In [2]:

```
#Write a program that calculates and prints the value according to the given formula:

#Q= Square root of [(2*C*D)/H]

#Following are the fixed values of C and H:

#C is 50. H is 30.

#D is the variable whose values should be input to your program in a comma-separated sequence.

import math

C=50
H=30
D=input('Enter Value of D')
list_D=D.split(',')
print(list_D)

for i in range(len(list_D)):
    Q=math.sqrt((2*C*int(list_D[i])/H))
    print(Q)
```

```
Enter Value of D1,3
['1', '3']
1.8257418583505538
3.1622776601683795
```

In [3]:

```
###Define a class named Shape and its subclass Square. The Square class has an init function which takes a length as argument. Both classes have an area function which can print the area of the shape where Shape’s area is 0 by default.

class Shape:
    def area(self,len=0):
        res=len*len
        return res
class Square(Shape):
    def _init_(self,length):
      self.length=length
    def area(self,len):
        res = len*len
        return res
obj_S1 = Shape()
print(obj_S1.area())
obj_S2 = Square()

print(obj_S2.area(3))
```

```
0
9
```

In [4]:

```
###Create a class to find the three elements that sum to zero from a set of n real numbers.

###Input array: [-25,-10,-7,-3,2,4,8,10]

###Output: [[-10,2,8],[-7,-3,10]]

class ThreeElements:
    def find_zero(self,array):
        res_array=[]
        for i in array:
            for j in array:
                for k in array:
                    if i+j+k==0:
                        x1=sorted([i,j,k])
                        if x1 not in res_array:
                                res_array.append(x1)
        print (res_array)
        return res_array

test1=ThreeElements()
test1.find_zero([-25,-10,-7,-3,2,4,8,10])
```

```
[[-10, 2, 8], [-7, -3, 10]]
```

Out[4]:

```
[[-10, 2, 8], [-7, -3, 10]]
```

In [5]:

```
###Create a Time class and initialize it with hours and minutes.

###Make a method addTime which should take two time object and add them. E.g.- (2 hour and 50 min)
+(1 hr and 20 min) is (4 hr and 10 min)

###Make a method displayTime which should print the time.

###Make a method DisplayMinute which should display the total minutes in the Time. E.g.- (1 hr 2 m
in) should display 62 minute.

class  Time(object):


    def __init__(self, hours, minutes):
        self.hours = hours
        self.minutes = minutes

    def addTime(t1, t2):
        sum= Time(0, 0) # create a new object sum of class Time
        sum.hours = t1.hours + t2.hours # sum of hours
        sum.minutes = t1.minutes + t2.minutes # sum of minutes
        while sum.minutes >= 60:  #if the minutes are more than 60
            sum.hours= sum.hours +1
            sum.minutes = sum.minutes - 60
        return sum

    def displayTime(self):
        print("Time is %d hours and %d minutes" %(self.hours, self.minutes))

    def displayMinutes(self):
        print((self.hours * 60) + self.minutes, "minutes")

p = Time(2, 50)
q = Time(1, 20)
r = Time.addTime(p,q)

r.displayTime()
r.displayMinutes()
```

```
Time is 4 hours and 10 minutes
250 minutes
```

In [ ]:

```
### Write a Person class with an instance variable, , and a constructor that takes an integer, , a
s a parameter. The constructor must assign  to  after confirming the argument passed as  is not ne
gative; if a negative argument is passed as , the constructor should set  to  and print Age is not
valid, setting age to 0.. In addition, you must write the following instance methods:

yearPasses() should increase the  instance variable by .

amIOld()  should perform the following conditional actions:

If , print You are young..

If  and , print You are a teenager..

Otherwise, print You are old..
```

```
Sample Input:

4

-1

10

16

18

Sample Output:

Age is not valid, setting age to 0.

You are young.

You are young.


You are young.

You are a teenager.


You are a teenager.

You are old.


You are old.

You are old.
```

In [6]:

```
class person:
    def __init__(self,a=0):
        if a<0:
            self.a=0
            print('Age is not Valid, Setting age to 0')
        else:
            self.a=a
        self.amiOld()

    def amiOld(self):
        if self.a<18:
            print('You are young.')
        if self.a>10 and self.a<=20:
            print('You are a teenager.')
        else:
            print('You are old')

    def yearPasses(self):
        print('Year passed')

individual1=person(15)
```

```
You are young.
You are a teenager.
```

In [ ]: