INTRODUCTION TO PYTHON PROGRAMMING FOR MATHEMATICS (MAT351): LAB RECORD

Name: Yana Kaveramma A.A

Register Number: 1740266

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)

In

[5]:

x2: (-0.20000000000000004-0.4j)

*#3*

*#Enhanced Unit Converter*

**def**

print\_menu

():

print

(

'1. Kilometers to Miles'

)

print

(

'2. Miles to Kilometers'

)

print

(

'3. Fahrenheit to Celsius'

)

print

(

'4. Celsius to Fahrenheit'

)

print

(

'5. Kilogram to pounds'

)

print

(

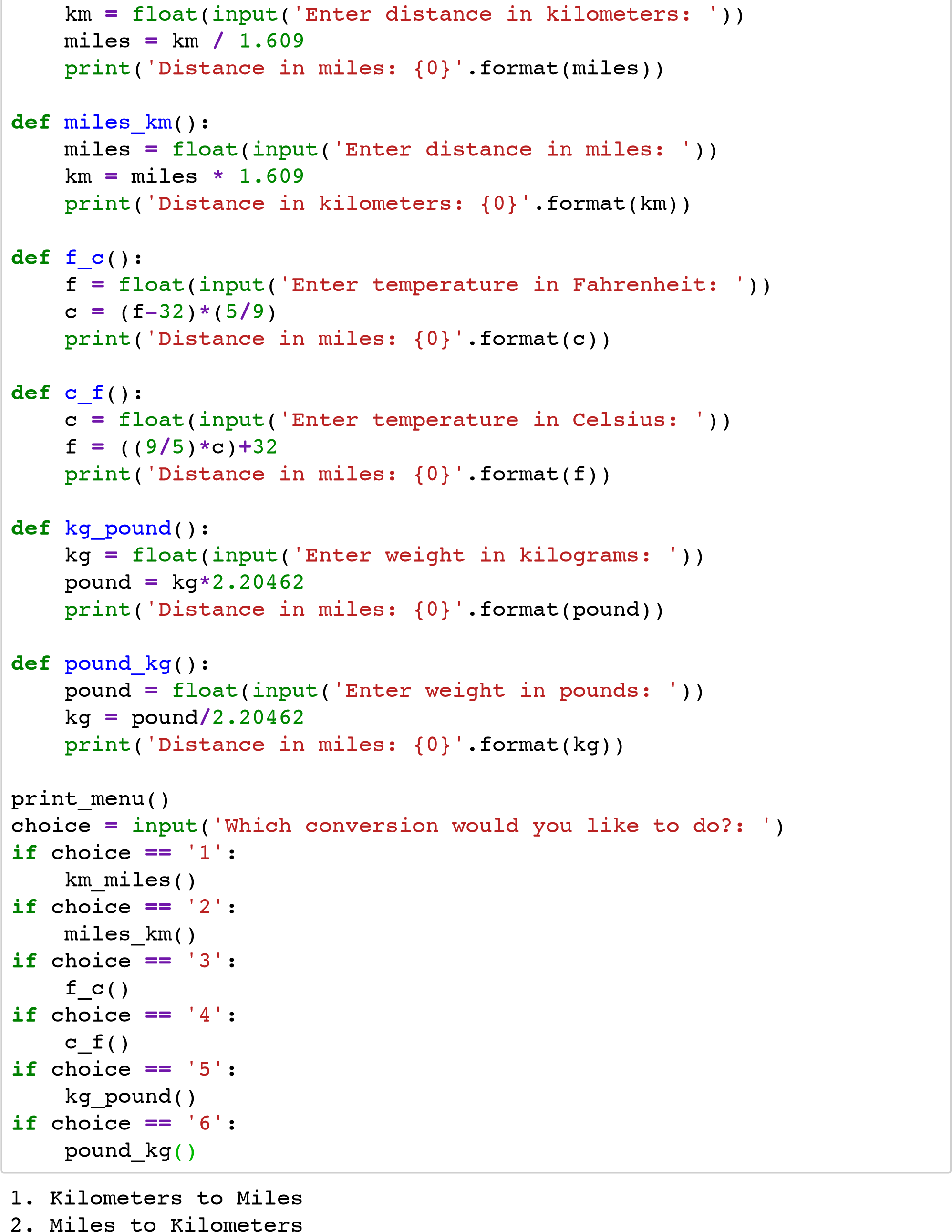
'6. Pounds to kilogram'

