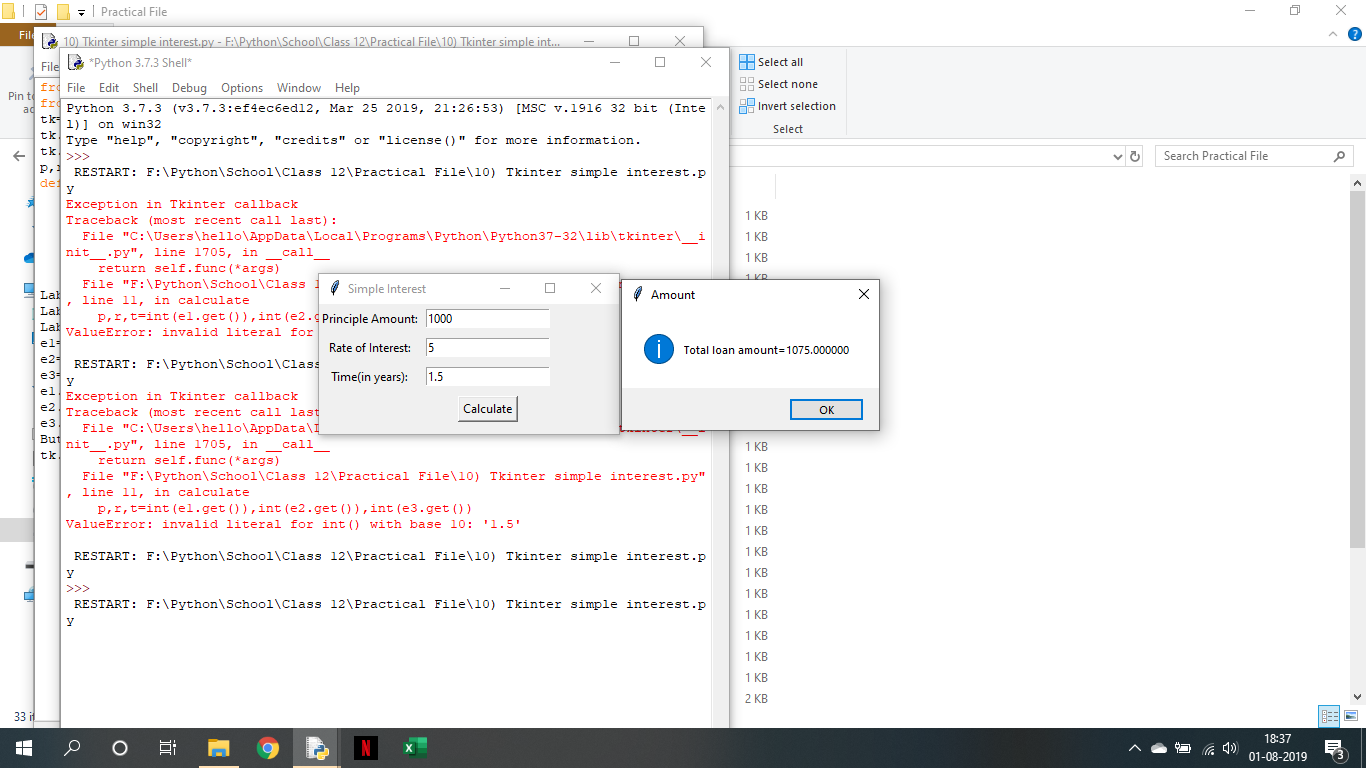
e1.grid(row=1,column=1,padx=5,pady=5)

e2.grid(row=2,column=1,padx=5,pady=5)

e3.grid(row=3,column=1,padx=5,pady=5)

Button(tk,text='Calculate',command=calculate).grid(row=4,column=1,pady=5)

tk.mainloop()

**Output:**

**Code:**

import timeit

print("Bubble Sort")

res=timeit.timeit('''

array=[2,4,3,2,1,6,3,8,6,4,6,9,0,1,2,4,5,6]

n=len(array)

for c in range(n):

for d in range(n-c-1):

if array[d]<array[d+1]:

swap=array[d]

array[d]=array[d+1]

array[d+1]=swap

print(array)

''', number=1)

print("Execution time:",res)

print("\nBinary Search")

res1=timeit.timeit('''

l=[2,3,4,5,7,8,9,11,12,15]

num=12

lb=0

ub=len(l)-1

while lb<=ub:

mid=(lb+ub)//2

if num>l[mid]:

lb=mid+1

elif num<l[mid]:

ub=mid-1

elif num==l[mid]:

print(num,"found at",mid)

break

else:

print("Seach not found")

''',number=1)

print("Execution time:",res1)

print("\nLinear Search")

res2=timeit.timeit('''

l=[2,4,5,3,6,4,5,23,78,89]

num=23

for i in l:

if i==num:

print(num,'found at',l.index(i))

break

''',number=1)

print("Execution time:",res2)

**Output:**

Bubble Sort

[9, 8, 6, 6, 6, 6, 5, 4, 4, 4, 3, 3, 2, 2, 2, 1, 1, 0]

Execution time: 7.61000000011336e-05

Binary Search

12 found at 8

Execution time: 4.8199999923781434e-05

Linear Search

23 found at 7

Execution time: 5.3200000024844485e-05

**Code:**

f=open('12) text handling 1.txt','r+') #Q12) (a)

x=f.read()

print(x)

f.close()

f=open('12) text handling 1.txt','a') #Q12) (b)

a=input('Enter the line you want to insert:')

f.write(a+'\n')

f=open('12) text handling 1.txt','r')

l=f.readlines()

print(l[-1])

f.close()

f=open('12) text handling 1.txt','r') #Q12) (c)

l=f.readlines()

print(l[-1])

f.close()

f=open('12) text handling 1.txt','r') #Q12) (d)

f.seek(10)

print(f.readline())

f.close()

f=open('12) text handling 1.txt','r') #Q12) (e)

a=int(input('Enter line number to be displayed:'))-1

print(f.readlines()[a])

f.close()

f=open('12) text handling 1.txt','r') #Q12) (f)

a=f.read()

l=a.split()

d={}

for x in l:

d[x[0]]=0

for x in l:

d[x[0]]+=1

for i in d:

print('Words beginning with',i,':',d[i])

f.close()

**Output:**

Neither apple nor pine are in pineapple. Boxing rings are square.

