

INTRODUCTION TO PYTHON PROGRAMMING FOR MATHEMATICS

MAT351

**LAB RECORD**

**Name**: Jeevan Koshy

**Register Number**: 1740256

**Class**: 3CMS

In

[3]:

Your Number Please: 105

1

*#1*

*#Calculating the Factors of an Integer*

**def**

factors

(

b

):

**for**

i

**in**

range

(

1

,

b

**+**

1

):

**if**

b

**%**

i

**==**

0

:

print

(

i

)

b

**=**

input

(

'Your Number Please: '

)

b

**=**

float

(

b

)

**if**

b

**>**

0

**and**

b

.

is\_integer

():

factors

(

int

(

b

))

**else**

:

print

(

'Please enter a positive integer'

)

3

5

7

15

21

35

In

[4]:

105

Enter a: 20

Enter b: 8

*#2*

*#Quadratic Equation Root Calculator*

**def**

roots

(

a

,

b

,

c

):

D

**=**

(

b

**\***

b

**-**

4

**\***

a

**\***

c

)

**\*\***

0.5

x\_1

**=**

(

**-**

b

**+**

D

)

**/**

(

2

**\***

a

)

x\_2

**=**

(

**-**

b

**-**

D

)

**/**

(

2

**\***

a

)

print

(

'x1: {0}'

.

format

(

x\_1

))

print

(

'x2: {0}'

.

format

(

x\_2

))

a

**=**

input

(

'Enter a: '

)

b

**=**

input

(

'Enter b: '

)

c

**=**

input

(

'Enter c: '

)

roots

(

float

(

a

,

)

float

(

b

)

,

float

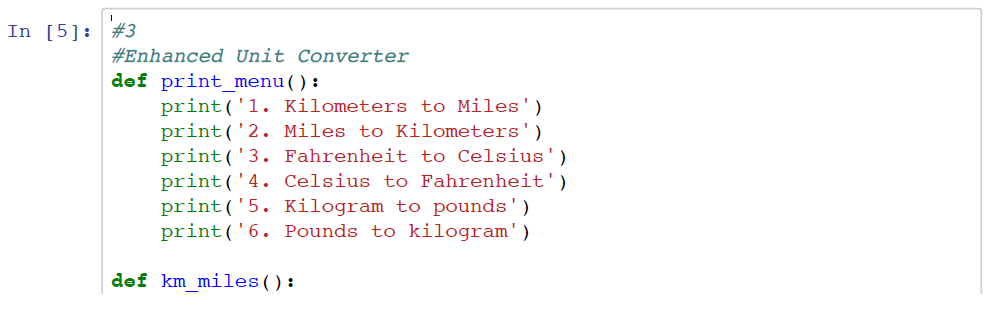
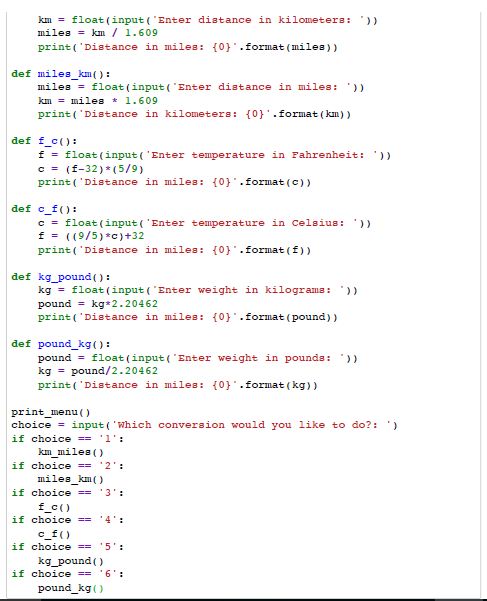
(

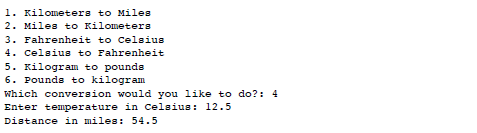
c

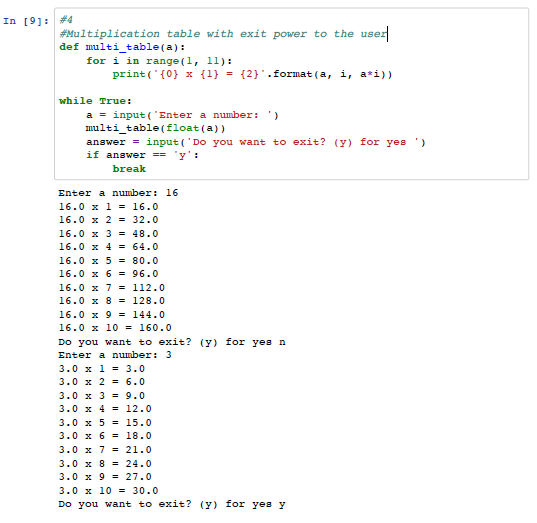
))

Enter c: 4 x1: (-0.19999999999999998+0.4j)

x2: (-0.20000000000000004-0.4j)

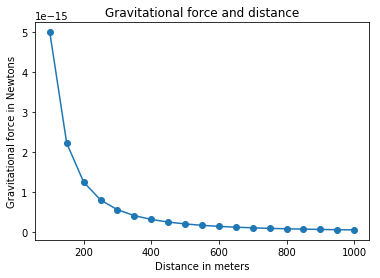






In

[7]:



*#5*

*#Gravitational force and Distance*

**import**

pip

**import**

matplotlib

.

pyplot

**as**

plt

*#draw graph*

**def**

draw\_graph

(

x

,

y

):

plt

.

plot

(

x

,

y

,

marker

**=**

'o'

)

plt

.

xlabel

(

'Distance in meters’

)

plt

.

ylabel

(

'Gravitational force in Newtons'

)

plt

.

title

(

'Gravitational force and distance'

)

plt

.

show

()

**def**

generate\_F\_r

():

*#Generate values for r*

r

**=**

range

(

100

,

1001

,

50

)

*#Empty list to store the calculated values*

F

**=**

[]

*#Constant,G*

G

**=**

6.674

**\***

(

10

**\*\*-**

11

)

*#Two masses*

m1

**=**

0.5

m2

**=**

1.5

*#Calculate force and add it to the list,F*

**for**

dist

**in**

r

:

force

**=**

G

**\***

(

m1

**\***

m2

)

**/**

(

dist

**\*\***

2

)

F

.

append

(

force

)

*#Call the draw graph function*

draw\_graph

(

r

,

F

)

generate\_F\_r

()