)

**def**

km\_miles

():



1. Fahrenheit to Celsius
2. Celsius to Fahrenheit
3. Kilogram to pounds
4. Pounds to kilogram

Which conversion would you like to do?: 4

Enter temperature in Celsius: 12.5

Distance in miles: 54.5

16.0 x 2 = 32.0 16.0 x 3 = 48.0 16.0 x 4 = 64.0 16.0 x 5 = 80.0

16.0 x 6 = 96.0

16.0 x 7 = 112.0 16.0 x 8 = 128.0

16.0 x 9 = 144.0

16.0 x 10 = 160.0

Do you want to exit? (y) for yes n

Enter a number: 3

In

[9]:

Enter a number: 16

x 1

= 16.0

16.0

*#4*

*#Multiplication table with exit power to the user*

**def**

multi\_table

(

a

):

**for**

i

**in**

range

(

1

,

11

):

print

(

'{0} x {1} = {2}'

.

format

(

a

,

i

,

a

**\***

i

))

**while**

**True**

:

a

**=**

input

(

'Enter a number: '

)

multi\_table

(

float

(

a

))

answer

**=**

input

(

'Do you want to exit? (y) for yes '

)

**if**

answer

**==**

'y'

:

**break**

3.0 x 1 = 3.0 3.0 x 2 = 6.0

3.0 x 3 = 9.0

3.0 x 4 = 12.0 3.0 x 5 = 15.0 3.0 x 6 = 18.0 3.0 x 7 = 21.0 3.0 x 8 = 24.0

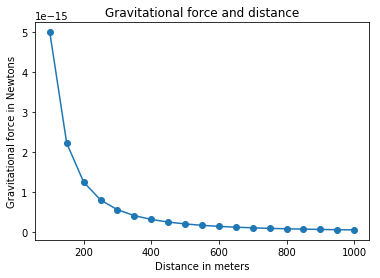
3.0 x 9 = 27.0

3.0 x 10 = 30.0

Do you want to exit? (y) for yes y

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

()

In

[10]:

Enter the number of categories:3

Enter category:Food

*#6*

*#Visualizing Your Expenses Using Bar Chart*

**import**

matplotlib

.

pyplot

**as**

plt

**def**

create\_bar\_chart

(

data

,

labels

):

num\_bars

**=**

len

(

data

)

positions

**=**

range

(

1

,

num\_bars

**+**

1

)

plt

.

barh

(

positions

,

data

,

align

**=**

'center'

)

plt

.

yticks

(

positions

,

labels

)

plt

.

xlabel

(

'Amount'

)

plt

.

ylabel

(

'Categories'

)

plt

.

title

(

'Weekly expenditures'

)

plt

.

grid

()

plt

.

show

()

n

**=**

int

(

input

(

'Enter the number of categories:'

))

labels

**=**

[]

expenditures

**=**

[]

**for**

i

**in**

range

(

n

):

category

**=**

input

(

'Enter category:'

)

expenditure

**=**

float

(

input

(

'Expenditure:'

))

labels

.

append

(

category

)

expenditures

.

append

(

expenditure

)

create\_bar\_chart

(

expenditures

,

labels

)