Writers write, but fingers don’t fing. Overlook and oversee are opposites.

A house can burn up as it burns down. An alarm goes off by going on.

Enter the line you want to insert:Hello people

A house can burn up as it burns down. An alarm goes off by going on.

Hello people

A house can burn up as it burns down. An alarm goes off by going on.

ple nor pine are in pineapple. Boxing rings are square.

Enter line number to be displayed:1

Neither apple nor pine are in pineapple. Boxing rings are square.

Words beginning with N : 1

Words beginning with a : 7

Words beginning with n : 1

Words beginning with p : 3

Words beginning with i : 2

Words beginning with B : 1

Words beginning with r : 1

Words beginning with s : 1

Words beginning with W : 1

Words beginning with w : 1

Words beginning with b : 4

Words beginning with f : 2

Words beginning with d : 2

Words beginning with O : 1

Words beginning with o : 4

Words beginning with A : 2

Words beginning with h : 1

Words beginning with c : 1

Words beginning with u : 1

Words beginning with g : 2

**Code:**

f=open('file1.txt','r')

a=f.read()

l=a.split()

vowel='aeiouAEIOU'

for x in l:

if x[0] not in vowel:

f1=open('file2.txt','a')

f1.write(x+' ')

print(f.read())

f.close()

f1.close()

**Output:**

Carry When Rains

**Code:**

f=open('14) Student.txt','r') #Q14) (a)

l=f.readlines()

l1=[]

for x in l:

l1.append(tuple(x,))

print(l1)

f=open('14) Student.txt','r') #Q14) (b)

a=f.read()

l=a.split()

cs=0

mm=0

bi=0

for x in l:

if x=='CSEE':

cs+=1

elif x=='Biology':

bi+=1

elif x=='MME':

mm+=1

print('CSEE:',cs,'\nMME:',mm,'\nBiology:',bi)

**Output:**

[('RajatSen 12345 1 CSEE\n'), ('JagatNarain 13467 3 CSEE\n'), ('AnuSharma 11756 2 Biology\n'), ('SumitaTrikha 23451 4 Biology\n'), ('SumderKumra 11234 3 MME\n'), ('KantiBhushan 23211 3 CSEE')]

CSEE: 3

MME: 1

Biology: 2

**Code:**

f=open('myfile.txt','r')

a=f.read()

l=a.split()

d={}

for x in l:

d[x]=0

for x in l:

d[x]+=1

def maxword():

global d

d1={}

w=d.values()

b=d.keys()

for x in range(len(d)):

d1[w[x]]=b[x]

return tuple(d1[max(w)])

print('Total number of words:',sum(d.values()))

print('Number of different words:',len(d))

print('The most common words:',maxword())

**Output:**

Total number of words:5

Number of different words:3

The most common words: (‘at’)

**Code:**

key={'a':'n', 'b':'o', 'c':'p', 'd':'q', 'e':'r', 'f':'s', 'g':'t', 'h':'u', 'i':'v', 'j':'w', 'k':'x', 'l':'y', 'm':'z', 'n':'a', 'o':'b', 'p':'c', 'q':'d', 'r':'e', 's':'f', 't':'g', 'u':'h', 'v':'i', 'w':'j', 'x':'k', 'y':'l', 'z':'m', 'A':'N', 'B':'O', 'C':'P', 'D':'Q', 'E':'R', 'F':'S', 'G':'T', 'H':'U', 'I':'V', 'J':'W', 'K':'X', 'L':'Y', 'M':'Z', 'N':'A', 'O':'B', 'P':'C', 'Q':'D', 'R':'E', 'S':'F', 'T':'G', 'U':'H', 'V':'I', 'W':'J', 'X':'K', 'Y':'L', 'Z':'M', ' ':' ','\n':'\n'}

def change():

l1=list(key.keys())

l2=list(key.values())

d={}

for x in range(len(l1)):

d[l2[x]]=l1[x]

return d

key1=change()

def cipher(a):

b=list(a)

d={}

for x in range(len(b)):

d[b[x]]=key[b[x]]

return d

def decipher(a):

b=list(a)

d={}

for x in range(len(b)):

d[b[x]]=key1[b[x]]

return d

cont='y'

while cont!='n':

option=(int(input('\n\t\t1)Cipher\n\t\t2)Decipher\n\n\t\tEnter the option number:')))

if option==1:

a=input('Enter the file name with extenstion:')

f=open(a,'r')

b=f.read()

f1=open('ciphered.txt','a')

c=list(cipher(b).values())

f1.write(str(''.join(c)))

elif option==2:

a=input('Enter the file name with extenstion:')

f=open(a,'r')

b=f.read()

f1=open('deciphered.txt','a')

c=list(decipher(b).values())

f1.write(str(''.join(c)))

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

cont='n'

f1.close()

**Output:**

A file is created by the name of ciphered.txt with the text ciphered and in deciphered a file is created with the name deciphered.txt

**Code:**

import numpy as np

print('5x5 Array')

x=np.random.random((5,5))

print('Array:')

print(x)

print('Minimum Value:',x.min())

print('Maximum Value:',x.max())

print('\nFrequent number')

x=np.random.randint(0, 10, 40)

print('Array:')

print(x)

print('Most frequent value in the above array:')

print(np.bincount(x).argmax())

print('\nCloset value')

x=np.arange(100)

print('Array:')

print(x)

a=np.random.uniform(0,100)

print('Value to compare:')

print(a)

index=(np.abs(x-a)).argmin()

print(x[index])

**Output:**

5x5 Array

Array:

[[0.59253884 0.74571727 0.18168803 0.74261942 0.98161418]

[0.12184972 0.31763405 0.17727049 0.67495719 0.63089555]

[0.27095215 0.89358776 0.36677565 0.38574441 0.65678865]

[0.3099814 0.12436212 0.30361121 0.94885937 0.17104231]

[0.51367618 0.27128165 0.88279344 0.76425053 0.96816248]]

Minimum Value: 0.12184971598592198

Maximum Value: 0.9816141778387583

Frequent number

Array:

[3 9 5 0 5 9 9 0 8 9 1 5 5 3 2 8 7 8 2 4 1 5 7 3 7 3 2 4 7 9 8 1 2 9 1 0 7

4 6 2]

Most frequent value in the above array:

9

Closet value

Array:

[ 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23

24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47

48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71

72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95

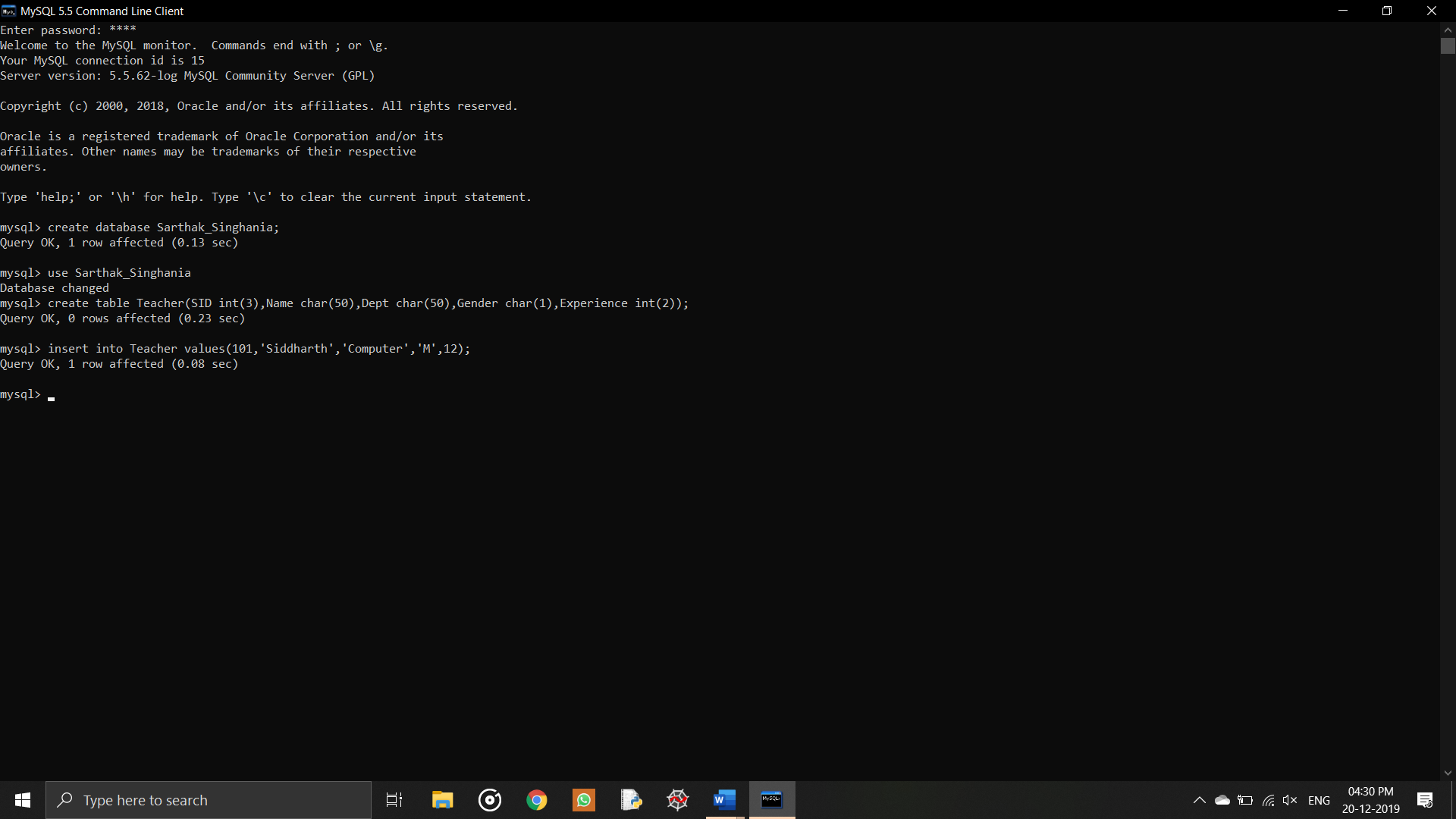
96 97 98 99]

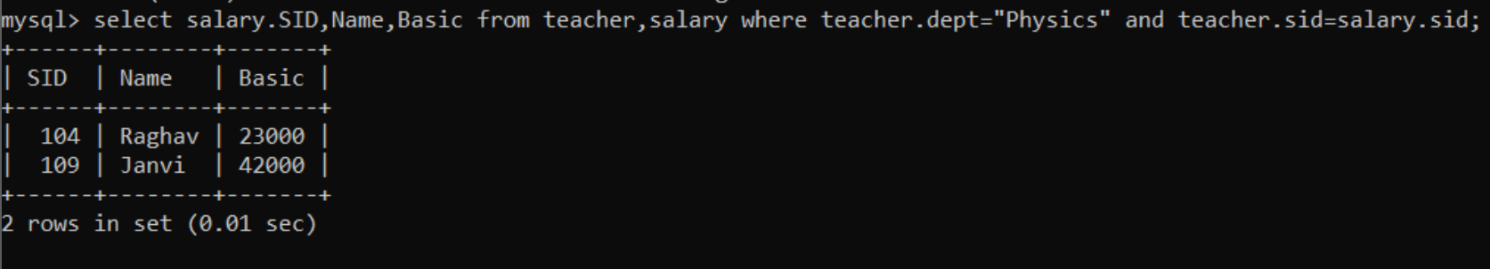
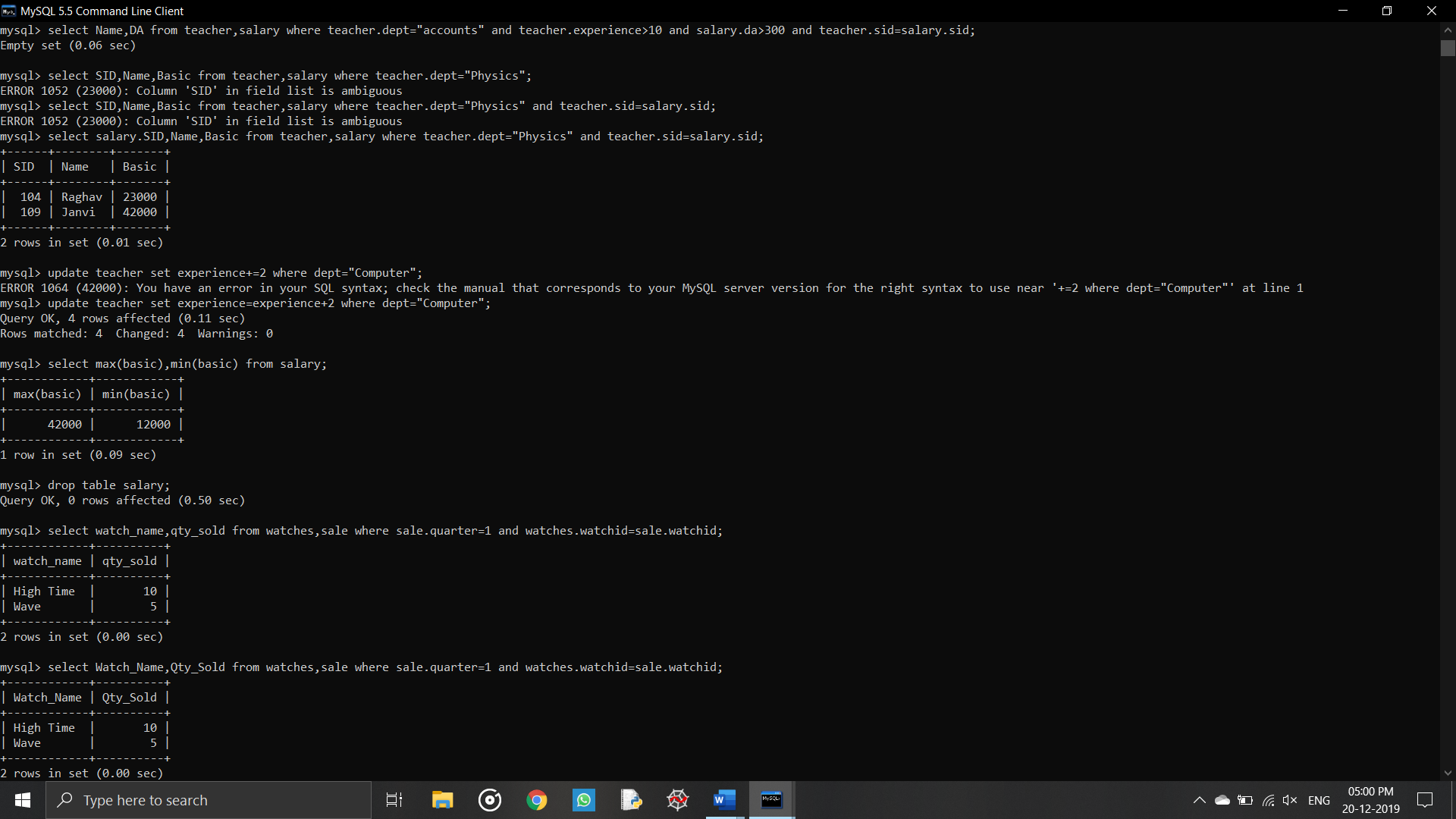
Value to compare:

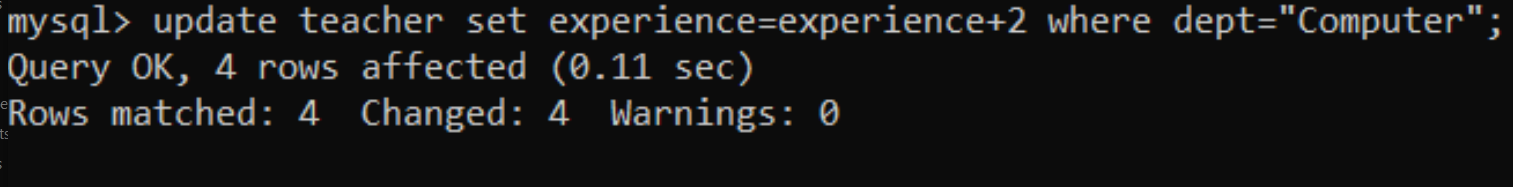
36.66172000326689

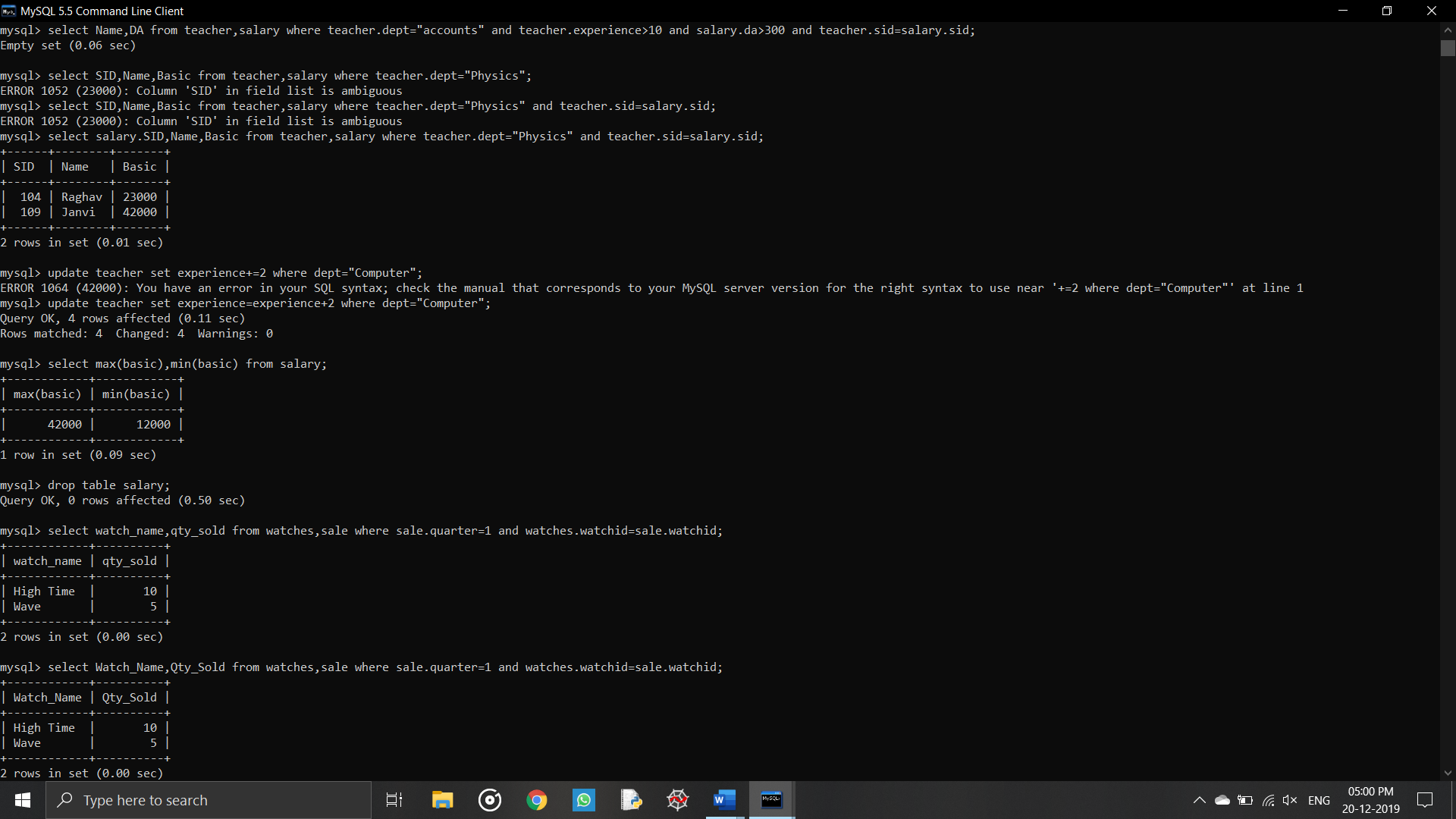
37

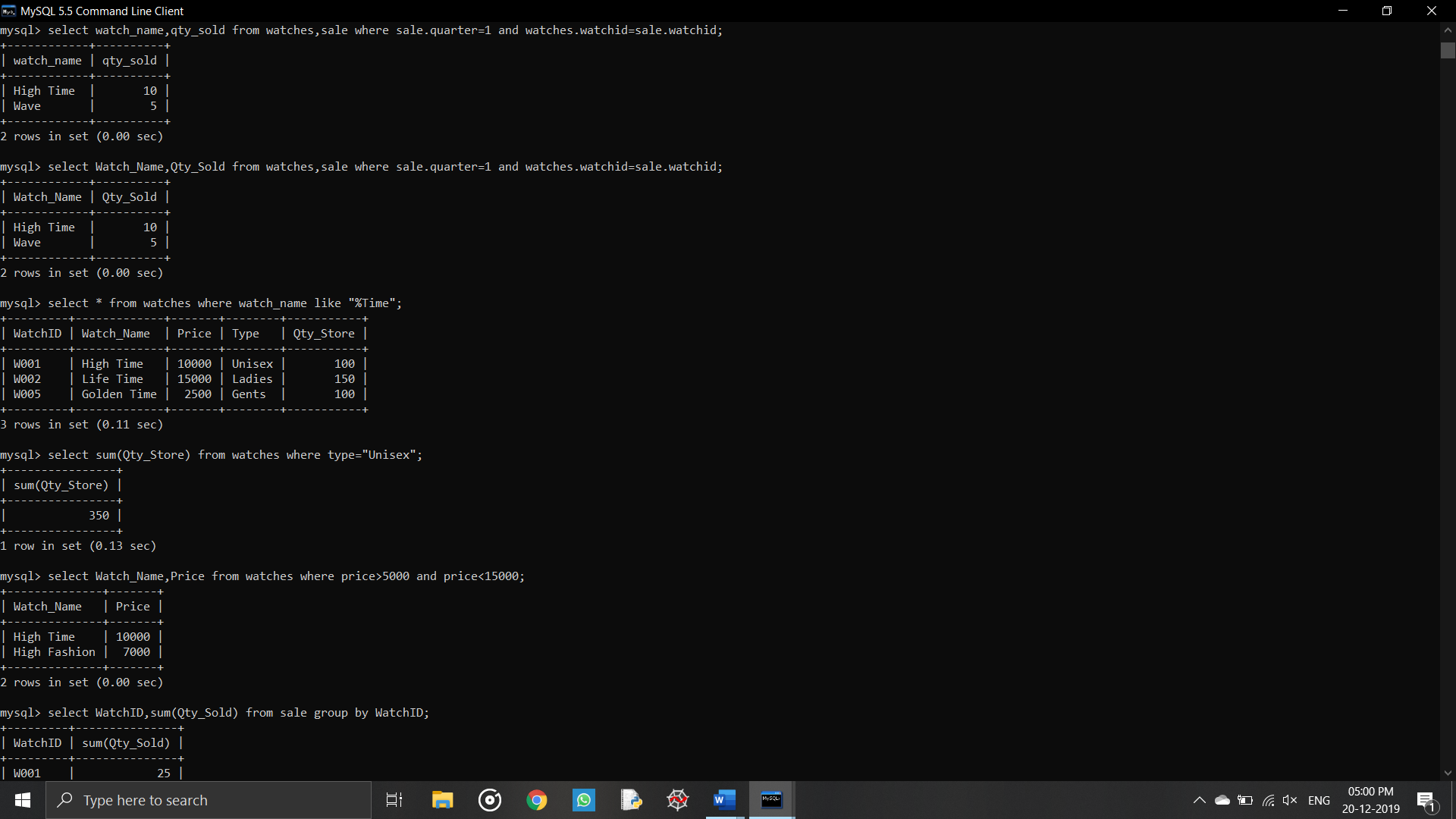
**MySQL**

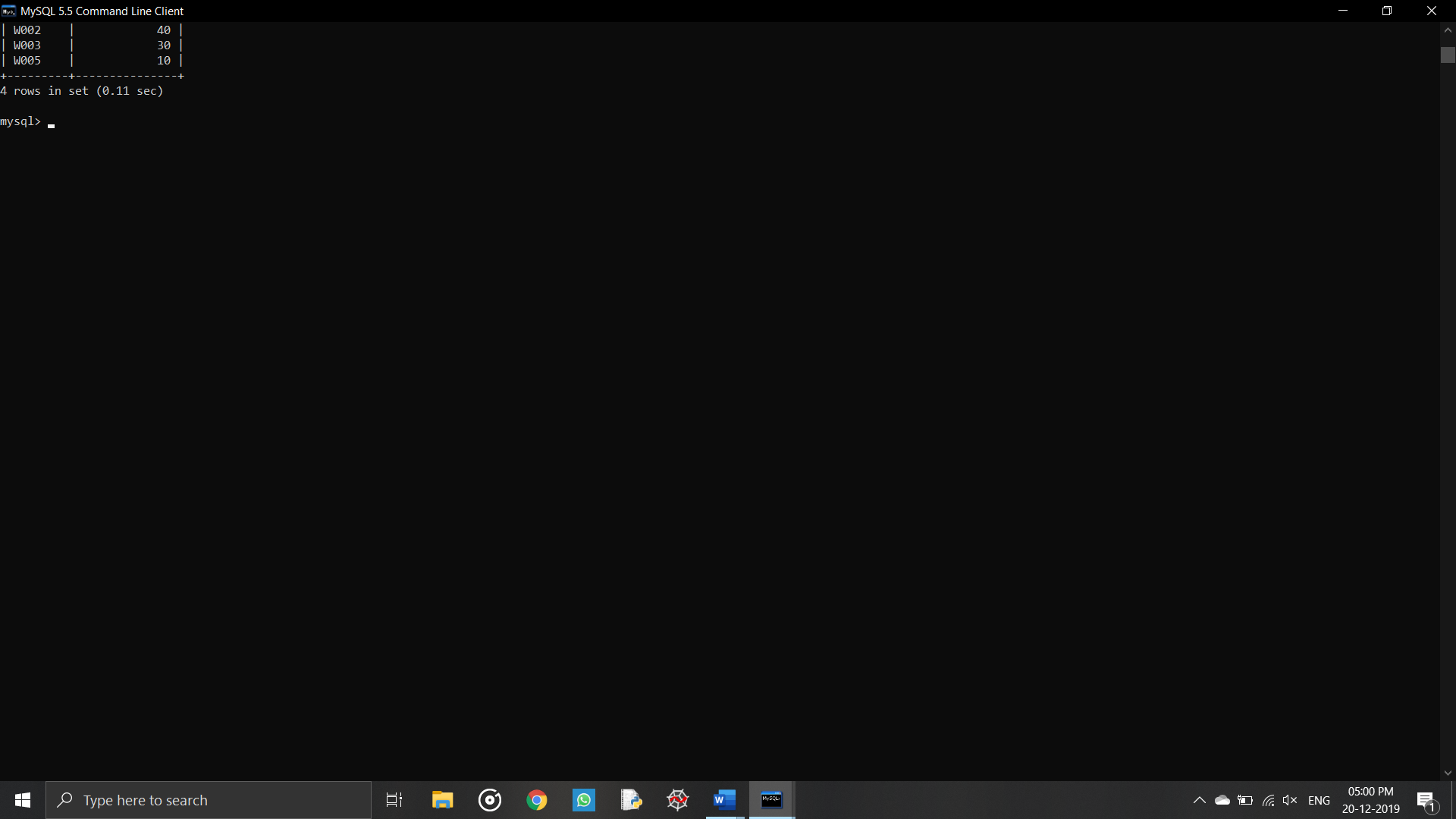
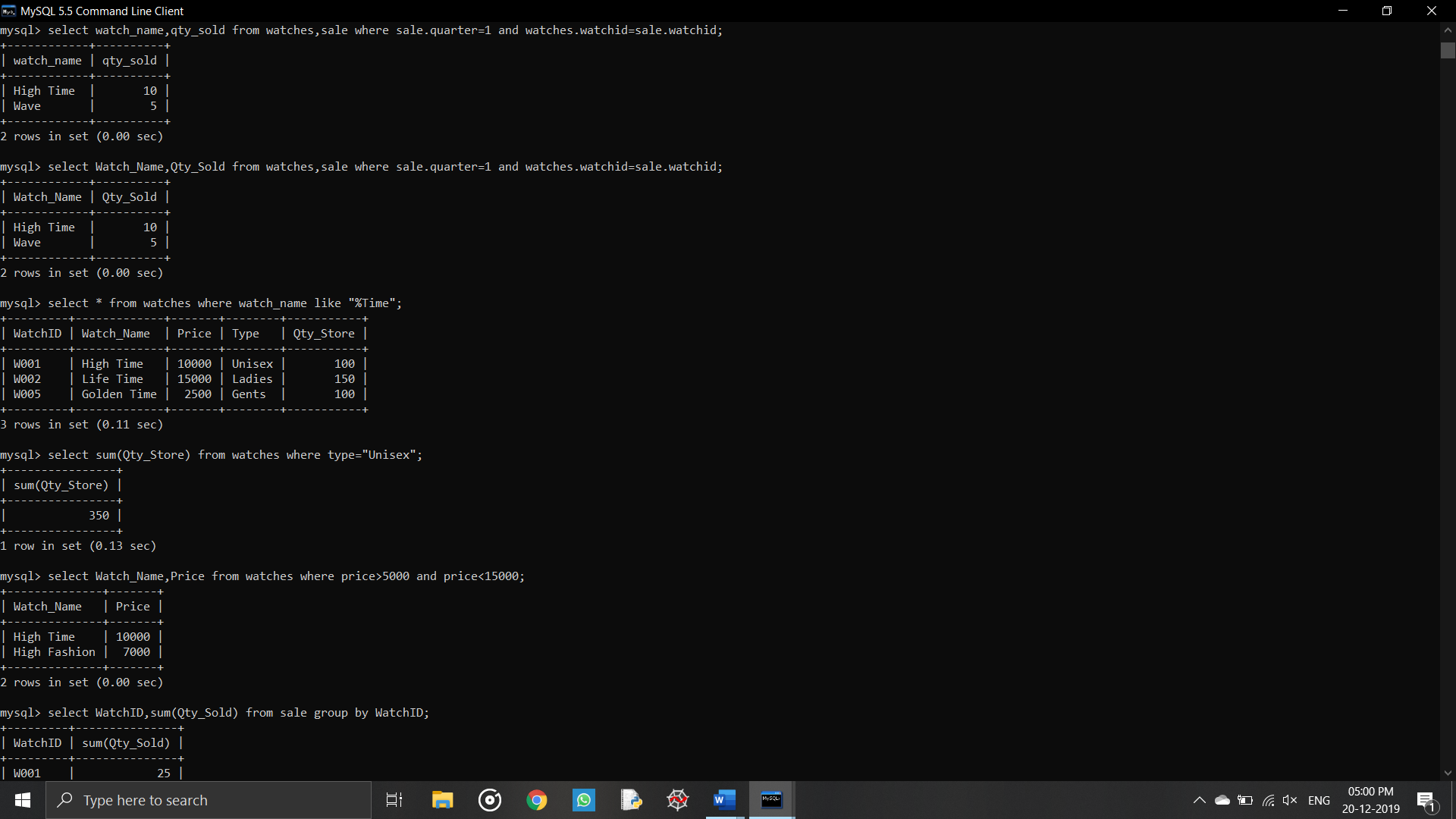


****Q1:





Q2:



**Code:**

import mysql.connector

mysql=mysql.connector.connect(host='localhost',user='root',password='root')

cur=mysql.cursor()

try:

cur.execute('create database mydb')

except:

pass

mysql=mysql.connector.connect(host='localhost',user='root',password='root',database='mydb')

cur=mysql.cursor()

try:

cur.execute('create table Item(Item\_code varchar(20),Item\_name varchar(20),Price float(11))')

except:

pass

def insert():

a=input('Enter Item Code:')

b=input('Enter Item Name:')

c=input('Enter Price:')

com='insert into item values(%s,%s,%s)'

val=(a,b,c)

cur.execute(com,val)

print('Record Inserted')

def view():

cur.execute('select \* from item')

for i in cur:

print(i)

def search():

a=input('Enter Item Code:')

com='select \* from item where Item\_code="%s"'%a

cur.execute(com,a)

for i in cur:

print(i)

while True:

option=(int(input('\n1)Insert Record\n2)View all records\n3)Search\nEnter the option number:')))

if option==1:

insert()

elif option==2:

view()

elif option==3:

search()

cont=input('Do you want to continue(y/n):')

if cont=='n':

break

**Output:**

1)Insert Record

2)View all records

3)Search

Enter the option number:1

Enter Item Code:S001

Enter Item Name:G-Shock

Enter Price:6000

Record Inserted

Do you want to continue(y/n):y

1)Insert Record

2)View all records

3)Search

Enter the option number:2

('S001', 'G-Shock', 6000.0)

Do you want to continue(y/n):y

1)Insert Record

2)View all records

3)Search

Enter the option number:3

Enter Item Code:S001

('S001', 'G-Shock', 6000.0)

**Code:**

import mysql.connector

mysql=mysql.connector.connect(host='localhost',user='root',password='root',database='mydb')

cur=mysql.cursor()

try:

cur.execute('create table Student(RollNo int primary key,Name varchar(50),Class int,DOB date,Gender char(2)')

except:

pass

def insert():

a=input('Enter Roll No:')

b=input('Enter Name:')

c=input('Enter Class:')

d=input('Enter Date(YY/MM/DD):')

e=input('Enter Gender(M/F):')

com='insert into Student values(%s,%s,%s,%s,%s)'

val=(a,b,c,d,e)

cur.execute(com,val)

print('Record Inserted')

def update():

a=input('Enter Roll No to update details:')

b=input('Enter Name:')

c=input('Enter Class:')

d=input('Enter Date(YY/MM/DD):')

e=input('Enter Gender(M/F):')

com='update Student set Name=%s,Class=%s,DOB=%s,Gender=%s where Roll\_No=%s'

val=(b,c,d,e,a)

cur.execute(com,val)

print('Updated')

while True:

option=(int(input('\n1)Insert Record\n2)Update record\nEnter the option number:')))

if option==1:

insert()

elif option==2:

update()

cont=input('Do you want to continue(y/n):')

if cont=='n':

break

**Output:**

1)Insert Record

2)Update record

Enter the option number:1

Enter Roll No:1

Enter Name:Sarthak

Enter Class:12

Enter Date(YY/MM/DD):02/06/28

Enter Gender(M/F):M

Record Inserted

Do you want to continue(y/n):y

1)Insert Record

2)Update record

Enter the option number:2

Enter Roll No to update details:1

Enter Name:Sarthak Singhania

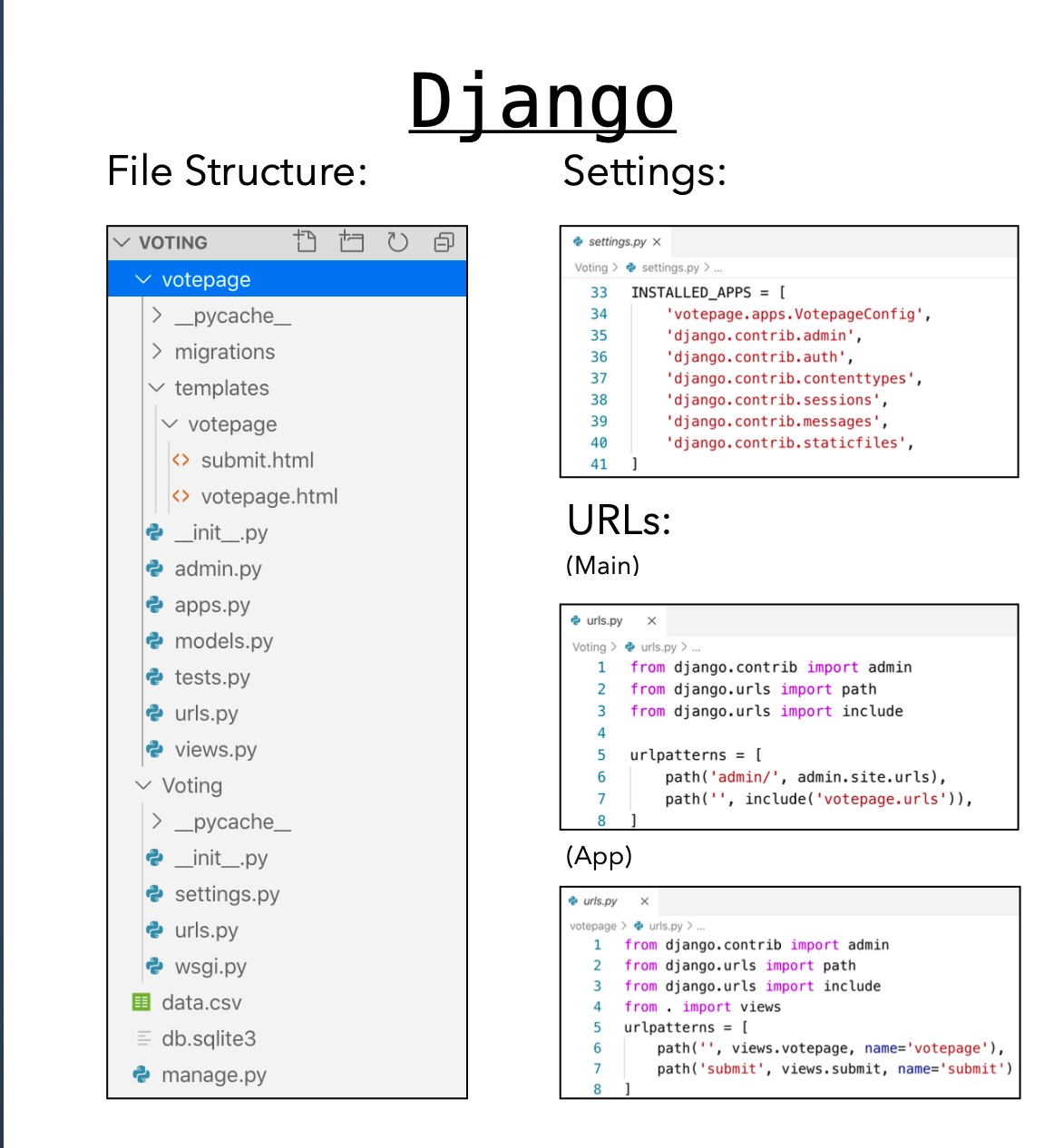
Enter Class:12

Enter Date(YY/MM/DD):02/06/28

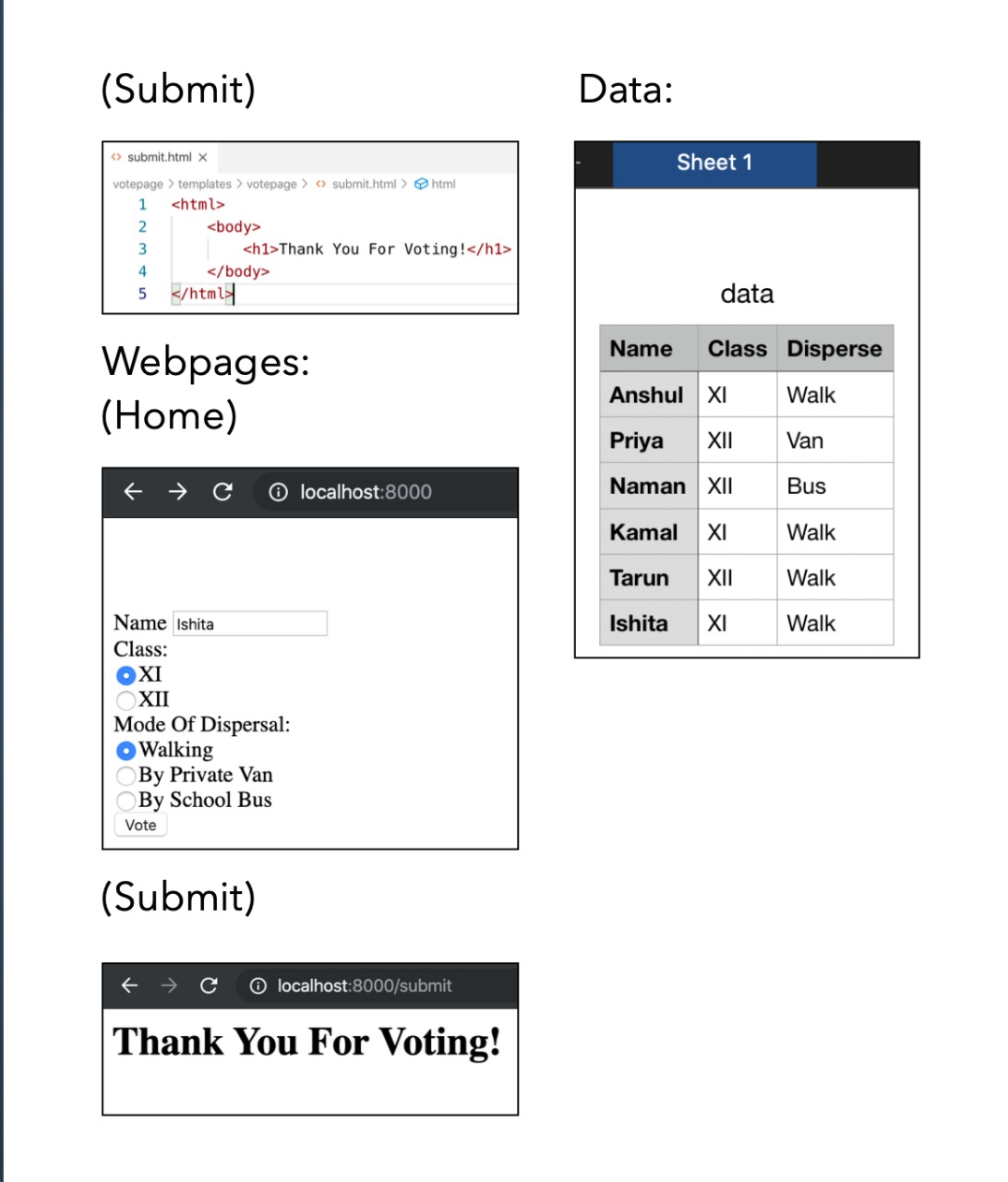
Enter Gender(M/F):M

Updated

**Django**

****

****

****