CBSE

Rahul R , Akash R , Basil J , Sundaram M

Physics Portal

2017-2018



Computer Project

**Index:**

|  |  |
| --- | --- |
| **Contents** | **Page no** |
| **Certificate Page** | **3** |
| **Acknowledgement** | **4** |
| **Introduction** | **5** |
| **Data Dictionary** | **6-7** |
| **System Requirement** | **8** |
| **Data flow diagram** | **9** |
| **Source Code** | **10-56** |
| **Output** | **57-67** |
| **Future Enhancement** | **68** |
| **Conclusion** | **68** |
| **Bibliography** | **68** |
|  |  |
|  |

**CERTIFICATE**

This is to certify that \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ , a student of class XII has

Successfully completed the project

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ under the guidance

Ms.Sonia Martin (subject Teacher) during the year 2016-2017

in computer practical examination conducted

by CBSE .

**Signature of the computer teacher**

**Signature of the examiner**

**Name of the examiner**

**Name of the Computer Teacher**

* **Sonia Martin**

**ACKNOWLEDGEMENT**

I would like to thank everyone who have kept me motivated and I'm grateful to them for helping me do this project.

I would like to thank the school vice-principal,Mrs.Shyla Purushottam for being a constant source of help and support and the Associate Vice Principal,Mrs.Sumedha Joshi for being equally helpful and supportive.

I would also like to thank my Computer Sciences teacher,Mrs. Sonia Martin, for her painstaking effort and spontaneous teaching which has also helped me do this project.

I would like to thank my parents as well,for their constant motivation and help in whatever ways possible,for without them,my work and effort would have no meaning.

I would also like to thank my project peers,Akash.R,Basil Jacob and Rahul Raman who have been putting equal effort and hardwork to complete this project.

I would like to thank my classmates and friends for helping me in times of need and making my work much more easier and enjoyable.

All the above people I have mentioned have helped me,directly or indirectly,and I'm indebted to them for what they have done.

**पावती**

मैं हर कोई है जो मुझे रखा है और प्रेरित मुझे मदद करने के लिए मुझे इस परियोजना के लिए आभारी हूं शुक्रिया अदा करना चाहूंगा ।

मैं स्कूल के वाइस-प्रिंसिपल, श्रीमती Shyla पुरुषोत्तम को सहायता और समर्थन का निरंतर स्रोत बनने और सहयोगी वाइस प्रिंसिपल, श्रीमती सुमेधा जोशी को समान रूप से सहायक और सहायक बनने के लिए धन्यवाद देना चाहूंगा ।

मैं भी अपने कंप्यूटर विज्ञान शिक्षक, श्रीमती सोनिया मार्टिन, उसे श्रमसाध्य प्रयास और सहज शिक्षण जो भी मदद की है मुझे इस परियोजना के लिए शुक्रिया अदा करना चाहूंगा ।

मैं अपने माता पिता के रूप में अच्छी तरह से शुक्रिया अदा करना चाहूंगा, उनके निरंतर प्रेरणा के लिए और जो भी संभव तरीके में मदद के लिए, उनके बिना, मेरे काम और प्रयास कोई मतलब नहीं होगा ।

मैं अपने प्रोजेक्ट साथियों, आकाश. आर, तुलसी जैकब और राहुल रमन को भी धन्यवाद देना चाहूंगा जो इस प्रोजेक्ट को पूरा करने के लिए बराबर प्रयास और मेहनत लगा रहे हैं ।

मैं जरूरत के समय में मेरी मदद करने और मेरे काम को और अधिक आसान और मनोरंजक बनाने के लिए अपने सहपाठियों और दोस्तों को धन्यवाद देना चाहूंगा ।

उपर्युक्त सभी लोगों ने मुझे मदद की है, प्रत्यक्ष या परोक्ष रूप से, और मैं उंहें क्या किया है के लिए उंहें ऋणी हूं ।

**INTRODUCTION**

To reduce the tension during exams, To reduce the calculations involved , and above all to develop love towards physics, we have created a platform for students as well as teachers to refer to and use formulas and topics for the Class XI and XII of the CBSE board.

This physics portal is made of a menu-driven program which gives the user the flexibility to choose between many options such as Previous year Question Papers, Weightage of Chapters , A Quick-Quiz to show the student’s level of preparation and enabling the user to do faster calculations in common Questions asked in exams. We have also included extra facts and points to make it more interesting. All the important topics have been covered to provide a recap of concepts before exams.

The user friendly interface created in this program was achieved by implementing the Tkinter GUI of Python. The program has been broken into multiple modules for the user to be able to read and understand the working of the program clearly. To allow only specific users to access our program we have created a Login page and a Sign up page allowing users to be a part of the program.

We have used multiple concepts such as pickling, Classes and Objects, Data File Handling, Modular Programming etc. The program was created, tested and debugged in Python version 2.7.13. The program requires a minimum of Dual-Core CPU,2gb RAM and 500kb available space.

Since this project is on physics concepts we have named our project as **PHYSICS PORTAL**

If you can spare time, you can also help us improvise our program by providing comments and feedback included in the program.

**Data Dictionary:**

|  |  |  |
| --- | --- | --- |
| **DATA TYPE** | **DATA MEMBERS** | **DESCRIPTION** |
| Instance variable | l,label1,label2.. | Used for creating Labels in Tkinter. |
| Instance variable | b1,b2,b3,b4,  button1,button2.button3 | Used for creating Buttons in Tkinter. |
| Instance variable | Entry1,entry2 | Used for creating a textfield inTkinter. |
| Instance variable | Root | Used for initializing Tkinter and creating an ordinary window with title bar. |
| File variable | fw,fr | Used for reading and writing into files. |
| List | vect1,vet2 | Variables used for storing the -i,j,k components of a vector in the form of a list. |
| Float | m1,m2,a,b | Used for accepting the mass of the blocks in float datatype. |
| int | ch,choice1 | Used for creating choice in menu driven program. |
| Float | Ystress,ystrain,bstress,bstrain,  Shstrain,shstress | Float variables used to store the Longitudinal,Bulk and  Shearing strain. |

**PRE -DEFINED MODULES IMPORTED**

1. MATH - FOR USING THE SIN,COS AND PI

FUNCTIONS.

1. RANDOM - FOR USING RANDRANGE AND

RANDINT FUNCTONS.

3) TKINTER - FOR ENABLING THE GUI

OF PYTHON.

**USER-DEFINED MODULES IMPORTED**

* Centre
* Moduli
* OTHER
* STRAIN
* STRESS
* Angular
* Horizontal
* Vector
* SECONDLAW

**System Requirements:**

**Hardware:**

* Modern Operating System:
  + Windows 7 or 10
  + Mac OS X 10.10 or higher, 64-bit
  + Linux: RHEL 6/7, 64-bit (almost all libraries also work in Ubuntu)
* x86 CPU (Intel / AMD architecture)
* 4 GB RAM
* 5 GB free disk space

The above requirements are required for the best results. 😊

**Software:**

* Python 2.7.12
* Tkinter Module (If not available,in the python program files then download from <https://www.python.org/>

**Data Flow Diagram:**

**Quiz:**

11 & 12

**Imp. Topics**

**Physics calc.**

Projectile

Newton FBD,SL

Vector

Stress , Strain

**Previous Q.P**

* 2014
* 2015
* 2016

**Facts**

**Feedback**

**User**

**Account-Management**

**Main Program**

**Sign-Up**

**Login**

**Start:**

from Tkinter import \*

def SIGNUP():

import Signup

def LOGIN():

import Login

root = Tk()

button1 = Button(root , text = "Sign\_up" , command = SIGNUP)

button2 = Button(root , text = "Login" , command = LOGIN)

button1.pack()

button2.pack()

root.mainloop()

**Main:**

#Creating a function of Horizontal projectile motion.

def Horizontal():

print "You have chosen horizontal projection"

print "What would you like to find:"

print """ 1. Time of flight

2. Maximum horizontal range

3. Velocity at any point """

ch=int(raw\_input("Enter your choice:"))

if ch==1:

height=float(raw\_input("Enter the horizontal height(H):"))

print " Now by using the formula..... T=[2\*(H)/g]\*\*0.5" # \*\* indicates to the power, and 'g' indicates force of gravity (9.8 m/s\*\*2), T is time of flight.

T=((2\*height)/9.8)\*\*0.5

print " The Time of flight is:",T

if ch==2:

initial=float(raw\_input("Enter the initial velocity of the projectile(U):"))

height=float(raw\_input("Enter the horizontal height(H):"))

print " Now by using the formula......R=u\*[2\*(H)/g]\*\*0.5" # \*\* indicates to the power, and 'g' indicates force of gravity (9.8 m/s\*\*2), R is the maximum horizontal range.

R= initial\*(2\*(height)/9.8)\*\*0.5

print " The Horizontal range is :",R

if ch==3:

initial=float(raw\_input("Enter the initial velocity of the projectile(U):"))

Time=float(raw\_input("Enter the time taken(T):"))

print " Now by using the formula....V=[U\*\*2 + (g\*T)\*\*2]\*\*0.5" # \*\* indicates to the power, and 'g' indicates force of gravity (9.8 m/s\*\*2), T is time of filght , U is the initial velocity.

V=(initial\*\*2 + (9.8\*Time)\*\*2)\*\*0.5 # V is the Velocity at that point

print "The velocity at that point is:",V

**Physics Calculator:**

import centre

import MODULI

import OTHER

import STRAIN

import STRESS

import Angular

import Horizontal

import SECONDLAW

import Vector

i=True

while i==True:

print "What would you like to do ,User:"

print """1) Centre of Mass

2) Free body diagram

3) Modulus

4) Projectile Motion

5) Secondary law of motion

6) Vector """

choice=int(raw\_input("What would you like to do ?:"))

if choice==1:

centre.centre()

elif choice==2:

import FBDS

elif choice==3:

print "What would you like to find ?"

print """1) Modulus

2) Strain

3) Stress

4) Solid prop """

option2=int(raw\_input("Enter your choice:"))

if option2==1:

MODULI.modulus()

elif option2==2:

STRAIN.strain()

elif option2==3:

STRESS.stress()

else:

OTHER.othersolidprop()

elif choice==4:

print "What would you like to find ?"

print """1) Angular projectile motion.

2) Linear projectile motion """

choice3=int(raw\_input("Enter your choice:"))

if choice3==1:

Angular.Angular()

else:

Horizontal.Horizontal()

elif choice==5:

SECONDLAW.SecondLawOfMotion()

elif choice==6:

Vector.vector()

want=raw\_input("Do you want to continue (Yes/no):")

if want=="no":

break

else:

print "Please Re-run the program."

**Sign-Up:**

from Tkinter import \*

def check():

fw = open(entry1.get()+".txt" , "w")

if entry2.get().isalnum() == False:

if entry3.get() == entry2.get():

fw.write(entry2.get())

print "Successfully created."

else:

print "Re\_password is not matching the password"

else:

print "The password should contain alphabets , digits and special characters."

root = Tk()

label1 = Label(root , text = "Username")

label2 = Label(root , text = "Password")

label3 = Label(root , text = "Re\_enter\_Password")

entry1 = Entry(root)

entry2 = Entry(root)

entry3 = Entry(root)

button1 = Button(root , text = "Create" , command = check)

label1.grid(row = 0 , column = 1)

label2.grid(row = 1 , column = 1)

label3.grid(row = 2 , column = 1)

entry1.grid(row = 0 , column = 2)

entry2.grid(row = 1 , column = 2)

entry3.grid(row = 2 , column = 2)

button1.grid(columnspan = 2)

root.mainloop()

**Login:**

from Tkinter import \*

def confirm():

try:

fr = open(entry1.get()+".txt" , "r")

except IOError:

print "There is no username existing."

else:

check = fr.read()

if check == entry2.get():

print "Successfully loged in."

import Main

else:

print "Enter valid password."

root = Tk()

label1 = Label(root , text = "Username")

label2 = Label(root , text = "Password")

entry1 = Entry(root)

entry2 = Entry(root)

button1 = Button(root , text = "Login" , command = confirm)

label1.grid(row = 0 , column = 1)

label2.grid(row = 1 , column = 1)

entry1.grid(row = 0 , column = 2)

entry2.grid(row = 1 , column = 2)

button1.grid(columnspan = 2)

root.mainloop()

**Horizontal:**

#Creating a function of Horizontal projectile motion.

def Horizontal():

print "You have chosen horizontal projection"

print "What would you like to find:"

print """ 1. Time of flight

2. Maximum horizontal range

3. Velocity at any point """

ch=int(raw\_input("Enter your choice:"))

if ch==1:

height=float(raw\_input("Enter the horizontal height(H):"))

print " Now by using the formula..... T=[2\*(H)/g]\*\*0.5" # \*\* indicates to the power, and 'g' indicates force of gravity (9.8 m/s\*\*2), T is time of flight.

T=((2\*height)/9.8)\*\*0.5

print " The Time of flight is:",T

if ch==2:

initial=float(raw\_input("Enter the initial velocity of the projectile(U):"))

height=float(raw\_input("Enter the horizontal height(H):"))

print " Now by using the formula......R=u\*[2\*(H)/g]\*\*0.5" # \*\* indicates to the power, and 'g' indicates force of gravity (9.8 m/s\*\*2), R is the maximum horizontal range.

R= initial\*(2\*(height)/9.8)\*\*0.5

print " The Horizontal range is :",R

if ch==3:

initial=float(raw\_input("Enter the initial velocity of the projectile(U):"))

Time=float(raw\_input("Enter the time taken(T):"))

print " Now by using the formula....V=[U\*\*2 + (g\*T)\*\*2]\*\*0.5" # \*\* indicates to the power, and 'g' indicates force of gravity (9.8 m/s\*\*2), T is time of filght , U is the initial velocity.

V=(initial\*\*2 + (9.8\*Time)\*\*2)\*\*0.5 # V is the Velocity at that point

print "The velocity at that point is:",V

**Angular:**

def Angular():

import math

print "You have chosen Angular projectile motion"

print "What would you like to find:"

print """ 1. Time of filght

2. Horizonatal range

3.Maximum height"""

ch=int(raw\_input("Enter your choice:"))

if ch==1:

initial=float(raw\_input("Enter the initial velocity(U):"))

angle=float(raw\_input("Enter the angle of projectile:"))

print "By using the formula... T=2\*U\*sin(angle)/g" # where T is the time of flight , 'g' is the force of gravity(9.8m/s\*\*2) , U is the initial velocity.

T=(2\*initial\*math.sin(angle))/9.8

print "Time of flight of the projectile is:",T

elif ch==2:

initial=float(raw\_input("Enter the initial velocity of the projectile(U):"))

angle=float(raw\_input("Enter the angle of the projectile:"))

print " By using the formula.... R=U\*cos(angle)\*2\*U\*sin(angle)/g" # where R is the horizontal range , U is the initial velocity , 'g' is the force of gratvity.

R=(initial\*math.cos(angle)\*2\*initial\*math.sin(angle))/9.8

print " The horizontal range is:",R

elif ch==3:

initial=float(raw\_input("Enter the initial velocity of the projectile(U):"))

angle=float(raw\_input("Enter the angle of the projectile:"))

print "By using the formula... H=U\*\*2\*sin(angle)\*\*2/2g" # where H is the horizontal range , U is the initial velocity of the projectile.

H=((initial\*\*2)\*(math.sin(angle)\*\*2))/19.6

print "The horizontal range is:",H

**Quiz:**

def quiz():

import random

for i in range(1,6):

r=(random.randint(1,10))

if r==1:

print "A stone is thrown horizontally with a speed of (2gh)\*\*0.5 from the top of wall of height h. It strikes the level ground through the foot of the wall at a distance x from the wall.What is the value of x?"

print """1)3g/2

2)2h

3)7h

4)4h"""

option1=int(raw\_input("Enter your option:"))

if option1==2:

print "Correct"

else:

print "Wrong"

print "Solution:"

print """The time taken to fall through height h=(2g/h)\*\*0.5

Now (2gh)\*\*0.5 = x/(2h/g)\*\*0.5) ; By solving the equation we get

x=2h"""

print " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ "

elif r==2:

print "The maximum range of projectile is 2/(3)\*\*0.5 times its actual range. What is the angle of projection for the actual range?"

print """1)75

2)60

3)30

4)45"""

option2=int(raw\_input("Enter your option:"))

if option2==3:

print "Correct"

else:

print "Wrong"

print "Solution:"

print """ v\*\*2/g = v\*\*2 \*sin2(angle)/g \* 2/(3)\*\*0.5

sin2(angle)=(3)\*\*0.5/2

angle=60"""

print "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"

elif r==3:

print "Find the depth of a lake for which the density of water will be 10% higher than the density at the surface. Compressiblity of water is 0.00005 per atmosphere. Given:average density of sea water=1100 kg m\*\*-3"

print """1)12 m

2)15 km

3)17 km

4)17 m"""

option3=int(raw\_input("Enter your choice:"))

if option3==3:

print "Correct"

else:

print "Wrong"

print "Solution:"

print """d=density , d=m/V d'=m/V-v

d/d'=m/V \* (V-v)/m

v/V = 1-d/d' = d'-d/d' = 10/110

B = 1/(5\*10\*\*-5 atm) = 1.01 \* 10\*\*5/d\*10\*\*-5 N m\*\*-2

P=dgh

B=hdg/v/V = 1.01\*10\*\*5/ 5\*10\*\*-5 \*1/11 \* 1/1100\*9.8

m=17 km"""

print "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"

elif r==4:

print "What mass must be suspended from a steel wire 2m long and 1mm diameter to stretch it by 1mm? Givem: Young's modulus= 2\*10\*\*12 dyne cm\*\*-2."

print """ 1)8.009 kg

2) 17 kg

3) 8 g

4) 17 g """

option4=int(raw\_input("Enter your choice:"))

if option4==1:

print "Correct"

else:

print "Wrong"

print "Solution:"

print """ Y = (F\*L)/(A\*delta(L))

M = 2\*10\*\*12 \* 22 \* 0.05 \*0.05\*0.1/7\*981\*200 = 8.009 kg"""

print "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"

elif r==5:

print " Can the rectangular component of a vector be greater than the vector itself ?"

option5=raw\_input("Enter yes/no:")

if option5=="yes":

print "Correct"

else:

print "Wrong"

print "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"

elif r==6:

print "If the given components of a given vector are perpendicular to each other , then they are called \_\_\_\_\_\_\_\_\_\_\_\_"

option6=raw\_input("Fill in the blank:")

if option6 in "rectangular componens":

print "Correct"

else:

print "False"

print "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"

elif r==7:

print "Two masses 8 kg and 12 kg are connected at the two ends of a frictionless pulley. Find the acceleration of the masses , and the tension in the string when the masses are released ?"

print """1) 7 ms\*\*-2 , 100 N

2)4 ms\*\*-2 , 87 N

3) 2 ms\*\*-2 , 96 N

4) 3 ms\*\*-2 , 79 N """

option7=int(raw\_input("Enter your choice:"))

if option7==3:

print "Correct"

else:

print "Wrong"

print "Solution:"

print """ a=(m1-m2)/(m1+m2) = 12-8/12+8 = 2 ms\*\*-2

T = 2m1m2/m1+m2 = 2(12)(8)/12+8 = 96 N"""

print "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"

elif r==8:

print " The angle through which the outer edge is raised above the inner edge is called \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"

option8=raw\_input("Fill in the blank:")

if option8 in "angle of banking":

print "Correct"

else:

print "Wrong"

print "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"

elif r==9:

print "70 kg man jumps to a height of a 0.8m . Find the impulse provided by ground to man ?"

print """1) 318.9 kg

2)283.32 kg

3)121.32 kg

4)277.2 kg"""

option9=int(raw\_input("Enter your choice:"))

if option9==4:

print "Correct"

else:

print "Wrong"

print "Solution:"

print """ Impulse provided = Change in momentum of the person

m(vf-vi)=m((2gh)\*\*0.5-0)

= 70\*3.96 = 277.2 kgms\*\*-1"""

print "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"

elif r==10:

print "The kinetic energy of a body is increased by 21% . What is the percentage increase in the linear momentum of the body ?"

print """1)10%

2)12%

3)17%

4)7%"""

option10=int(raw\_input("Enter your choice:"))

if option10==1:

print "Correct"

else:

print "Wrong"

print "Solution:"

print """ Ek2=121\*EKI/100

v2=11\*v1/10 or m2v2=11/10\*m\*v1

p2=11\*p1/10

p2-p1/p\*100=1/10\*100 = 10%"""

print "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"

**Quiz-12:**

from Tkinter import \*

def key1():

if var1.get() == 1:

print "correct"

else:

print "False"

def key2():

if var2.get() == 1:

print "correct"

else:

print "False"

def key3():

if var3.get() == 1:

print "correct"

else:

print "False"

root = Tk()

label1 = Label(root , text = " 1.When the seperation between two charges is increased , the electric potential energy of the charges..")

var1 = IntVar()

button1 = Checkbutton(root , text = "Increases", command = key1)

button1.place(relx =.1 , rely =.15 , anchor = "c" )

button2 = Checkbutton(root , text = "Decreases" , variable = var1 , command = key1)

button2.place(relx =.1 , rely =.20 , anchor = "c" )

button3 = Checkbutton(root , text = "Same", command = key1)

button3.place(relx =.09 , rely =.25 , anchor = "c")

button4 = Checkbutton(root , text = "May increase or decrease", command = key1)

button4.place(relx =.1255 , rely =.30 , anchor = "c")

label1.place(relx =.2 , rely =.1 , anchor = "c")

label2 = Label(root , text = " 2.An electric dipole is placed in a uniform electric field. The net electric force on the dipole is ....")

var2 = IntVar()

button5 = Checkbutton(root , text = "always zero", variable = var2, command = key2)

button5.place(relx =.1 , rely =.40 , anchor = "c" )

button6 = Checkbutton(root , text = "Depends on orientation of dipole", command = key2)

button6.place(relx =.14, rely =.45 , anchor = "c")

button7 = Checkbutton(root , text = "Can never be zero", command = key2)

button7.place(relx =.11 , rely =.50 , anchor = "c")

button8 = Checkbutton(root , text = "Depends on the strength of the dipole", command = key2)

button8.place(relx =.15 , rely =.55 , anchor = "c")

label2.place(relx =.19 , rely =.35 , anchor = "c")

label3 = Label(root , text = " 3.If a body is charged by rubbing it then it's weight..")

var3 = IntVar()

button9 = Checkbutton(root , text = "remains precisely constant", command = key3)

button9.place(relx =.13 , rely =.65 , anchor = "c")

button10 = Checkbutton(root , text = "Increases slightly", command = key3)

button10.place(relx =.11, rely =.70 , anchor = "c")

button11 = Checkbutton(root , text = "Decreases slightly", command = key3)

button11.place(relx =.11 , rely =.75 , anchor = "c")

button12 = Checkbutton(root , text = "May increase or decrease slightly", variable = var3 , command = key3)

button12.place(relx =.14, rely =.80 , anchor = "c")

label3.place(relx =.1 , rely =.60 , anchor = "c")

root.mainloop()

import Q2

**Quiz-12(continuation):**

from Tkinter import \*

def key4():

if var4.get() == 1:

print "correct"

else:

print "False"

def key5():

if var5.get() == 1:

print "correct"

else:

print "False"

def key6():

if var6.get() == 1:

print "correct"

else:

print "False"

root = Tk()

label4 = Label(root , text = " 4.If the flux of the electric field through a closed surface is zero..")

var4 = IntVar()

button14 = Checkbutton(root , text = "The electric field must be zero everywhere on the surface ", command = key4)

button14.place(relx =.13 , rely =.15 , anchor = "c" )

button15 = Checkbutton(root , text = "The electric field may be zero everywhere on the surface", variable = var4, command = key4)

button15.place(relx =.125 , rely =.20 , anchor = "c" )

button16 = Checkbutton(root , text = "The charge inside the surface must be zero", command = key4)

button16.place(relx =.1 , rely =.25 , anchor = "c")

button17 = Checkbutton(root , text = "The charge in the vicinity of the surface must be zero", command = key4)

button17.place(relx =.12 , rely =.30 , anchor = "c")

label4.place(relx =.12 , rely =.1 , anchor = "c")

label5 = Label(root , text = " 5.An electric dipole is placed at the centre of the sphere..")

var5 = IntVar()

button18 = Checkbutton(root , text = "The flux of the electric field through the sphere is zero.", variable = var5, command = key5)

button18.place(relx =.12 , rely =.40 , anchor = "c" )

button19 = Checkbutton(root , text = "The electric filed is zero at every point of the sphere", command = key5)

button19.place(relx =.115, rely =.45 , anchor = "c")

button20 = Checkbutton(root , text = "The electric field is not zero anywhere on the sphere.", command = key5)

button20.place(relx =.115 , rely =.50 , anchor = "c")

button21 = Checkbutton(root , text = "The electric field is zero on a circle of the sphere", command = key5)

button21.place(relx =.105 , rely =.55 , anchor = "c")

label5.place(relx =.11 , rely =.35 , anchor = "c")

label6 = Label(root , text = " 6.A positive point charge q is bought near an isolated metal cube ..")

var6 = IntVar()

button22 = Checkbutton(root , text = "The cube becomes negatively charged", command = key6)

button22.place(relx =.09 , rely =.65 , anchor = "c")

button23 = Checkbutton(root , text = "The cube becomes positively charged", command = key6)

button23.place(relx =.09, rely =.70 , anchor = "c")

button24 = Checkbutton(root , text = "The interior becomes positively charged and the surface becomes negatively charged", variable = var6, command = key6)

button24.place(relx =.18, rely =.75 , anchor = "c" )

button25 = Checkbutton(root , text = "The interior remains charge free and the surface gets non-uniform charge distribution.", command = key6)

button25.place(relx =.18, rely =.80 , anchor = "c")

label6.place(relx =.13 , rely =.60 , anchor = "c")

root.mainloop()

**Question Paper:**

from Tkinter import \*

def t2014():

def nextt():

master = Toplevel()

photo1 = PhotoImage(file = 'C:\Users\Rahul\Desktop\Computer Project \\2.gif')

label1 = Label(master , image = photo1)

label1.pack()

master.mainloop()

root = Toplevel()

photo = PhotoImage(file = 'C:\\Users\\Rahul\\Desktop\\Computer Project\\1.gif')

button = Button(root,text = 'next' ,command = nextt)

button.pack()

label = Label(root , image = photo)

label.pack()

root.mainloop()

def t2015():

def nextt():

master = Toplevel()

photo1 = PhotoImage(file = 'C:\\Users\\Rahul\\Desktop\\Computer Project\\4.gif')

label1 = Label(master , image = photo1)

label1.pack()

master.mainloop()

root = Toplevel()

photo = PhotoImage(file = 'C:\\Users\\Rahul\\Desktop\\Computer Project\\3.gif')

button = Button(root,text = 'next' ,command = nextt)

button.pack()

label = Label(root , image = photo)

label.pack()

root.mainloop()

def t2016():

def nextt():

master = Toplevel()

photo1 = PhotoImage(file = 'C:\\Users\\Rahul\\Desktop\\Computer Project\\6.gif')

label1 = Label(master , image = photo1)

label1.pack()

master.mainloop()

root = Toplevel()

photo = PhotoImage(file = 'C:\\Users\\Rahul\\Desktop\\Computer Project\\5.gif')

button = Button(root,text = 'next' ,command = nextt)

button.pack()

label = Label(root , image = photo)

label.pack()

root.mainloop()

def confirm():

l = [t2014,t2015,t2016]

l1 = [2014,2015,2016]

value = entry1.get()

for i in range(len(l)):

if int(value)== l1[i]:

l[i]()

root1 = Tk()

label1 = Label(root1 , text = "Enter the year")

entry1 = Entry(root1)

label1.grid(row = 0 , column = 1)

entry1.grid(row = 0 , column = 2)

button1 = Button(root1 , text = "Enter" , command = confirm)

button1.grid(columnspan = 2)

root1.mainloop()

**FACTS**

from Tkinter import \*

root=Tk()

def Fact1():

print '''FACT NUMBER #1:

As Small as a Sugar Cube

Atoms have 99.9999999999999 per cent empty space. If you force all the atoms jointly, eradicating the space between them, grinding them down so the all those vast empty cathedrals would compressed into the small nuclei, a single teaspoon or sugar cube of the resulting mass would weigh five billion tons; about ten times the weight of all the humans who are currently alive.'''

def Fact2():

print '''FACT NUMBER #2:

The Inconsistent Light

We are always told to be as fast as light. Well here is answer for all those who hate being told that. Light travels fast only in a vacuum. It is slowed whenever it passes through something, being measured as traveling as slowly as just 38 miles per hour at absolute zero (-273.15C) through ultra-cooled rubidium.'''

def Fact3():

print '''FACT NUMBER #3:

The Mysterious Question

Despite all the advances made in astrophysics in recent years, we dont know what makes up the majority of the universe. The possible to make reasonable estimates of the mass of the universe, except the visible matter (stars, planets, stellar objects) only accounts for 2% of that;what exactly makes up the rest-so- called 'dark matter' and 'dark energy' remains a mystery'''

def Fact4():

print '''FACT NUMBER #4:

The Smart Swimmer

One of the most amazing & interesting fact about Physics on earth is The Dead Sea which is known for its density due to the presence of salt, as a result of which you can easily float on it without drowning, so one can always claim to be a swimmer there.'''

def Fact5():

print '''FACT NUMBER #5:

Ultimate expansion:

It is proved by scientific theories that the universe is constantly expanding. It is expanding at a decent pace and it is believed that galaxies will evaporate in the coming 10^19 to 10^20 years. It has been learnt from a number of theories by different Physicists worldwide that only White Dwarfs (a type of star) would be able to survive as their lifetime is more than 10^32 years.'''

l1=Label(root,text="PHYSICS FACTS!",fg="blue",cursor="dot",bg="pink")

b=Button(root,text="Fact Number 1",activebackground="green",fg='blue',bg='yellow',command=Fact1)

b1=Button(root,text="Fact Number 2",activebackground="green",fg='blue',bg='yellow',command=Fact2)

b2=Button(root,text="Fact Number 3",activebackground="green",fg='blue',bg='yellow',command=Fact3)

b3=Button(root,text="Fact Number 4",activebackground="green",fg='blue',bg='yellow',command=Fact4)

b4=Button(root,text="Fact Number 5",activebackground="green",fg='blue',bg='yellow',command=Fact5)

l1.pack()

b.pack()

b1.pack()

b2.pack()

b3.pack()

b4.pack()

root.mainloop()

**FEEDBACK PRGORAM**

class feedback:

def \_\_init\_\_(self):

self.fb=""

self.name=""

self.rate=0

def inp(self):

self.name=raw\_input("Enter your name:")

self.fb=raw\_input("Enter your feed back:")

self.rate=input("Enter rating out of 5:")

Obj=feedback()

Obj.inp()

fw=open("Feedbacknames.txt","a")

fw.write(Obj.name+" ")

fw.write(Obj.fb+" ")

fw.write(str(Obj.rate)+"\n")

fw.close()

**VECTORS**

"""This program is for finding the sum,difference or multiplication of two vectors"""

def vector():

vect1=eval(raw\_input("Enter the coefficients of the vector in list fom:")) #Input of two vectors in the form of i,j and k.

vect2=eval(raw\_input("Enter the coefficients of the vector in list fom:"))

choice=raw\_input(" Addition,Subtraction,Dot product or cross product?:") #Choice for the user from the given operations

if(choice=="Dot" or choice=="Dot product" or choice=="dot product" or choice=="dot"):

print vect1[0]\*vect2[0]+vect1[1]\*vect2[1]+vect1[2]\*vect2[2],"Is the magnitude of the dot produchoicet"

elif(choice=="Cross" or choice=="Cross product" or choice=="cross product" or choice=="cross"):

print vect1[1]\*vect2[2]-vect1[2]\*vect2[1],"i",-1\*(vect1[0]\*vect2[2]-vect2[0]\*vect1[2]),"j",(vect1[0]\*vect2[1]-vect2[0]\*vect2[1]),"k"

elif(choice=="Add" or choice=="Addition" or choice=="Sum" or choice=="addition"):

print vect1[0]+vect2[0],"i",+vect1[1]+vect2[1],"j",+vect1[2]+vect2[2],"k"

elif(choice=="Subtract" or choice=="difference" or choice=="subtract" or choice=="sub"):

print vect1[0]-vect2[0],"i",+vect1[1]-vect2[1],"j",+vect1[2]-vect2[2],"k"

else:

print "Wrong choice,Exiting....." #if entered a wrong choice then program exits

**CHAPTERS**

#OPENING TKINTER WINDOW FROM ANOTHER CLASS USING INHERITANCE

from Tkinter import \*

class Information(object):

def info1(self):

root1=Tk()

self.label= Label(root1, text='''Chapter\_1: Physical World

Physics \_ scope and excitement; nature of physical laws; Physics, technology and society.

Chapter\_2: Units and Measurements

Need for measurement: Units of measurement; systems of units; SI units, fundamental and derived units. Length, mass and time measurements; accuracy and precision of measuring instruments; errors in measurement; significant figures.

Dimensions of physical quantities, dimensional analysis and its applications.''',justify="center")

self.label.place(x=20,y=40)

self.weight=Label(root1,text="Weightage 3",font=("Helvetica",14))

self.weight.pack()

root1.mainloop()

def info2(self):

root2=Tk()

self.label=Label(root2,text='''Frame of reference, Motion in a straight line: Position-time graph, speed and velocity.

Elementary concepts of differentiation and integration for describing motion.Uniform and non-uniform motion, average speed and instantaneous velocity. Uniformly accelerated motion, velocity time and position-time graphs.

Relations for uniformly accelerated motion (graphical treatment).''',justify="center")

self.label.place(x=20,y=40)

self.weight=Label(root2,text="Weightage 10",font=("Helvetica",14))

self.weight.pack()

root2.mainloop()

def info3(self):

root3=Tk()

self.label=Label(root3,text='''Intuitive concept of force. Inertia, Newton's first law of motion; momentum and Newton's second law of motion; impulse; Newton's third law of motion.

Law of conservation of linear momentum and its applications.

Equilibrium of concurrent forces. Static and kinetic friction, laws of friction, rolling friction, lubrication.

Dynamics of uniform circular motion: Centripetal force, examples of circular motion (vehicle on a level circular road, vehicle on banked road).''',justify="center")

self.label.place(x=20,y=40)

self.weight=Label(root3,text="Weightage 10",font=("Helvetica",14))

self.weight.pack()

root3.mainloop()

def info4(self):

root4=Tk()

self.label=Label(root4,text='''Work done by a constant force and a variable force; kinetic energy, work-energy theorem, power.

Notion of potential energy, potential energy of a spring, conservative forces: conservation of mechanical energy (kinetic and potential energies); non-conservative forces: motion in a vertical circle; elastic and inelastic collisions in one and two dimensions.''',justify="center")

self.label.place(x=20,y=40)

self.weight=Label(root4,text="Weightage 6",font=("Helvetica",14))

self.weight.pack()

root4.mainloop()

def info5(self):

root5=Tk()

self.label=Label(root5,text='''Centre of mass of a two-particle system, momentum conservation and centre of mass motion.

Centre of mass of a rigid body; centre of mass of a uniform rod.

Moment of a force, torque, angular momentum, laws of conservation of angular momentum and its applications.

Equilibrium of rigid bodies, rigid body rotation and equations of rotational motion, comparison of linear and rotational motions.

Moment of inertia, radius of gyration.Values of moments of inertia, for simple geometrical objects (no derivation). Statement of parallel and perpendicular axes theorems and their applications.''',justify="center")

self.label.place(x=20,y=40)

self.weight=Label(root5,text="Weightage 6",font=("Helvetica",14))

self.weight.pack()

root5.mainloop()

def info6(self):

root6=Tk()

self.label=Label(root6,text='''Keplar's laws of planetary motion.The universal law of gravitation.

Acceleration due to gravity and its variation with altitude and depth.

Gravitational potential energy and gravitational potential. Escape velocity. Orbital velocity of a satellite. Geo-stationary satellites.''',justify="center")

self.label.place(x=20,y=40)

self.weight=Label(root6,text="Weightage 5",font=("Helvetica",14))

self.weight.pack()

root6.mainloop()

def info7(self):

root7=Tk()

self.label=Label(root7,text='''Chapter\_9: Mechanical Properties of Solids

Elastic behaviour, Stress-strain relationship, Hooke's law, Young's modulus, bulk modulus, shear modulus of rigidity, Poisson's ratio; elastic energy.

Chapter\_10: Mechanical Properties of Fluids

Pressure due to a fluid column; Pascal's law and its applications (hydraulic lift and hydraulic brakes). Effect of gravity on fluid pressure.

Viscosity, Stokes' law, terminal velocity, streamline and turbulent flow, critical velocity.Bernoulli's theorem and its applications.

Surface energy and surface tension, angle of contact, excess of pressure across a curved surface, application of surface tension ideas to drops, bubbles and capillary rise.

Chapter\_11: Thermal Properties of Matter

Heat, temperature, thermal expansion; thermal expansion of solids, liquids and gases, anomalous expansion of water; specific heat capacity; Cp, Cv - calorimetry; change of state - latent heat capacity.

Heat transfer-conduction, convection and radiation, thermal conductivity, Qualitative ideas of Blackbody radiation, Wein's displacement Law, Stefan's law, Green house effect. ''',justify="center")

self.label.place(x=20,y=40)

self.weight=Label(root7,text="Weightage 10",font=("Helvetica",14))

self.weight.pack()

root7.mainloop()

def info8(self):

root8=Tk()

self.label=Label(root8,text='''Thermal equilibrium and definition of temperature (zeroth law of thermodynamics).Heat, work and internal energy. First law of thermodynamics. Isothermal and adiabatic processes.

Second law of thermodynamics: reversible and irreversible processes. Heat engine and refrigerator.''',justify="center")

self.label.place(x=20,y=40)

self.weight=Label(root8,text="Weightage 5",font=("Helvetica",14))

self.weight.pack()

root8.mainloop()

def info9(self):

root9=Tk()

self.label=Label(root9,text='''Equation of state of a perfect gas, work done in compressing a gas.

Kinetic theory of gases - assumptions, concept of pressure. Kinetic interpretation of temperature; rms speed of gas molecules; degrees of freedom, law of equi-partition of energy (statement only) and application to specific heat capacities of gases;

concept of mean free path, Avogadro's number.''')

self.label.place(x=20,y=40)

self.weight=Label(root9,text="Weightage 5",font=("Helvetica",14))

self.weight.pack()

root9.mainloop()

def info10(self):

root10=Tk()

self.label=Label(root10,text='''Chapter\_14: Oscillations

Periodic motion - time period, frequency, displacement as a function of time. Periodic functions.

Simple harmonic motion (S.H.M) and its equation; phase; oscillations of a spring-restoring force and force constant; energy in S.H.M. Kinetic and potential energies; simple pendulum derivation of expression for its time period.

Free, forced and damped oscillations (qualitative ideas only), resonance.

Chapter\_15: Waves

Wave motion. Transverse and longitudinal waves, speed of wave motion. Displacement relation for a progressive wave. Principle of superposition of waves, reflection of waves, standing waves in strings and organ pipes, fundamental mode and harmonics,

Beats, Doppler effect.''')

self.label.place(x=20,y=40)

self.weight=Label(root10,text="Weightage 10",font=("Helvetica",14))

self.weight.pack()

root10.mainloop()

#MAIN WINDOW

class MyFirstGUI(Information):

def \_\_init\_\_(self, master):

self.master = master

master.title("Chapters")

self.label = Label(master, text="PHYSICS CLASS 11 CHAPTERS",font=("Helvetica", 16))

self.label.pack()

self.label1= Label(master, text="1.Physical World and Measurement")

self.label1.place(x=20,y=40)

self.info\_button1= Button(master, text="Info", command=self.info1)

self.info\_button1.place(x=600,y=40)

self.label2 = Label(master, text="2.Kinematics")

self.label2.place(x=20,y=60)

self.info\_button2= Button(master, text="Info", command=self.info2)

self.info\_button2.place(x=600,y=60)

self.label3 = Label(master, text="3.Laws of Motion")

self.label3.place(x=20,y=80)

self.info\_button3 = Button(master, text="Info", command=self.info3)

self.info\_button3.place(x=600,y=80)

self.label4 = Label(master, text="4.Work, Energy and Power")

self.label4.place(x=20,y=100)

self.info\_button4 = Button(master, text="Info", command=self.info4)

self.info\_button4.place(x=600,y=100)

self.label5 = Label(master, text="5.Motion of System of Particles")

self.label5.place(x=20,y=120)

self.info\_button5 = Button(master, text="Info", command=self.info5)

self.info\_button5.place(x=600,y=120)

self.label6 = Label(master, text="6.Gravitation")

self.label6.place(x=20,y=140)

self.info\_button6= Button(master, text="Info", command=self.info6)

self.info\_button6.place(x=600,y=140)

self.label7 = Label(master, text="7.Properties of Bulk Matter")

self.label7.place(x=20,y=160)

self.info\_button7 = Button(master, text="Info", command=self.info7)

self.info\_button7.place(x=600,y=160)

self.label8 = Label(master, text="8.Thermodynamics")

self.label8.place(x=20,y=180)

self.info\_button8 = Button(master, text="Info", command=self.info8)

self.info\_button8.place(x=600,y=180)

self.label9 = Label(master, text="9.Kinetic Theory of Gases")

self.label9.place(x=20,y=200)

self.info\_button9 = Button(master, text="Info", command=self.info9)

self.info\_button9.place(x=600,y=200)

self.label10 = Label(master, text="10.Oscillation & Waves")

self.label10.place(x=20,y=220)

self.info\_button10 = Button(master, text="Info", command=self.info10)

self.info\_button10.place(x=600,y=220)

root = Tk()

my\_gui = MyFirstGUI(root)

root.mainloop()

**FREE BODY DIAGRAM 1**

#python program for FBD's of blocks attached by string one below the other

def two\_blocks(m1,m2):

g=10

t=(m2\*g)

t1=(m1\*g)+t

return t1,"N",t,"N"

m1=float(raw\_input("enter the mass of one block"))

m2=float(raw\_input("enter the mass of the second block"))

print "the two tensions are",two\_blocks(m1,m2)

#and

#to calculate the acceleration of the block

def acc\_blocks(a,b):

g=10

acc1=(g\*(2\*a-b))/(4\*a+b)

acc2=2\*acc1

return acc1,acc2

a=float(raw\_input("enter the mass of heavier block"))

b=float(raw\_input("enter the mass of lighter block"))

result=acc\_blocks(a,b)

print "the acceleration of the block is",result

#to calculate the tension of bob in a moving lift

m=float(raw\_input("enter the mass of the bob"))

n=raw\_input("enter the condition ")

acc=float(raw\_input("enter the acc of the lift"))

g=10

if n=="up" and acc>0:

t=m\*(g+acc)

print "the tension is ",t

elif n=="up" and acc<0:

t=m\*(g+acc)

print "the tension is ",t

elif n=="down" and acc>0:

t=m\*(g-acc)

print "the tension is ",t

elif n=="down" and acc<0:

t=m\*(g-acc)

print "the tension is ",t

**FREE BODY DIAGRAM 2**

#Python program for FBD's of blocks attached by string one below the other

def two\_blocks(m1,m2):

g=10

t=(m2\*g)

t1=(m1\*g)+t

return t1,"N",t,"N"

#and

#to calculate the acceleration of the block

def acc\_blocks(a,b):

g=10

acc1=(g\*(2\*a-b))/(4\*a+b)

acc2=2\*acc1

return acc1,acc2

def lift():

#to calculate the tension of bob in a moving lift

m=float(raw\_input("Enter the mass of the bob"))

n=int(raw\_input(" 1.Is the lift is moving up,\n 2.Or is the lift moving down"))

acc=float(raw\_input("Enter the acc of the lift"))

g=10

if n==1 and acc>0:

t=m\*(g+acc)

return t

elif n==2 and acc>0:

t=m\*(g-acc)

return t

print(" Press 1 for Tension related problem\n Press 2 for Acceleration of the blocks\n Press 3 for Lift based problems")

choice=int(raw\_input(""))

if(choice==2):

a=float(raw\_input("Enter the mass of heavier block"))

b=float(raw\_input("Enter the mass of lighter block"))

result=acc\_blocks(a,b)

print "The acceleration of the blocks are",result

if(choice==1):

m1=float(raw\_input("Enter the mass of one block"))

m2=float(raw\_input("Enter the mass of the second block"))

print "The two tensions are",two\_blocks(m1,m2)

if(choice==3):

tension=lift()

print "The value of Tension is",tension

**Moduli:**

def modulus():

print "You have chosen to calculate elastic moduli."

print """What would you like to calculate?

1)Young's Modulus

2)Bulk's Modulus

3)Shear Modulus"""

choice1=int(raw\_input("Enter your choice(NUMBER)."))

if choice1==1:

print "You have chosen to calculate Young's Modulus."

ystress=float(raw\_input("Enter the stress:"))

ystrain=float(raw\_input("Enter the longitudinal strain:"))

ymodulus=ystress/ystrain

print "FORMULA: Young's Modulus= Stress/Longitudinal Strain"

print "Young's Modulus=",ymodulus

elif choice1==2:

print "You have chosen to calculate Bulk's Modulus."

bstress=float(raw\_input("Enter the stress:"))

bstrain=float(raw\_input("Enter the volumetric strain:"))

bmodulus=bstress/bstrain

print "FORMULA: Bulk's Modulus= Stress/Volumetric Strain"

print "Bulk's Modulus=",bmodulus

elif choice1==3:

print "You have chosen to calculate Shear's Modulus."

shstress=float(raw\_input("Enter the tangential stress:"))

shstrain=float(raw\_input("Enter the shear strain:"))

shmodulus=shstress/shstrain

print "FORMULA: Bulk's Modulus= Tangential Stress/Shear Strain"

print "Shear Modulus=",shmodulus

else:

print "ERROR: Choice of calculation doesn't exist."

**Other:**

def othersolidprop():

print "You have chosen to calculate other solid properties."

print """What would you like to calculate?

1)Compressibility

2)Elastic Potenial Energy

3)Poisson's Ratio"""

choice1=int(raw\_input("Enter your choice(NUMBER)."))

if choice1==1:

print "You have chosen to calculate Compressibility."

bmodu=float(raw\_input("Enter the Bulk's Modulus:"))

compr=1/bmodu

print "FORMULA: Compressiblity= 1/Bulk's Modulus"

print "Compressibilty=",compr

elif choice1==2:

print "You have chosen to calculate Elastic Potential Energy."

ymodulus1=float(raw\_input("Enter the Young's Modulus:"))

strain1=float(raw\_input("Enter the strain:"))

ymodulus=(1/2)\*ymodulus1

strain=(strain1)\*\*2

epe=ymodulus/strain

print "FORMULA: Elastic Potential Energy=1/2 \* Young's Modulus/Strain^^2 "

print "Elastic Potential Energy=",epe

elif choice1==3:

print "You have chosen to calculate Poisson's Ratio."

lastrain=float(raw\_input("Enter the Lateral Strain:"))

lostrain=float(raw\_input("Enter the Longitudinal Strain:"))

poratio=lastrain/lostrain

print "FORMULA: Poisson's Ratio= Lateral Strain/Longitudinal Strain"

print "Poisson's Ratio=",poratio

else:

print "ERROR: The choice of calculation doesn't exist."

**Second-Law:**

def SecondLawOfMotion():

print """Based on Second Law,what would you like to calculate?

1)Acceleration

2)Force

3)Mass

4)Momentum"""

choice1=int(raw\_input("Enter your choice:"))

if choice1==1:

print "ACCELERATION."

force=float(raw\_input("Enter force:"))

mass=float(raw\_input("Enter mass:"))

accel=force/mass

print "FORMULA USED: Acceleration=Force/Mass"

print "The acceleration is=",accel

if choice1==2:

print "FORCE."

mass=float(raw\_input("Enter mass:"))

accel=float(raw\_input("Enter acceleration:"))

force=mass\*accel

print "FORMULA USED: Force=Mass\*Acceleration"

print "The force is=",force

if choice1==3:

print "MASS."

accel=float(raw\_input("Enter acceleration:"))

force=float(raw\_input("Enter force:"))

mass=force/accel

print "FORMULA USED: Mass=Force/Acceleration"

print "The mass is=",mass

if choice1==4:

print "MOMENTUM."

mass=float(raw\_input("Enter mass:"))

velocity=float(raw\_input("Enter velocity:"))

momentum=mass\*velocity

print "FORMULA USED: Momentum=Mass\*Velocity"

print "The momentum is=",momentum

**Stress:**

def stress(): #Stress Module

print "You have chosen to calculate stress."

print "FORMULA: Stress=Force/Area"

choice=int(raw\_input("Type 1 for SI units and 2 for CGS units."))

force=float(raw\_input("Enter the force:"))

area=float(raw\_input("Enter the area:"))

stress=force/area

if choice==1:

print "The stress =",stress,"N/m^^2"

elif choice==2:

print "The stress =",stress,"dyne/cm^^2"

else:

print "ERROR:Choice of units doesn't exist."

**Strain:**

def strain():

print "You have chosen to calculate strain."

print """What would you like to calculate?

1) Longitudinal Strain

2) Volumetric Strain

3) Lateral Strain"""

choice1=int(raw\_input("Enter your choice(NUMBER)."))

if choice1==1:

print "You have chosen to calculate Longitudinal strain."

length=float(raw\_input("Enter the length:"))

changelength=float(raw\_input("Enter the change in length:"))

print "FORMULAE: Longitudinal Strain= change in length/original length"

lstrain=changelength/length

print "The longitudinal strain=",lstrain

elif choice1==2:

print "You have chosen to calculate Volumetric Strain."

volume=float(raw\_input("Enter the volume:"))

changevolume=float(raw\_input("Enter the change in volume:"))

print "FORMULAE: Volumetric Strain= change in volume/original volume"

vstrain=changevolume/volume

print "The volumetric strain=",vstrain

elif choice1==3:

print "You have chosen to calculate Lateral Strain."

laterallength=float(raw\_input("Enter the lateral length:"))

changelaterallength=float(raw\_input("Enter the change in lateral length:"))

print "FORMULAE: Lateral Strain= -(Change In Lateral Length/Lateral Length)"

lateralstrain1=changelaterallength/laterallength

lateralstrain=-(lateralstrain1)

print "The lateral strain=",lateralstrain

else:

print "ERROR: Choice of calculation doesn't exist."

**Centre:**

#to calculate centre of mass on x and y plane.

def centre():

mass=eval(raw\_input("Enter the masses(in lists):"))

x=eval(raw\_input("Enter the x coordinates(in list):"))

y=eval(raw\_input("Enter the y coordinates(in list):"))

length=len(mass)

m=[]

for a in range(0,length):

m.insert(a,mass[a]\*x[a])

length=len(m)

s=0

for j in range(0,length):

s=s+m[j]

s1=0

for k in range(0,length):

s1=s1+mass[k]

X=float((s))/float(s1)

print "The x coordinate centre of mass:",X

l=[]

for b in range(0,length):

l.insert(b,mass[b]\*y[b])

length=len(l)

s2=0

for c in range(0,length):

s2=s2+l[c]

s3=0

for o in range(0,length):

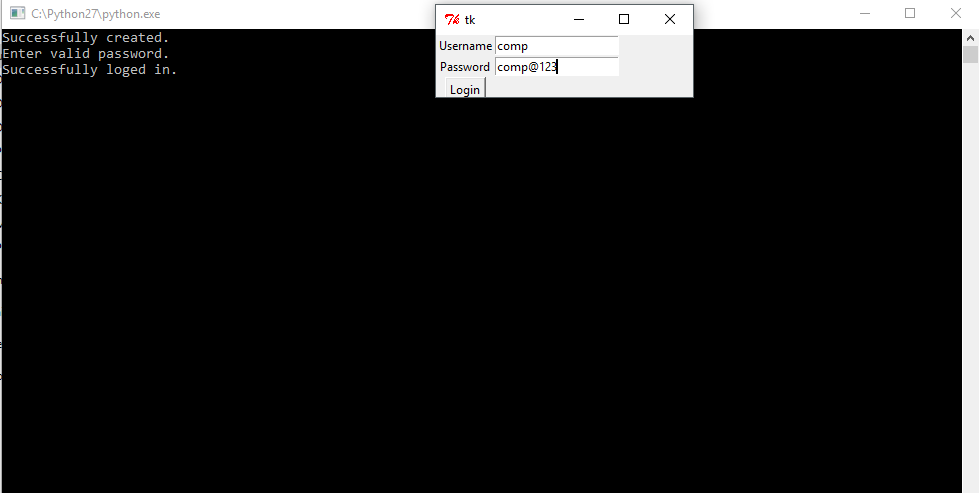
s3=s3+mass[o]

Y=float((s2))/float(s3)

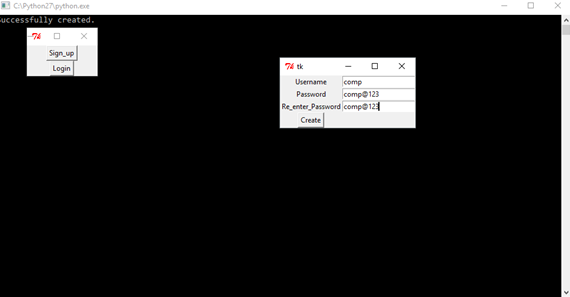
print "The x coordinate centre of mass:",Y

**Screenshots:**

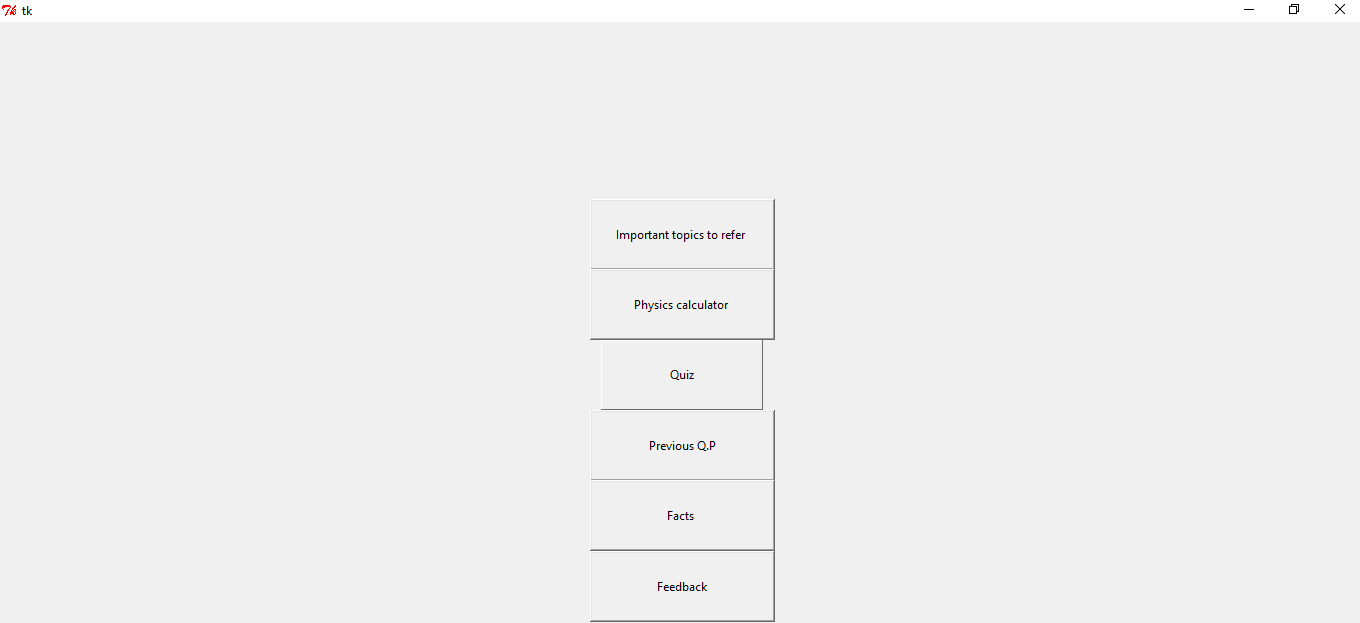
**Login:**



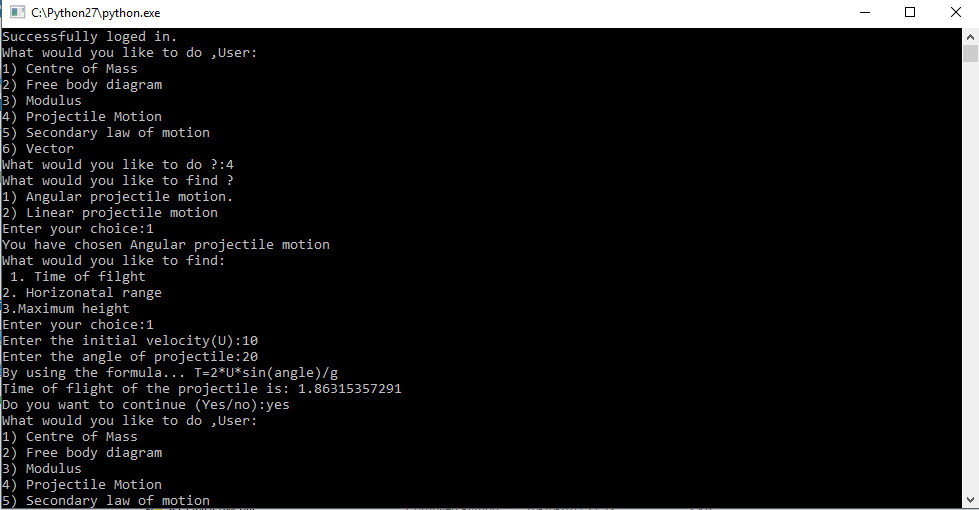
**Sign-up:**



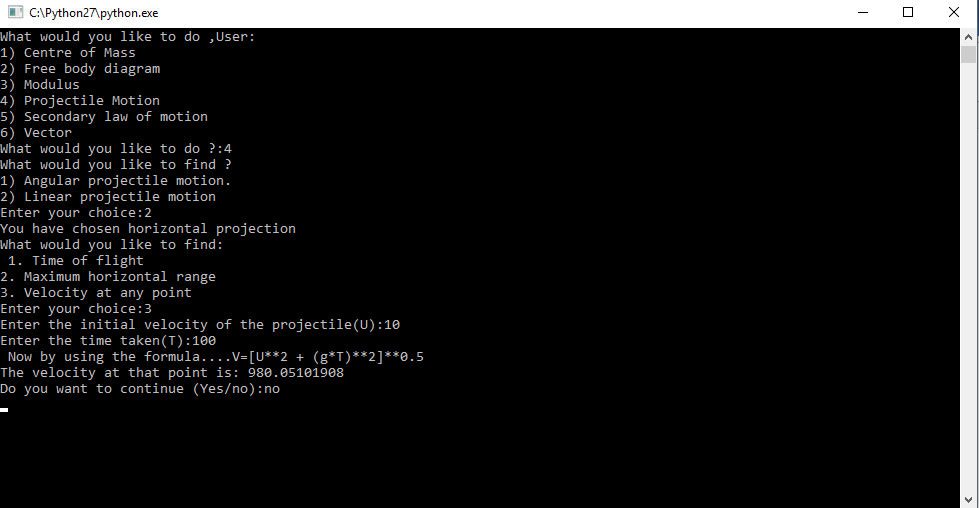
**Main:**

****

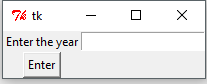
**Angular Projection:**

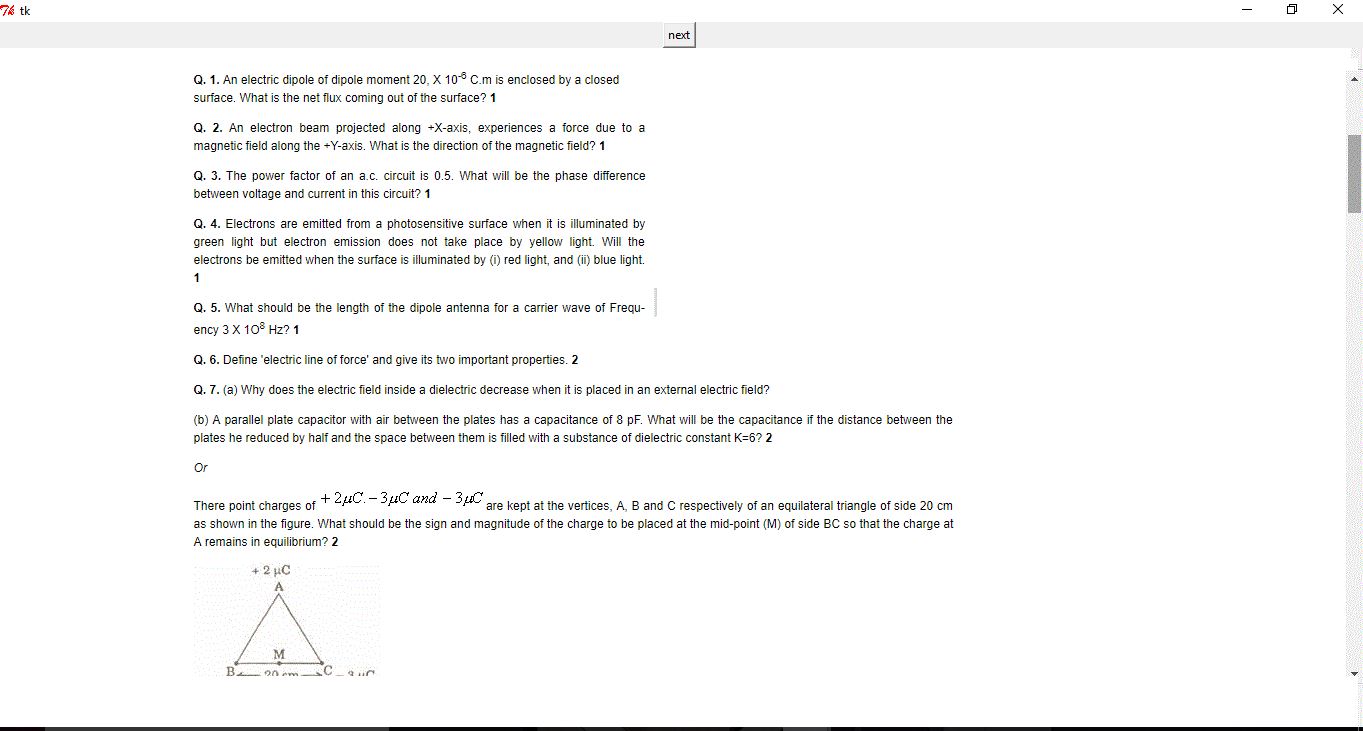
****

**Horizontal Projection:**

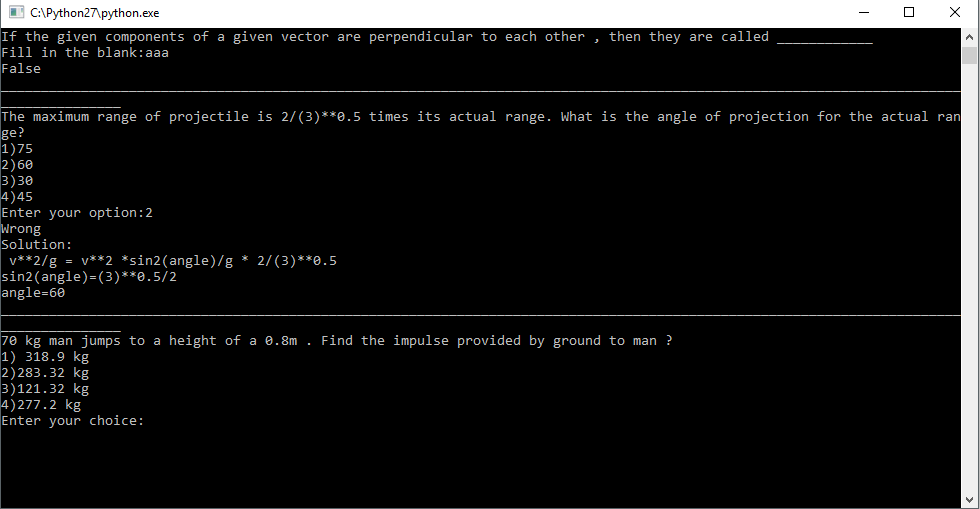
****

**Question -\_paper:**

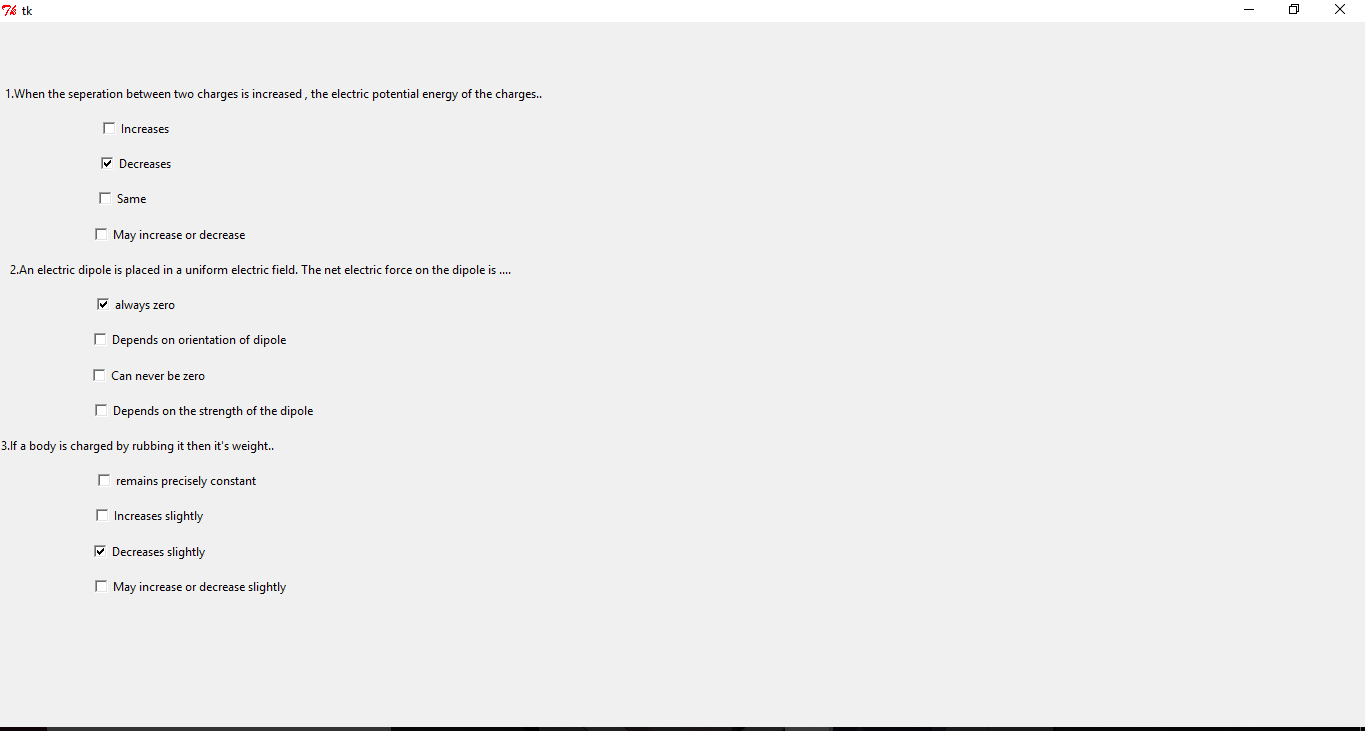
****

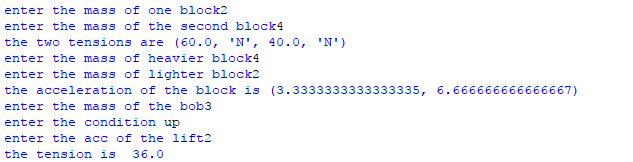
****

**Quiz-11:**

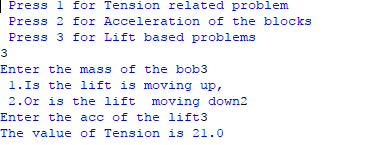
****

**Quiz-12:**

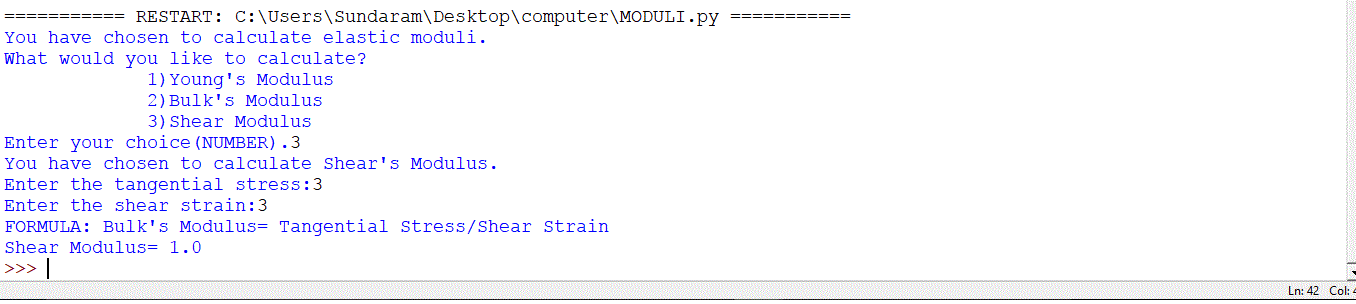
****

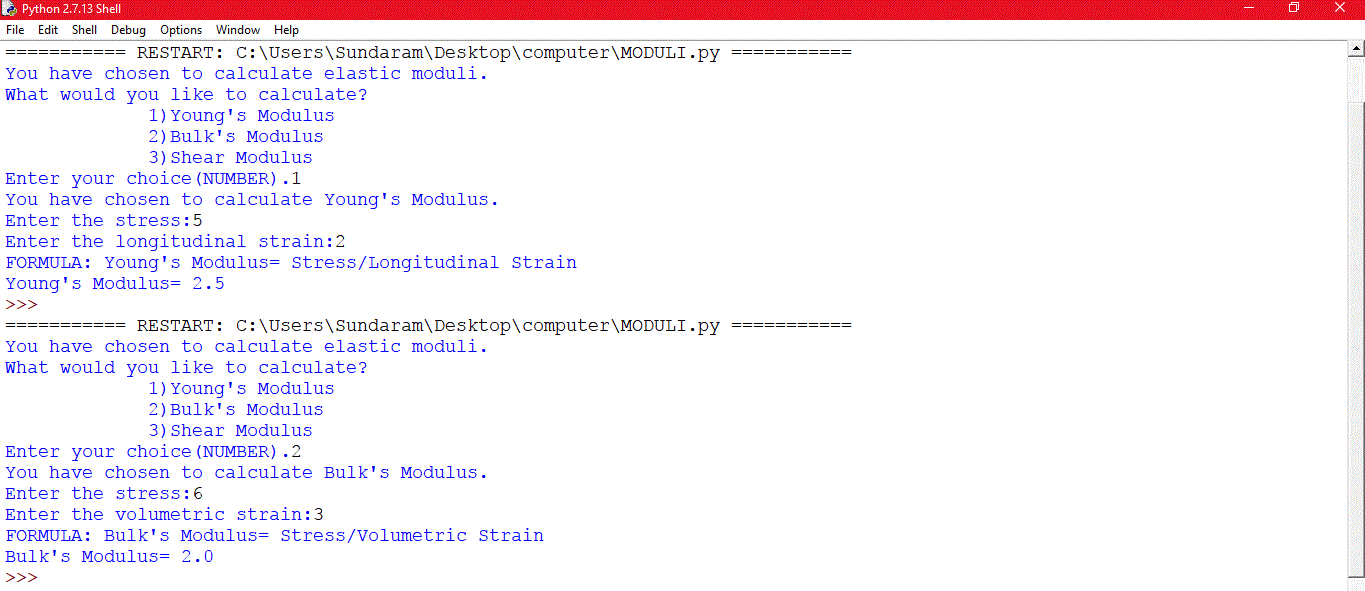
**FREE BODY DIAGRAM 1:**

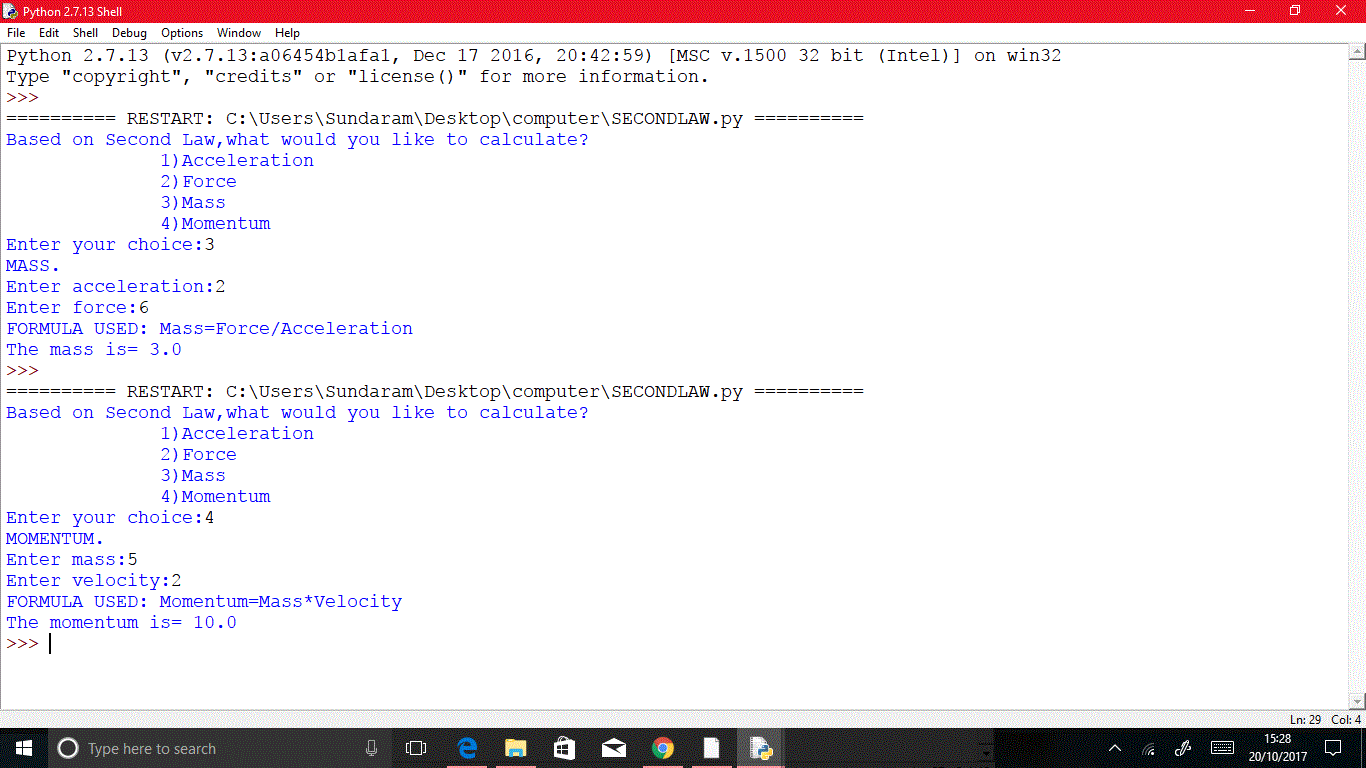
**FREE BODY DIAGRAM 2:**

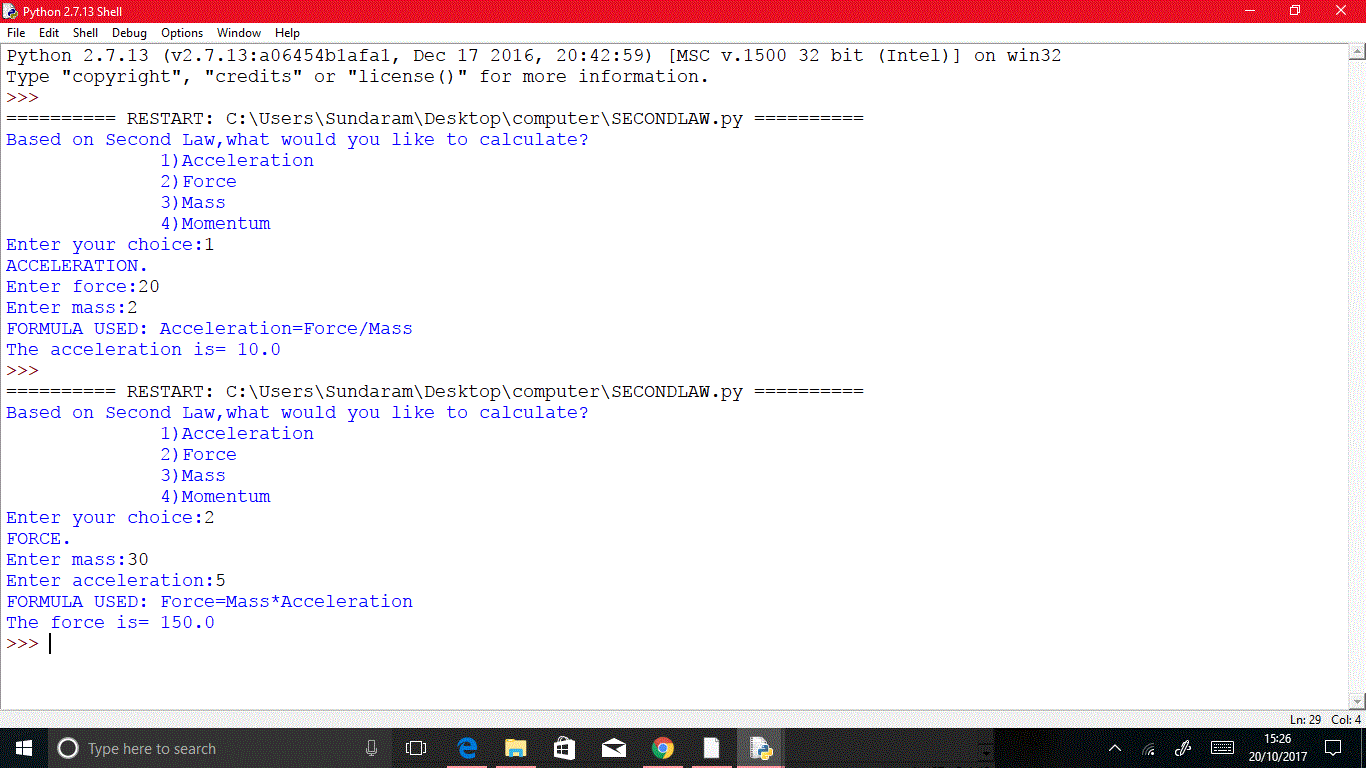


**Moduli:**

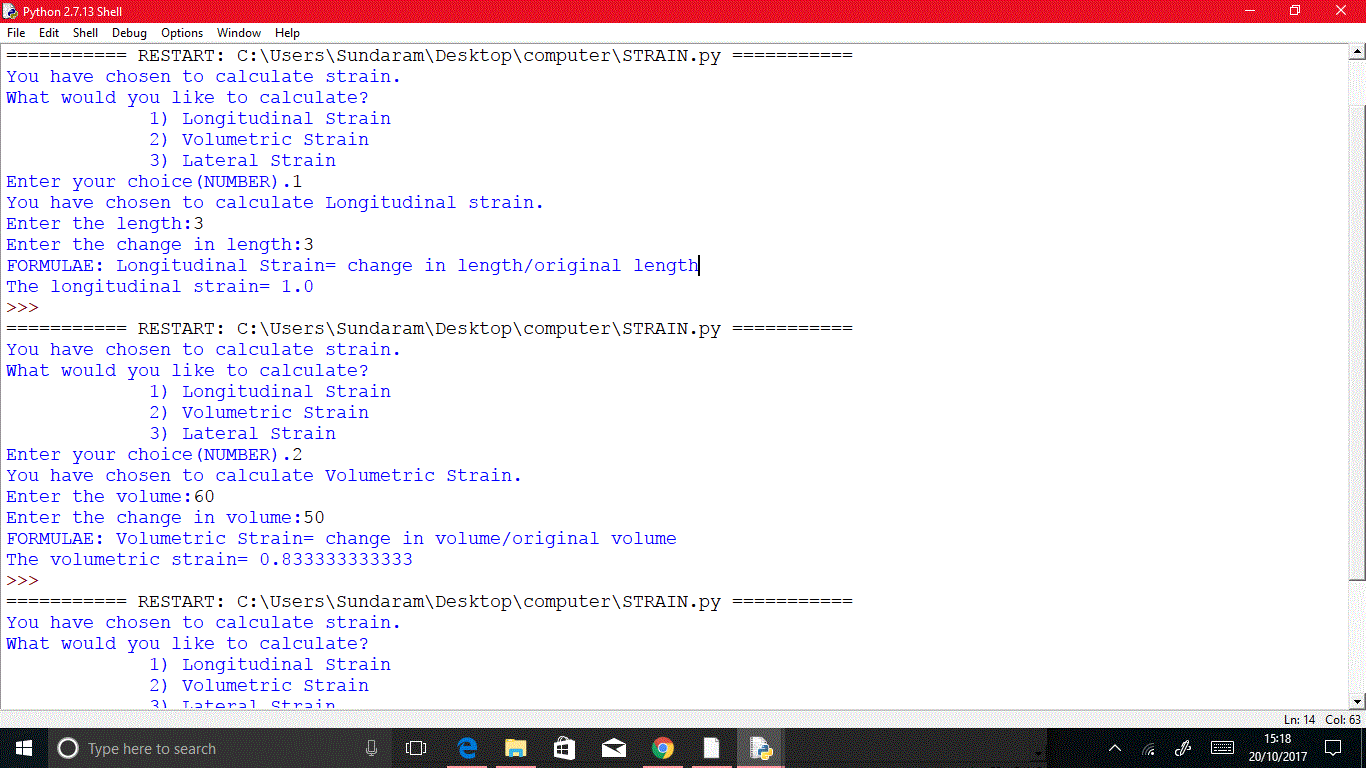


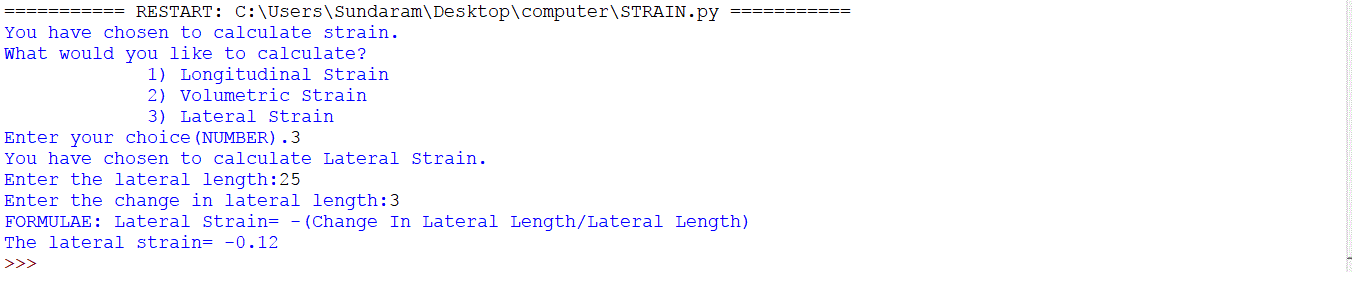


**Second-Law:** 

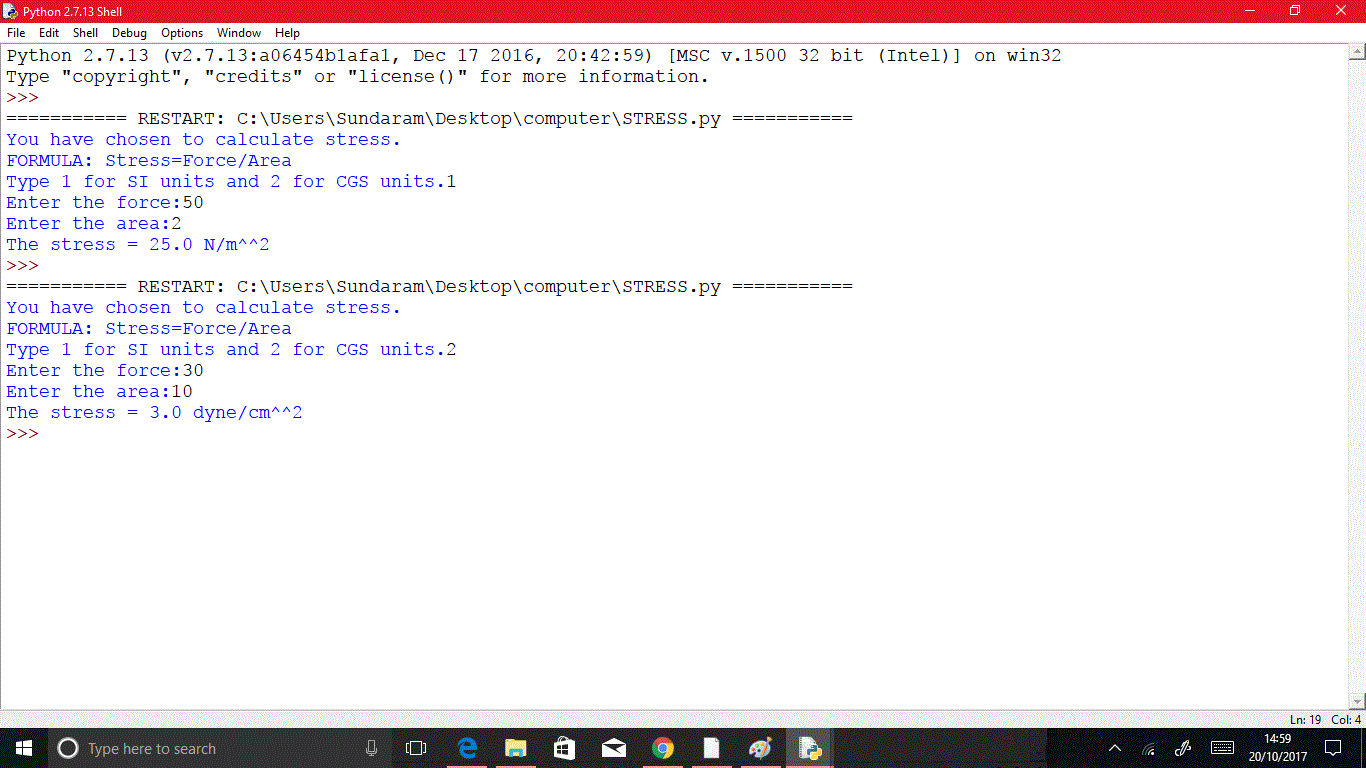


**Strain:**

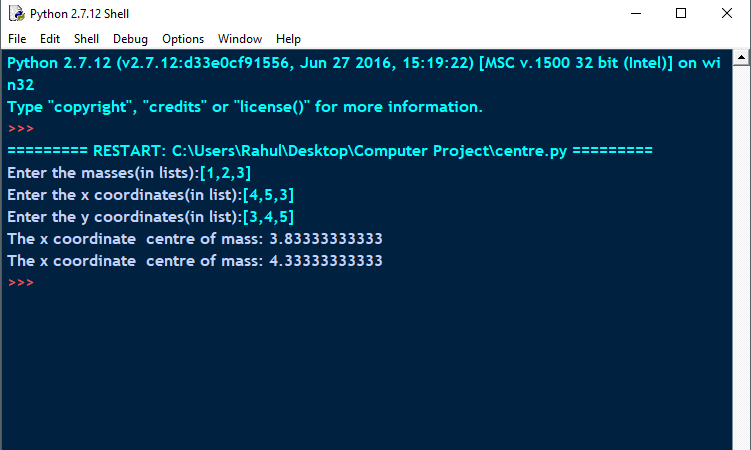


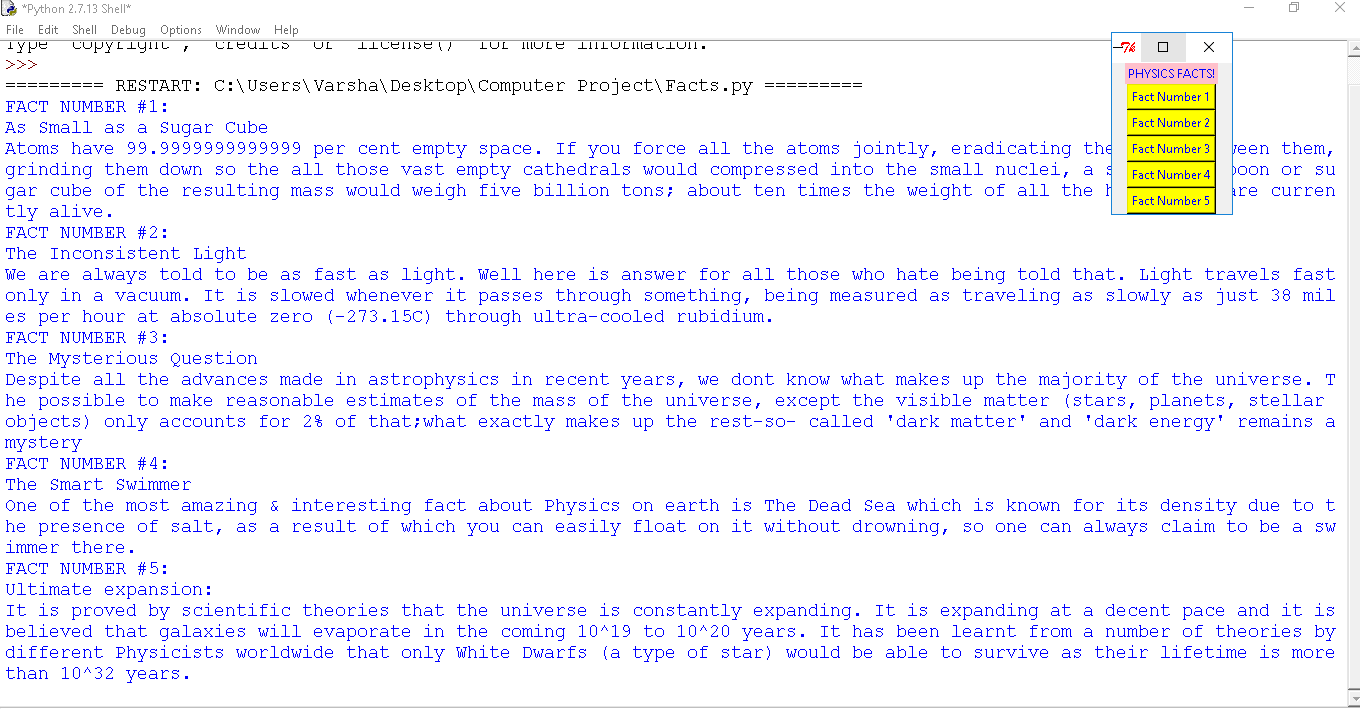


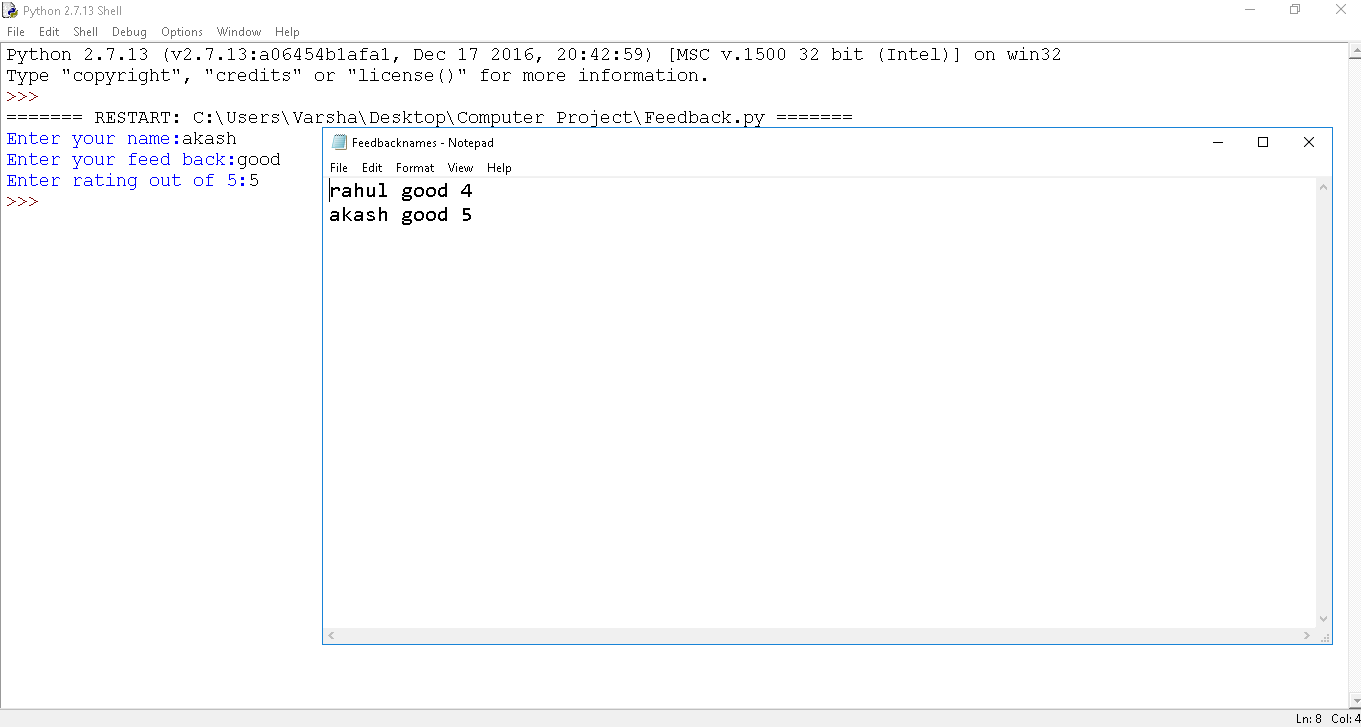
**Stress:**

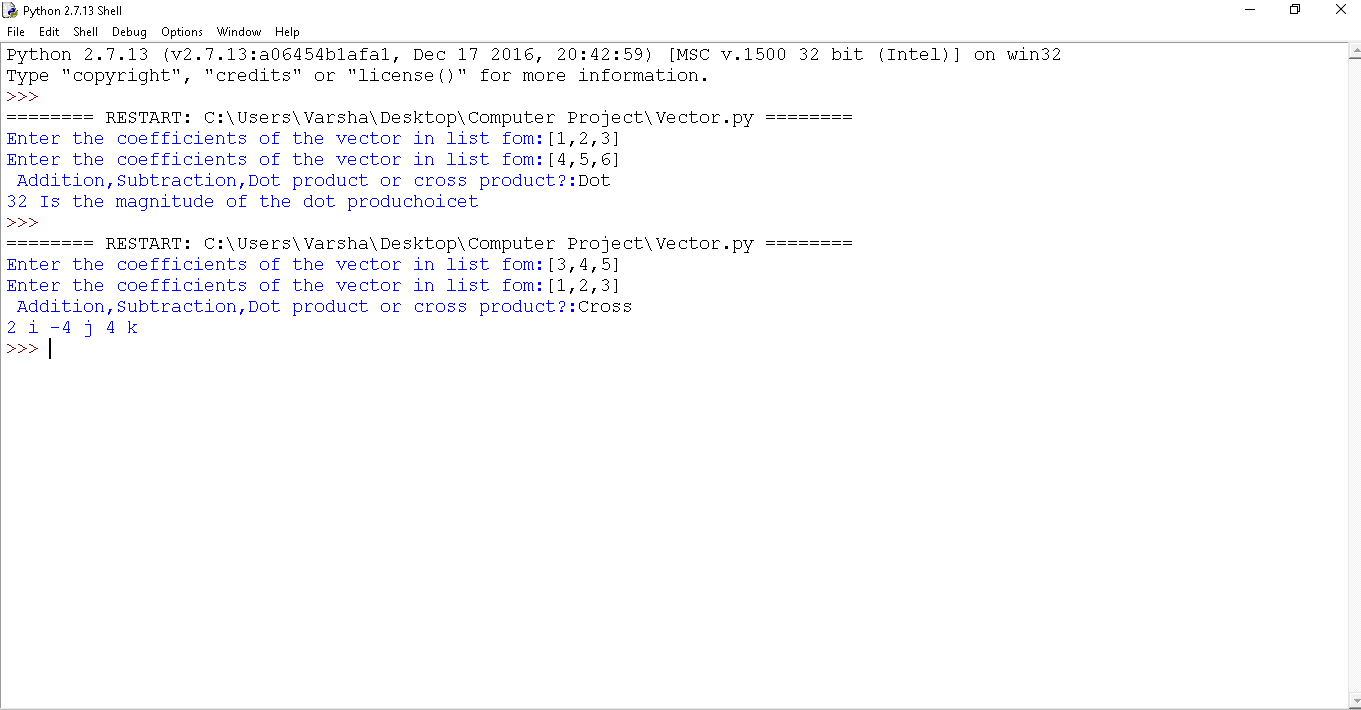


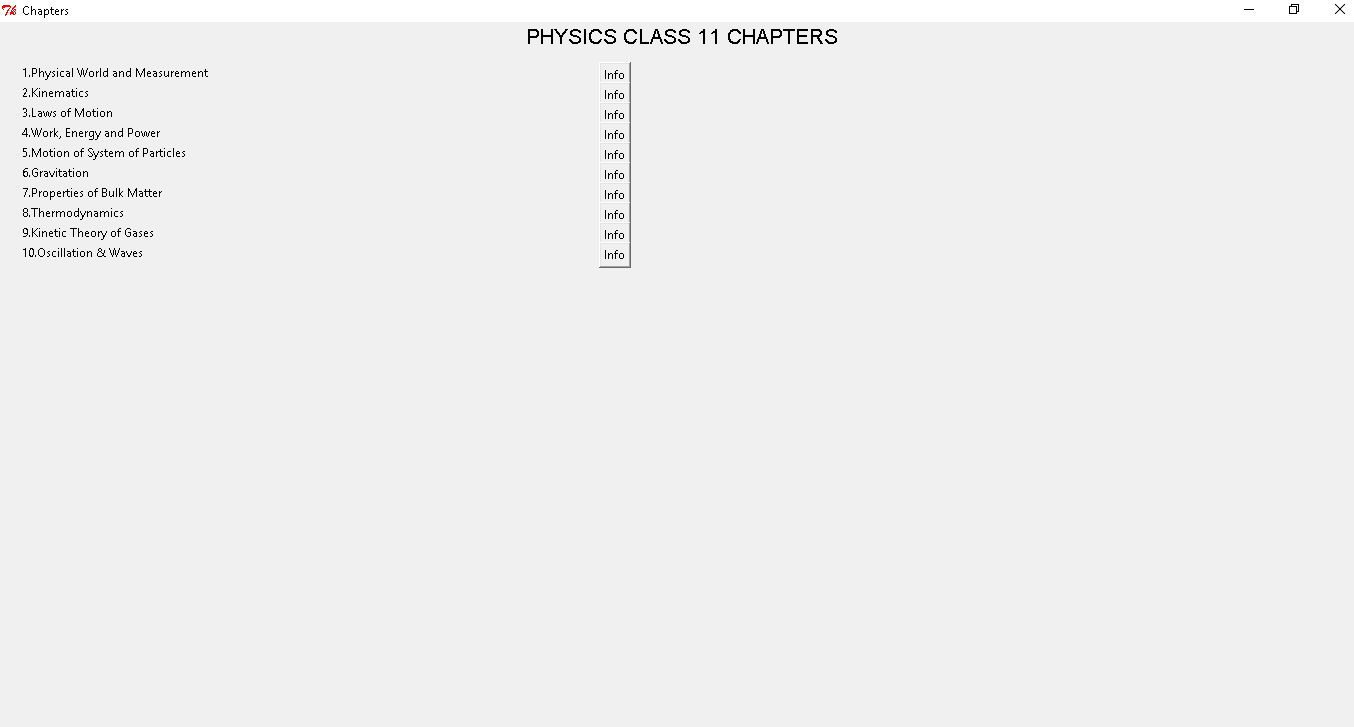
**Centre:**

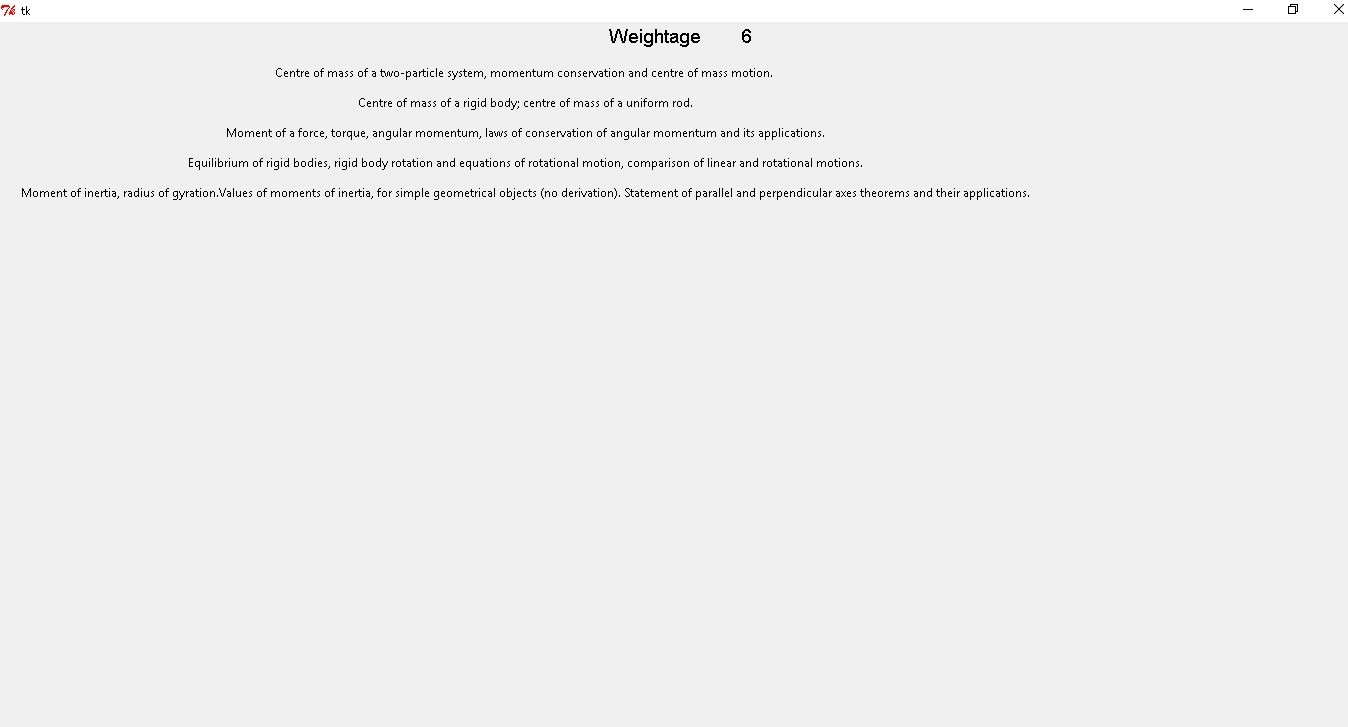
****











**Future Enhancements:**

We have tried our best to use all the concepts we have learned in our school and made a useful program for schools. We would make the program to have more interactions with the user and have graphical interpretation for making students like us to understand all the physics topics in a simpler manner.

Most importantly updating physics portal with the latest topics, connect our physics portal with the internet and make it worldwide online class. 😊

**Conclusions:**

This project has been successfully completed in 2 years by updating our program on a regular basis. Though the programming language we use is pretty easy it was hard to manage our school studies and the project. But with our team coordination we were able to succeed in our project. None of us in our team knew how to work with Tkinter. But with our determination of making this project a user interference and user friendly, we have put extra efforts in learning the new module and its applications. We have learned Tkinter from you tube and from stack overflow as well. To manage our school as well as our project we used to do the project together, divided our work and take the best out of every individual of our team and use it in our project.😊

**Bibliography:**

* <https://stackoverflow.com>
* You-Tube thenewboston
* Quora
* Python programmer