Assignment T7

1. 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 a variable whose values should be input to your program in a comma-separated sequence.

```
import math

numbers = input("Provide D: ")
numbers = numbers.split(',')

result_list = []
for D in numbers:
   Q = round(math.sqrt(2 * 50 * int(D) / 30))
   result_list.append(Q)

print(result_list)
```

Output = Provide D: 5
[4]

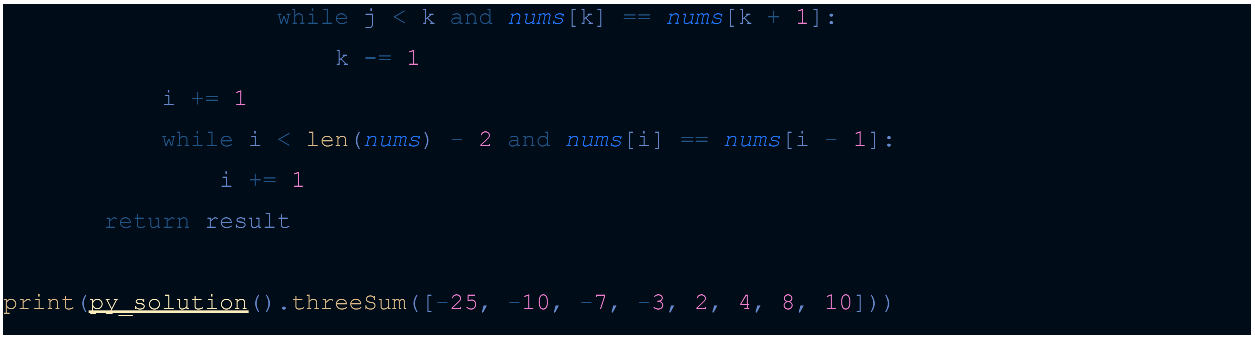
2. Define a class named Shape and its subclass Square. The Square class has an init function which
takes length as argument. Both classes have an area function which can print the area of the shape
where Shape's area is 0 by default.

```
class Shape():
   def __init__(self):
       pass

   def area(self):
       return 0

class Square(Shape):
   def __init__(self,length = 0):
       Shape.__init__(self)
       self.length = length

   def area(self):
       return self.length*self.length

Asqr = Square(5)
print(Asqr.area())

print(Square().area())
```

Output = 25
0

3. Create a class to find three elements that sum to zero from a set of n real numbers
Input array: [-25,-10,-7,-3,2,4,8,10]
Expected output: [[-10,2,8],[-7,-3,10]]

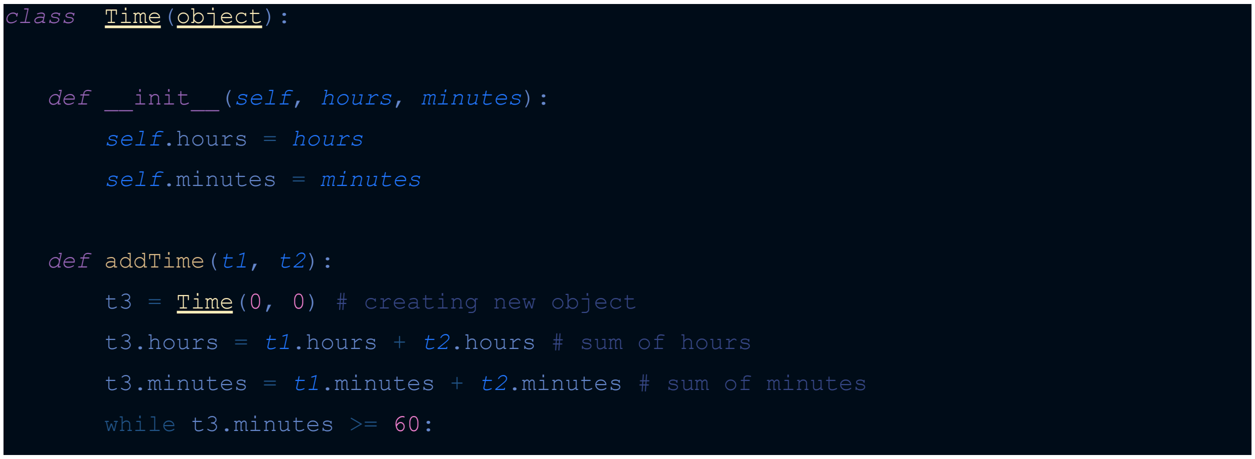
```
class py_solution:
def threeSum(self, nums):
       nums, result, i = sorted(nums), [], 0
       while i < len(nums) - 2:
           j, k = i + 1, len(nums) - 1
           while j < k:
               if nums[i] + nums[j] + nums[k] < 0:
                   j += 1
               elif nums[i] + nums[j] + nums[k] > 0:
                   k -= 1
               else:
                   result.append([nums[i], nums[j], nums[k]])
                   j, k = j + 1, k - 1
                   while j < k and nums[j] == nums[j - 1]:
                       j += 1
```



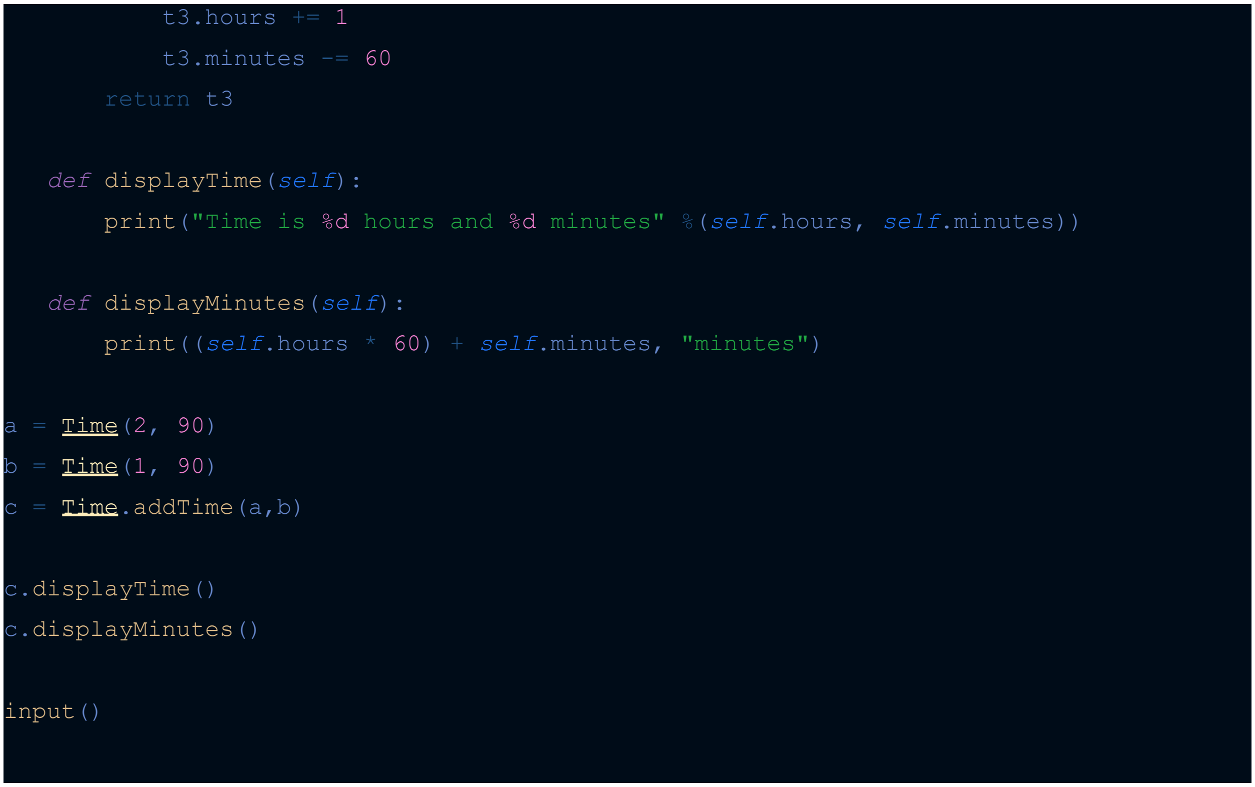
```
                    while j < k and nums[k] == nums[k + 1]:
                        k -= 1
            i += 1
            while i < len(nums) - 2 and nums[i] == nums[i - 1]:
                i += 1
        return result

print(py_solution().threeSum([-25, -10, -7, -3, 2, 4, 8, 10]))
```

Output = [[-10, 2, 8], [-7, -3, 10]]

4. Create a Time class and initialize it with hours and minutes.
Create a method addTime which should take two Time objects and add them.
E.g.- (2 hour and 50 min)+(1 hr and 20 min) is (4 hr and 10 min)
Create another method displayTime which should print the time.
Also create a method displayMinute which should display the total minutes in the Time.
E.g.- (1 hr 2 min) should display 62 minute.
yearPasses() should increase age by the integer value that you are passing inside the function.
amIOld() should perform the following conditional actions:I
if the age is between 0 and <13, print "You are young".
If age is >=13 and <=19 , print "You are a teenager".
Otherwise, print "You are old".



```
class  Time(object):

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

    def addTime(t1, t2):
        t3 = Time(0, 0) # creating new object
        t3.hours = t1.hours + t2.hours # sum of hours
        t3.minutes = t1.minutes + t2.minutes # sum of minutes
        while t3.minutes >= 60:
```



```
            t3.hours += 1
            t3.minutes -= 60
        return t3

    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")

a = Time(2, 90)
b = Time(1, 90)
c = Time.addTime(a,b)

c.displayTime()
c.displayMinutes()

input()
```

Output =
Time is 6 hours and 0 minutes
360 minutes

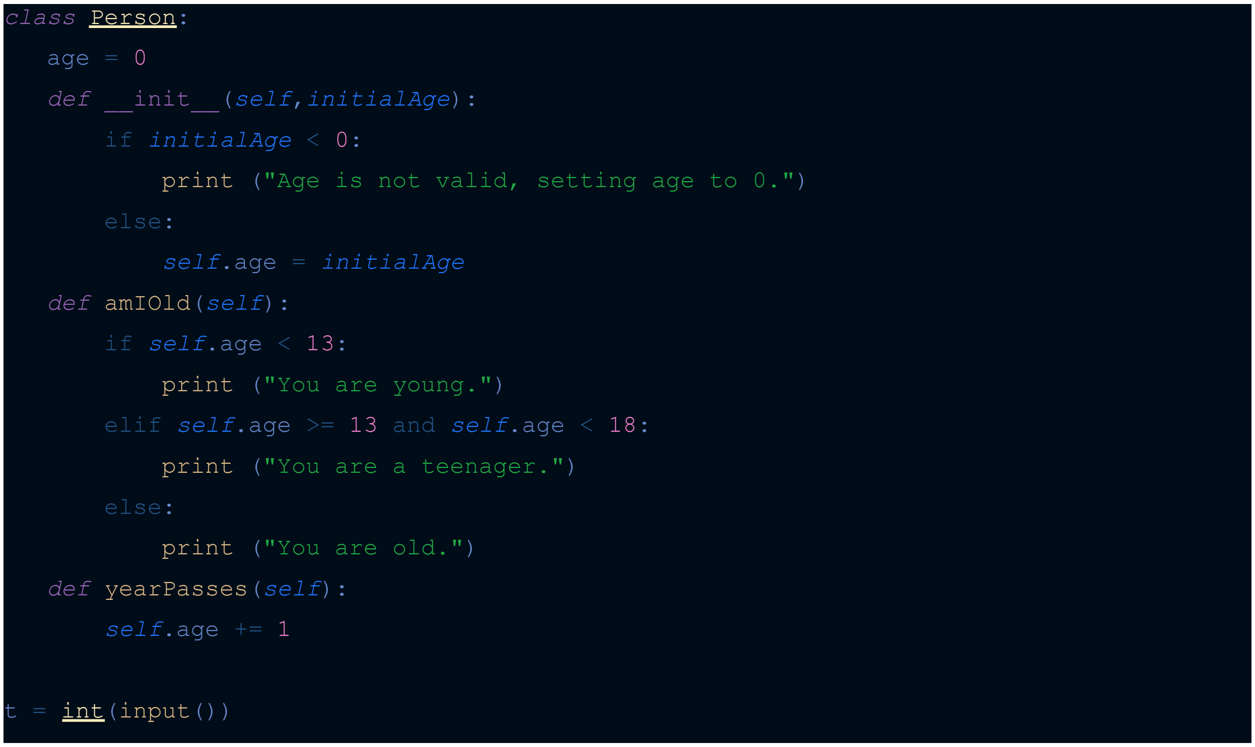
5. Write a Person class with an instance variable “age” and a constructor that takes an integer as a
parameter. The constructor must assign the integer value to the age variable after confirming the
argument passed is not negative; if a negative argument is passed then the constructor should set
age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

Sample Input for amIOld():

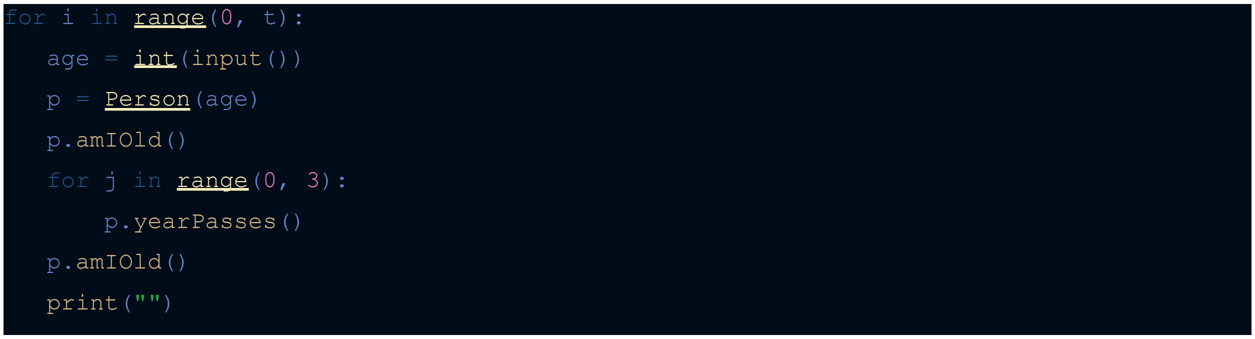
-1
4
10
16
18
64
38

Expected Output for amIOld():
Age is not valid, setting age to 0.
You are young.
You are young.
You are a teenager.
You are a teenager.
You are old.
You are old.

Consider the age variable to be set to 38 then:
Sample Input for yearPasses(): 4
Expected Output for yearPasses(): 42



```
class Person:
   age = 0
   def __init__(self,initialAge):
       if initialAge < 0:
           print ("Age is not valid, setting age to 0.")
       else:
           self.age = initialAge
   def amIOld(self):
       if self.age < 13:
           print ("You are young.")
       elif self.age >= 13 and self.age < 18:
           print ("You are a teenager.")
       else:
           print ("You are old.")
   def yearPasses(self):
       self.age += 1

t = int(input())
```



```python
for i in range(0, t):
   age = int(input())
   p = Person(age)
   p.amIOld()
   for j in range(0, 3):
       p.yearPasses()
   p.amIOld()
   print("")
```