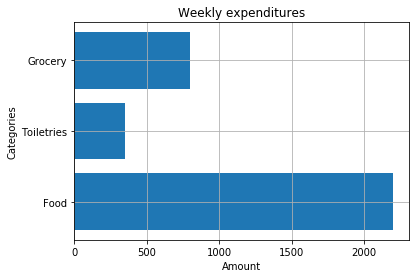
Expenditure:2200

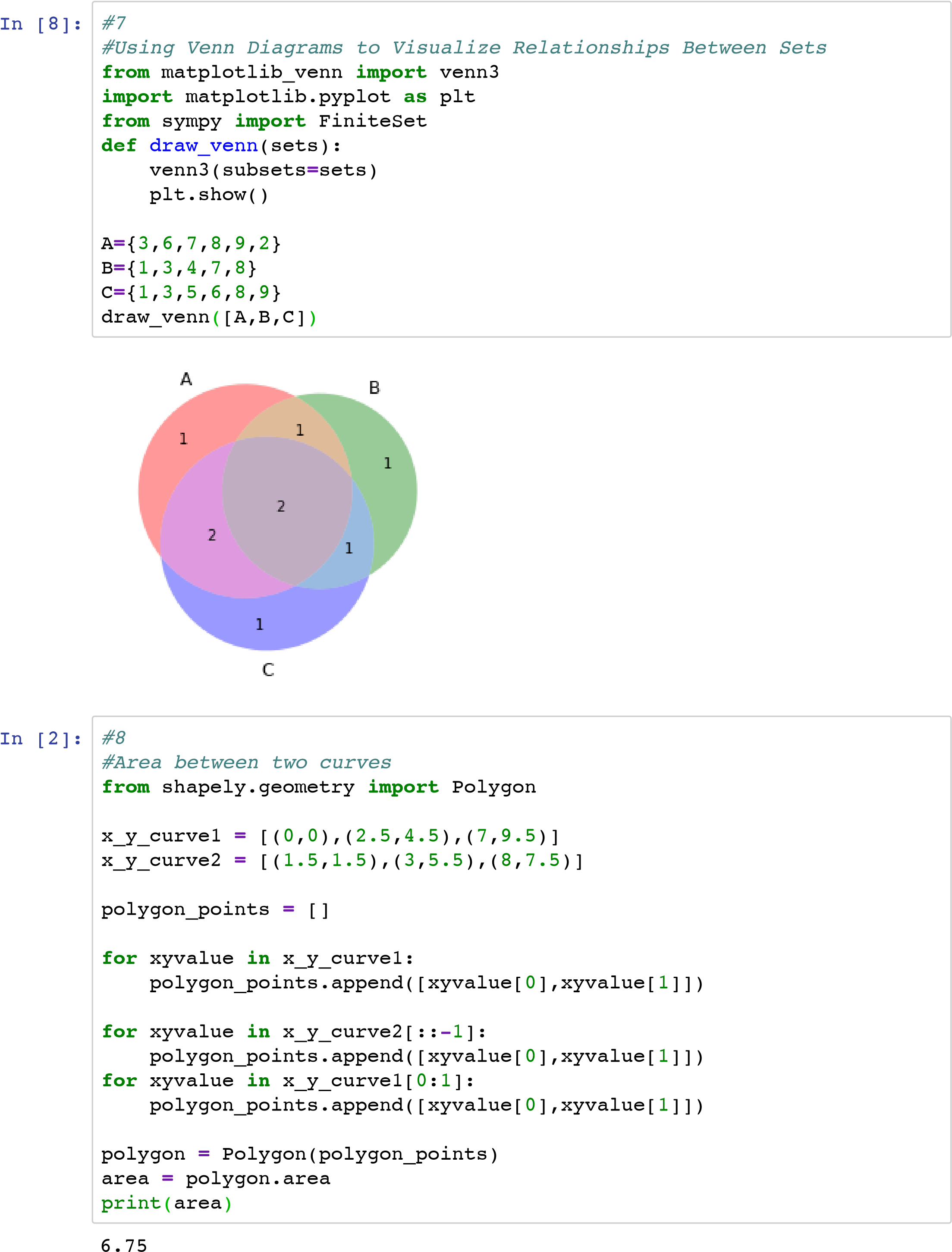
Enter category:Toiletries

Expenditure:350

Enter category:Grocery

Expenditure:800





In [ ]: