Q30. Write A Program To Calculate Sum Of Non Negative Numbers In A List Of 10 Numbers.

```
# Program To Calculate Sum Of Non Negative Numbers In A List Of 10 Numbers.

def create_list(ls):
    total = 0
    count = 0
    while True:
        x = int(input())
        ls.append(x)
        if x >= 0:
            total += x

        count += 1
        if count == 10:
            break

    print("Sum Values: ", total)

print("Enter Values Into The List")
ls = []
create_list(ls)
```

****
****
***

Q31. Write A Program That Accepts A List, Seprate The List With Comma And Add "and" Before The Last Item In The List

```
# Program Accepts A List, Seprate The List With Comma And Add "and" Before The Last Item In The List

def acceptlist(spamlist):
    returnstring = ''

    for i in range (len(spamlist)):
        if i == len(spamlist)-1:
            returnstring += 'and ' + spamlist[i]
        else:
            returnstring += spamlist[i] + ', '

    return returnstring

spam= ['Apples', 'Banana', 'Tofu', 'Cats']
result = acceptlist(spam)
print(result)
```

****
****
***

Q32. Write A Python Program to Count Words In A String Using Dictionary

```
#Python Program to Count Words In A String Using Dictionary

string = input("Please Enter The String : ")
words = []

words = string.split() # or string.lower().split()
myDict = {}
for key in words:
   myDict[key] = words.count(key)

print("Dictionary Items  :  ",  myDict)
```

****

Q33. Write A Program To Count The Number Of Vowels In A Given String

```
# Program To Count The Number Of Vowels In A Given String
string=input("Enter The String:")
vowels=0
for i in string:
     if(i=='a' or i=='e' or i=='i' or i=='o' or i=='u' or i=='A' or i=='E' or i=='I' or i=='O' or i=='U'):
          vowels=vowels+1
print("Number Of Vowels Present Are: ", vowels)
```

****

Q34. Write A Program To Check Whether A String Is Palindrome Or Not

```
#  Program To Check Whether A String Is Palindrome Or Not

string = input(("Enter The Input String: "))

if string == string[::-1]:
     print("The String Is Palindrome In Nature")
else:
     print("The String Is Not Palindrome In Nature")
```

****

Q35. Write A Program To Check The Number Of Times A Character Appears In A String

```
# Program To Count The Number Of Vowels In A Given String

def check(string, ch):
    count = 0
    for i in string:
        if i == ch:
            count += 1
    print("Number Of Occurrences Of ", ch, "Is/Are ", count)

string=input("Enter The String:")
ch = input("Enter Your Desired Character: ")

check(string,ch)
```

****
****
***

Q36. Write A Program To Check Whether Two Strings Are Anagram Or Not.

```
# A Program To Check Whether Two Strings Are Anagram Or Not

def anagramcheck(str1, str2):
    list_str1 = list(str1)
    list_str1.sort()
    list_str2 = list(str2)
    list_str2.sort()

    return (list_str1 == list_str2)

str1 = input("Enter The First String: ")
str2 = input("Enter The Second String: ")
print(anagramcheck(str1, str2))
```

****
****
***

Q37. Write A Program To Remove All Duplicate Characters From A String And Print The String WIth Unique Characters

```
#  Program To Remove All Duplicate Characters From A String And Print The String WIth Unique Characters

ls = {}
string = input("Enter Your String: ")
```

```
res = []

for c in string:
   if c not in ls:
     res.append(c)
     ls[c]=1

print("The Resultant String Is: ",end=" ")
print("".join(res))
```

****
****
***

Q38. Write A Program To Perform The Following Operations On String HelloWorld

> s[5: -2] > s[-4:] > s[-8:-2] > s[-1:] > s[:-1] > s[: :-1]) > s[: :-2])

```
# Program To Perform The Operations Mentioned Above

s = "Hello World"

print("Result 1: ", s[5: -2])

print("Result 2: ", s[-4: ])

print("Result 3: ", s[-8:-2])

print("Result 4: ", s[-1: ])

print("Result 5: ", s[ :-1])

print("Result 6: ", s[ : :-1])

print("Result 7: ", s[ : :-2])
```

****
****
***

Q39. Write A Program To Perform Various Slicing Operations On Given List ls = [0,11,22,33,44,55,66,77,88,99]

```
ls = [0, 11, 22, 33, 44, 55, 66, 77, 88, 99]

print("Result 1: ", ls[0:7:2])

print("Result 2: ", ls[2:7:1])
print("Result 3: ", ls[7:2:1])

print("Result 4: ", ls[7:2:-1])

print("Result 5: ", ls[:7:1])

print("Result 6: ", ls[5::1])

print("Result 7: ", ls[:-5:-1])

print("Result 8: ", ls[-5::-1])

print("Result 9: ", ls[5:-1:1])

print("Result 10: ", ls[-1:5:-1])

print("Result 11: ", ls [::-1])

print("Result 12: ", ls[::2])

print("Result 13: ", ls[::-2])

print("Result 14: ", ls[3:-2:-2])

print("Result 15: ", ls[::])
```

****
****
***

Q40. Write a Python Program that reads ‘n’ elements from the user and creates a list, and then display the same.

```
def create_list(ls, n):

    while True:
        x = input()
        ls.append(x)
        if len(ls) == n:
            break


    print("List: ", ls)

n = int(input("Enter The Number Of Elements: "))
ls = []
create_list(ls, n)
```

****
****
***

Q41. Write a Python Program that finds the sum of all even numbers and odd numbers in a predefined list

```
# Assuming Predefined List To Be Set Of Integers From 1 to 10

ls = [1,2,3,4,5,6,7,8,9,10]
odd_sum = 0
even_sum = 0

for x in ls:
    if x % 2 != 0:
        odd_sum += x
    else:
        even_sum += x

print("Even Number Summation Equals ", even_sum)
print("Odd Numbers Summation Equals ", odd_sum)
```

****
****
***

Q42: Write a Python Program that creates a list of 10 integers. Then create two lists-Odd List and Even List that has all odd and even values in the list respectively

```
# Even_Odd List Program
def create_list(ls):

    print("Enter 10 Values Into The List")
    while True:
        x = int(input())
        ls.append(x)
 if len(ls) == 10:
            break

    return ls

def even_odd_lists(ls):
    ls_odd = []
    ls_even = []

    for i in ls:
        if i % 2 != 0:
            ls_odd.append(i)
        else:
            ls_even.append(i)

    print("Odd_List: ", ls_odd)
    print("Even_List; ", ls_even)

ls = []
ls = create_list(ls)
```

```
even_odd_lists(ls)
```

****
****
***

Q43: Write a Python Program that creates a list of 20 numbers and then create a list that contains all the numbers from the original list that are divisible by 3

```
# DivisibleBy3 List Numbers

def create_list(ls):

    print("Enter 20 Values Into The List")
    while True:
        x = int(input())
        ls.append(x)
        if len(ls) == 20:
            break

    return ls

def mod_three_lists(ls):
    ls_mod3 =[]

    for i in ls:
        if i % 3 == 0:
            ls_mod3.append(i)

    print("Divisible By Three List: ", ls_mod3)

ls = []
ls = create_list(ls)
mod_three_lists(ls)
```

****
****
***

Q44: Write a Python Program that counts the number of times a value appears in the list without using built in function

```
# Value Count
def create_list(ls):
    # Press Anything Other Than Enter To Stop Accepting The Input

    while True:
        x = input()

        if x == "":
            break

        ls.append(x)

    return ls

def count_value(ls, se):
    count = 0
    for x in ls:
        if x == se:
            count += 1

    print(se, "Appeared For ", count, " Times In ", ls)


ls = []
ls = create_list(ls)

se = input("Enter The Search Element: ")
count_value(ls, se)
```

****
****
***

Q45: Write a Python program to illustrate operations of queues using list

```
# Implement Queue Using List
def queue_list(queue):
    while True:
        print("Enter 1: Add Element To The Queue")
        print("Enter 2: Remove Element From The Queue")
        print("Enter 3: Print The Queue")
        print("Enter 4: Exit")
        x = int(input("Enter Your Choice: "))

        if x == 1:
            val = int(input("Enter The Element To Be Added Into The Queue: "))
            queue.append(val)
            print(val, " Inserted Successfully")

        elif x == 2:
            if len(queue) == 0:
                print("Queue Is Empty. Nothing To Be Removed")
            else:
                print(queue.pop(0), " Removed Successfully")

        elif x == 3:
            if len(queue) == 0:
                print("Queue Is Empty")
            else:
                print("Queue: ", queue)

        elif x== 4:
            print('Thank You')
            exit()

 else:
     print("Invalid Input Entered")

queue=[]
queue_list(queue)
```

****
****
***

Q46. Consider The String 'brontosaurus'. Write A Python Code That Implements And Returns The Functionality Of A Histogram Using Dicitonary For The Given String.

```
word = 'brontosaurus'
d = dict()
for c in word:
    if c not in d:
        d[c] = 1
    else:
        d[c] = d[c] + 1
print(d)
```

****
****
***

Q47. Write a Python program to get the maximum and minimum value in a dictionary.

```
my_dict = {'a': 1, 'b': 2, 'c': 3, 'd': 4, 'e': 5}

dict_min = min(my_dict.values())
dict_max = max(my_dict.values())

print("In Dictionary", my_dict, ", Max Value Is %d And Min Value Is %d" % (dict_max, dict_min))
```

****
****
***

Q48. Write A Program To Find Top 3 Items In A Shop.

```
# Program To Find Top 3 Items In Shop.

from collections import Counter
my_dict = {'Item1': 45.50, 'Item2': 25, 'Item3': 41.3, 'Item4': 55, 'Item5': 24}

k = Counter(my_dict)

# Finding 3 highest values
high = k.most_common(3)

print("The Top 3 Items In The Shop Are: ")
print("Keys: Values")

for i in high:
  print(i[0]," :",i[1]," ")
```

****
****
***

Q49. Write A Program To Accept The Marks And Names Of 4 Students Using Dictionary. Using This Dictionary, Create Another Dictionary Having The Names Of The Students And The Total Marks
Scored By Them. Find Out The Topper And Print His/Her Name And Total Score.

```
# Program To Find Out The Topper

dict = {'Tom': [70, 80, 86, 44], 'Harry': [65, 77, 45, 88], 'Alice': [88,99,76,89], 'Kevin': [99, 66, 76, 81]}
print("Original Dictionary: ", dict)

sum_dict = {k: sum(v) for k, v in dict.items()}
print("Dictionary Having Values Representing Sum Of Scores Secured By A Student: ",sum_dict)

print("Topper Score Is: ", max(sum_dict.values()))

v = list(sum_dict.values())
k = list(sum_dict.keys())
topper_index = k[v.index(max(v))]

print("Topper Name: ", topper_index)
```

****
****
***

Q50. Write A Python Program That Accepts A Sentence And Build Dictionary With Letter, Digits, Uppercase, Lowercase As Key And Their Count In Sentence As Values.

```
# Python Program That Accepts A Sentence And Build Dictionary With Letter, Digits, Uppercase, Lowercase As Key And Their Count In Sentence As Values.
```

```
string = input("Enter Your String: ")

letter_count = 0
digit_count = 0
uppercase_count = 0
lowercase_count = 0

for i in range(len(string)):
   if string[i].isdigit():
      digit_count += 1
   elif string[i].isupper():
      uppercase_count += 1
   elif string[i].islower():
      lowercase_count += 1

letter_count = lowercase_count + uppercase_count

dictionary = {'Letter': letter_count, 'Digit': digit_count, 'UpperCase': uppercase_count, 'Lowercase': lowercase_count
}
print(dictionary)
```

```
***************************************************************************************************************************************************************************************************
```

Q51. Write A Python Program That Accepts A Sentence And Count The Number Of Letters, Digits, Uppercase Characters, Lowercase Characters And Spaces In It.

```
# Program That Accepts A Sentence And Count The Number Of Letters, Digits, Uppercase Characters, Lowercase Characters And Spaces In It


string = input("Enter Your String: ")

letter_count = 0
digit_count = 0
uppercase_count = 0
lowercase_count = 0
space_count = 0

for i in range(len(string)):
   if string[i].isdigit():
      digit_count += 1
   elif string[i].isupper():
      uppercase_count += 1
   elif string[i].islower():
      lowercase_count += 1
   elif string[i].isspace():
      space_count += 1

letter_count = lowercase_count + uppercase_count
```

```
print("Digit Count: ", digit_count)
print("Upper Case Count: ", uppercase_count)
print("Lower Case Count: ", lowercase_count)
print("Space Count: ", space_count)
print("Letter Count: ", letter_count)
```

****
****
***

Q52. Write A Program To Swap The Cases In A Given Statement

```
# Program To Swap The Cases In A Given Statement

def swap_case(s):
    result = ""
    for letter in s:
        if letter == letter.upper():
            result += letter.lower()
        else:
            result += letter.upper()
    return result


string = input("Enter The String: ")
print(swap_case(string))
```

****
****
***

Q53. Write A Python Program That Performs A Validty Check Upon The Password Entered By The User.
    >: At least One Upper Case Character Should Be Present
    >: At least One Lower Case Character Should Be Present
    >: At least One Digit Should Be Present
    >: At least One Character Among '$', '#' And '@' Should Be Present
    >: Password Length Should Be Between 6 To 16 Characters In Total

```
# Python Program That Performs A Validty Check Upon The Password Entered By The User

string = input("Enter Your Password To Check For It's Validity: ")

letter_count = 0
digit_count = 0
```

```
uppercase_count = 0
lowercase_count = 0
special_count = 0
string_length = 0

for i in range(len(string)):
    if string[i].isdigit():
        digit_count += 1
    elif string[i].isupper():
        uppercase_count += 1
    elif string[i].islower():
        lowercase_count += 1
    elif string[i] == '$' or string[i] == '#' or string [i] == '@':
        special_count += 1

string_length = len(string)
letter_count = lowercase_count + uppercase_count


if uppercase_count >= 1 and lowercase_count >= 1 and digit_count >= 1 and special_count >= 1 and (string_length
>=6 and string_length <= 16):
    print("Password Satisfies The Following Constraints On Validity Check: ")
    print("1: At least One Upper Case Character Should Be Present")
    print("2: At least One Lower Case Character Should Be Present")
    print("3: At least One Digit Should Be Present")
    print("4: At least One Character Among '$', '#' And '@' Should Be Present")
    print("5: Password Length Should Be Between 6 To 16 Characters In Total")
else:
    print("Password Does Not Satisfy The Following Constraints On Validity Check: ")
    print("1: At least One Upper Case Character Should Be Present")
    print("2: At least One Lower Case Character Should Be Present")
    print("3: At least One Digit Should Be Present")
    print("4: At least One Character Among '$', '#' And '@' Should Be Present")
    print("5: Password Length Should Be Between 6 To 16 Characters In Total")
```

****
****
***

Q54. Write A Python Program That Checks The Presence Of A Substring In A Given String

```
# Program That Checks The Presence Of A Substring In A Given String

InputString = input("Enter Your String: ")
InputSubString = input("Enter The SubString Whose Presence Is To Be Found Out: ")

if InputSubString in InputString:
    print("{} Is Present In {}".format(InputSubString, InputString))
else:
    print("{} Is Not Present In {}".format(InputSubString, InputString))
```

****
****
***

Q55. Write A Program To Print The Longest Word And It's Length In A Given String

```
# Program To Print The Longest Word And It's Length In A Given String

String = input("Enter The String: ")
Word = ''
MaxLen = 0
MaxWord = ''

for i in String + ' ':
    if i == ' ':
        if len(Word) > MaxLen:
            MaxWord = Word
        Word = ''
    else:
        Word += i


print("Longest Word: ", MaxWord)
print("Length: ", len(MaxWord))
```

****
****
***

Q56. Write A Program To Get A String From A Given String Where All Occurences Of It's First Character Have Been Changed To '$' Except The First Character Itself.

```
# Program To Get A String From A Given String Where All Occurences Of It's First Character Have Been Changed
To '$' Except The First Character Itself.

String = input("Enter The String: ")

NewString = FirstCharacter = String[0]

print("Original String: ", String)

for i in range(1, len(String)):
    if String[i] == FirstCharacter:
        NewString += '$'
    else:
        NewString += String[i]


print("Result String: ", NewString)
```

****
****
***

Q57. Write A Program That Adds 'ing' At The End Of The String If It Doesn't End With 'ing' Or Add 'ly' If It Ends With 'ing'. Only Do The Changes If The String Length Is Greater Than 3.

```
#Program That Adds 'ing' At The End Of The String If It Doesn't End With 'ing' Or Add 'ly' If It Ends With 'ing'. Only Do The Changes If The String Length Is Greater Than 3.

String = input("Enter The String: ")
NewString = ''

Last3Characters = String[(len(String)-3): (len(String)+1)]
print("Last 3 Characters Of Given String: ", Last3Characters)

if Last3Characters == 'ing' and len(String) >= 3:
    NewString = String + 'ly'
elif Last3Characters != 'ing' and len(String) >= 3:
    NewString = String + 'ing'
elif len(String) < 3:
    NewString = String


print("Original String: ", String)
print("New String: ", NewString)
```

****
****
***

Q58. Write A Program To Reverse The Words In A Given String.

```
# Program To Reverse The Words In A Given String
InputString = input("Enter The Input String: ")
Word = InputString.split(' ')
Word = list(reversed(Word))
print("Output String: ", " ".join(Word))
```

****
****
***

Q59. Write A Program To Illustrate Caeser Encryption

```
# Program To Illustrate Caeser Encryption
```

```
def encrypt(text, s):
    result = ""
    for i in range(len(text)):
        char = text[i]
        if char.isupper():
            result += chr((ord(char) + s - 65) % 26 + 65)
        else:
            result += chr((ord(char) + s - 97) % 26 + 97)

    return result


text = input("Enter The Input String: ")
s = int(input("Enter The Shift Value: "))

print("Plain Text: " + text)
print("Shift Value : " + str(s))
print("Cipher: " + encrypt(text, s))
```

****
****
***

Q60.Write a Python program to display a number in left, right and center aligned of width 10.

```
# A Python program to display a number in left, right and center aligned of width 10.

x = int(input("Enter A Number: "))
print("Left Aligned (Width 10): "+" {:< 10d}".format(x))
print("Right Aligned (Width 10): "+" {:10d}".format(x))
print("Center Aligned (Width 10): "+" {:^10d}".format(x))
```

****
****
***

Q61. Given A Point P(X, Y), WAP To Check In Which Quadrant It Lies In.

```
# Program To Check The Quadrant In Which The Co-Ordinate Lies

def coordinate(x,y):
    if x >= 0 and y >= 0:
        print("({},{}) Lies In First Quadrant".format(x,y))
    elif x <= 0 and y >= 0:
        print("({},{}) Lies In Second Quadrant".format(x, y))
    elif x <= 0 and y <= 0:
        print("({},{}) Lies In Third Quadrant".format(x, y))
    else:
        print("({},{}) Lies In Fourth Quadrant".format(x, y))


x_pos = int (input("Enter The X Co-Ordinate (0-90): "))
y_pos = int (input("Enter The Y Co-Ordinate (0-90): "))

coordinate(x_pos,y_pos)
```

****
****
***

Q62.Write A Program To Check If The 3 Points In A Graph Are Collinear In Nature

```
# Program To Check If The 3 Points Are Collinear In Nature.

X1 = float(input("Enter Value Of X1: "))
Y1 = float(input("Enter Value Of Y1: "))
X2 = float(input("Enter Value Of X2: "))
Y2 = float(input("Enter Value Of Y2: "))
X3 = float(input("Enter Value Of X3: "))
Y3 = float(input("Enter Value Of Y3: "))

Area = 0.5 * (X1 * (Y2 - Y3) + X2 * (Y3 - Y1) + X3 * (Y1 - Y2))

if Area == 0.0:
    print("The Given Points Are Collinear In Nature")
else:
    print("The Given Points Aren't Collinear In Nature")
```

****
****
***

Q63. Write A Program To Calculate Arc Length From A Given Value Of Angle.

```
# Program To Calculate Arc Length From A Given Value Of Angle.

def arcLength (diameter, angle):
    if angle >= 360:
        print("Angle Cannot Be Formed. Hence, No Value Of Arc Length Can Be Calculated.")
        exit(0)
    else:
        arc = (3.142857142857143 * diameter) * (angle / 360.0)
        return arc

diameter = float(input("Enter The Value Of Diameter: "))
angle = float(input("Enter The Value Of Angle: "))
arc_len = arcLength(diameter, angle)
```

```
print("Arc Length: {} Units".format(arc_len))
```

****
****
***

Q64. Write A Function Named move_rectangle That Takes A Rectangle And Two Numbers Named dx And dy. It S hould Change The Location Of Rectangle By Adding dx To x Co-Ordinate And dy To y
Co-Ordinate.

```
# Program To Change The Postion Of A Rectangle By dx And dy Parameters

class Point(object):
    "Represents A Point In 2-D Space"

class Rectangle(object):
    """Represents A Rectangle.
    Attributes: Width, Height, Corner"""

def move_rectangle(rect, dx, dy):
    rect.corner.x += dx
    rect.corner.y += dy


box = Rectangle()
box.width = input("Enter The Width Of The Rectangle: ")
box.height = input("Enter The Height Of The Rectangle: ")
box.corner = Point()
box.corner.x = 0.0
box.corner.y = 0.0

print("Dimensions Of Rectangle - Width = {} Units".format(box.width))
print("Dimensions Of Rectangle - Height = {} Units".format(box.height))
print("Initial Value Of X Co-Ordinate Representing Bottom Left Corner Of The Rectangle: ", box.corner.x)
print("Initial Value Of Y Co-Ordinate Representing Bottom Left Corner Of The Rectangle: ", box.corner.y)


dx = float(input("Enter The Value Of dx By Which You Wish To Move The Position Of Rectangle: "))
dy = float(input("Enter The Value Of dy By Which You Wish To Move The Position Of Rectangle: "))


move_rectangle(box, dx, dy)


print("Updated Value Of X Co-Ordinate Representing Bottom Left Corner Of The Rectangle: ", box.corner.x)
print("Updated Value Of Y Co-Ordinate Representing Bottom Left Corner Of The Rectangle: ", box.corner.y)
```

****

Q65. Write A Program To Perform Basic Operations Such As Addition, Multiplication And Subtraction On Complex Numbers Using Concept Of Classes.

```
# Program To Perform Basic Operations On Complex Numbers

class Complex():

    def initComplex(self):
        self.realPart = int(input("Enter the Real Part: "))
        self.imgPart = int(input("Enter the Imaginary Part: "))

    def display(self):
        print(self.realPart,"+",self.imgPart,"i", sep="")

    def sum(self, c1, c2):
        self.realPart = c1.realPart + c2.realPart
        self.imgPart = c1.imgPart + c2.imgPart

    def diff(self, c1, c2):
        self.realPart = c1.realPart - c2.realPart
        self.imgPart = c1.imgPart - c2.imgPart

    def mul(self, c1, c2):
        self.realPart = c1.realPart * c2.realPart
        self.imgPart = c1.imgPart * c2.imgPart


c1 = Complex()
c2 = Complex()
c3 = Complex()
c4 = Complex()
c5 = Complex()

print("Enter First Complex Number")
c1.initComplex()
print("First Complex Number: ", end="")
c1.display()

print("Enter Second Complex Number")
c2.initComplex()
print("Second Complex Number: ", end="")
c2.display()

print("Sum Of Two Complex Numbers Is: ", end="")
c3.sum(c1, c2)
c3.display()

print("Difference Of Two Complex Numbers Is: ", end="")
c4.diff(c1, c2)
```

```
c4.display()

print("Multiplication Of Two Complex Numbers Is: ", end="")
c5.mul(c1, c2)
c5.display()
```

****
****
***

Q66. Write a definition for a class named Circle with attributes center and radius, where center is a Point object and radius is a number. Instantiate a Circle object that represents a circle with its center at (150, 100) and radius 75. Write a function named point_in_circle that takes a Circle and a Point and returns True if the Point lies in or on the boundary of the circle. Write a function named rect_in_circle that takes a Circle and a Rectangle and returns True if the Rectangle lies entirely in or on the boundary of the circle.

```
# Program To Check If The Given Point Lies Within The Circle

import copy
import math


class Point:
    """Represents a Point.
    Attributes: x, y
    """


class Rectangle:
    """ Represents a rectangle
    attributes:width,height,corner
    """


def distance_between_points(p1,p2):
    return math.sqrt((p1.x-p2.x)*(p1.x-p2.x) + (p1.y-p2.y)*(p1.y-p2.y))


class Circle:
    """Represents a circle.
    Attributes: center, radius
    """


def print_point(p):
    print('(%g,%g)'%(p.x, p.y))


def point_in_circle(point, circle):
    """Checks whether a point lies inside a circle (or on the boundary).
    point: Point object
    circle: Circle object
    """
    d = distance_between_points(point, circle.center)
    print("Distance Between Centre Of Circle And Given Point Co-Ordinates Is:  ", d)
    return d <= circle.radius


def rect_in_circle(rect, circle):
    """Checks whether the corners of a rect fall in/on a circle.
    rect: Rectangle object
    circle: Circle object
    """
    p = copy.copy(rect.corner)

    if not point_in_circle(p, circle):
        return False
    p.x += rect.width

    if not point_in_circle(p, circle):
        return False
    p.y -= rect.height

    if not point_in_circle(p, circle):
        return False
    p.x -= rect.width

    if not point_in_circle(p, circle):
        return False

    return True


def rect_circle_overlap(rect, circle):
    """Checks whether any corners of a rect fall in/on a circle.
    rect: Rectangle object
    circle: Circle object
    """
    p = copy.copy(rect.corner)

    if point_in_circle(p, circle):
        return True
    p.x += rect.width

    if point_in_circle(p, circle):
        return True
    p.y -= rect.height

    if point_in_circle(p, circle):
        return True
    p.x -= rect.width

    if point_in_circle(p, circle):
        return True

    return False
```

```
box = Rectangle()
box.width = 100.0
box.height = 200.0
box.corner = Point()
box.corner.x = 50.0
box.corner.y = 50.0
print("X Co-Ordinate Of Corner Of Box: ", box.corner.x)
print("Y Co-Ordinate Of Corner Of Box: ", box.corner.y)


circle = Circle
circle.center = Point()
circle.center.x = 150.0
circle.center.y = 100.0
circle.radius = 75.0

print("X Co-Ordinate Of Centre Of Circle: ", circle.center.x)
print("Y Co-Ordinate Of Centre Of Circle: ", circle.center.y)
print("Radius Of The Circle: ", circle.radius)


print("***********************************************************************************
**********************")
print("Is Point Present Within The Circle?", point_in_circle(box.corner, circle))
print("***********************************************************************************
**********************")
print("Is Rectangle Present Within The Circle?", rect_in_circle(box, circle))
print("***********************************************************************************
**********************")
print("Does The Rectangle And Circle Overlap?", rect_circle_overlap(box, circle))
print("***********************************************************************************
**********************")
```

****
****
***

Q67. Write A Progam That Creates A Class Time With Attributes Hours, Minutes And Seconds. With Appropriate Read And Write Functions, Create The Following Functions.

> Pure Functions That Adds Two Time Objects And Returns The Resultant Time Object.

> Modifier Called Increment Which Adds A Given Number Of Seconds To A Time Object.

> Write A Pure Version Of Increment That Creates And Returns A New Time Object Rather Than Modifying The Parameter.

> Pure Function Called mul_time That Takes A Time Object And Multiplies It With A Given Constant To Return A New Time Object That Contains The Product Of Original Time And The Constant.

```
# Program To Perform The Above Mentioned Operations

import copy


class Time:
    """
    attributes: hours, minutes, seconds
    """

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

    def add_time(self, t1, t2):

        t = Time(0, 0, 0)
        t.seconds = t1.seconds + t2.seconds
        t.minutes = t1.minutes + t2.minutes
        t.hours = t1.hours + t2.hours

        if t.seconds >= 60:
            t.seconds -= 60
            t.minutes += 1
            if t.minutes >= 60:
                t.minutes -= 60
                t.hours += 1

        return t

    def increment(self, t, sec):
        t.seconds += sec
        while t.seconds >= 60:
            t.seconds -= 60
            t.minutes += 1
            if t.minutes >= 60:
                t.minutes -= 60
                t.hours += 1

        return t

    def pure_increment(self, time, seconds):
        new_time = copy.copy(time)
        a = new_time.seconds + seconds
        b = new_time.minutes + a // 60
        new_time.hours += b // 60
        new_time.minutes = b % 60
        new_time.seconds = a % 60
        return new_time

    def mul_time(self, t, n):
        new_time = copy.copy(t)
        new_time.seconds *= n
        new_time.minutes *= n
        new_time.hours *= n

        while new_time.seconds >= 60:
            new_time.seconds -= 60
            new_time.minutes += 1

        while new_time.minutes >= 60:
            new_time.minutes -= 60
            new_time.hours += 1

        return new_time

    def print_time(self):
        return "Hours: {} Minutes: {} Seconds: {}".format(self.hours, self.minutes, self.seconds)


while True:
    print("Enter 1: Add Two Times To Find The Resultant Time")
    print("Enter 2: Increment Time By Adding Seconds To It")
    print("Enter 3: Pure Version Of Increment")
    print("Enter 4: Multiply Time")
    print("Enter 5: Exit")

    choice = int(input("Enter Your Choice: "))

    if choice == 1:
        A_hours = int(input("Enter The Value Of Hours For T1: "))
        if A_hours < 0 or A_hours > 23:
            print("Invalid Hour Value Entered!")
            continue

        A_minutes = int(input("Enter The Value Of Minutes For T1: "))
        if A_minutes < 0 or A_minutes > 59:
            print("Invalid Minute Value Entered!")
            continue

        A_seconds = int(input("Enter The Value Of Seconds For T1: "))
        if A_seconds < 0 or A_seconds > 59:
            print("Invalid Seconds Value Entered!")
            continue

        t1 = Time(A_hours, A_minutes, A_seconds)

        B_hours = int(input("Enter The Value Of Hours For T2: "))
        if B_hours < 0 or B_hours > 23:
            print("Invalid Hour Value Entered!")
            continue

        B_minutes = int(input("Enter The Value Of Minutes For T2: "))
        if B_minutes < 0 or B_minutes > 60:
            print("Invalid Minute Value Entered!")
            continue
        B_seconds = int(input("Enter The Value Of Seconds For T2: "))
        if B_seconds < 0 or B_seconds > 60:
            print("Invalid Seconds Value Entered!")
            continue

        t2 = Time(B_hours, B_minutes, B_seconds)

        t3 = Time(0, 0, 0)
        t3 = t3.add_time(t1, t2)
        print("Time T1: {}\nTime T2: {}\nTime T3(Time T1 + Time T2): {}".format(t1.print_time(), t2.print_time(),
                                                                             t3.print_time()))

        print("************************************************************************************************
*******************")

    elif choice == 2:

        sec = int(input("Enter The Value Of Seconds To Be Added To Given Time: "))

        hours = int(input("Enter The Value Of Hours For Your Desired Time: "))
        if hours < 0 or hours > 23:
            print("Invalid Hour Value Entered!")
            continue

        minutes = int(input("Enter The Value Of Minutes For Your Desired Time: "))
        if minutes < 0 or minutes > 59:
            print("Invalid Minute Value Entered!")
            continue

        seconds = int(input("Enter The Value Of Seconds For Your Desired Time: "))
        if seconds < 0 or seconds > 59:
            print("Invalid Seconds Value Entered!")
            continue

        t = Time(hours, minutes, seconds)
        print("Entered Time: ", t.print_time())

        t = t.increment(t, sec)
        print("Incremented Time: ", t.print_time())

        print("************************************************************************************************
*******************")

    elif choice == 3:

        sec = int(input("Enter The Value Of Seconds To Be Added To Given Time: "))
        hours = int(input("Enter The Value Of Hours For For Your Desired Time: "))
        if hours < 0 or hours > 23:
            print("Invalid Hour Value Entered!")
            continue

        minutes = int(input("Enter The Value Of Minutes For Your Desired Time: "))
        if minutes < 0 or minutes > 59:
            print("Invalid Minute Value Entered!")
            continue

        seconds = int(input("Enter The Value Of Seconds For Your Desired Time: "))
        if seconds < 0 or seconds > 59:
            print("Invalid Seconds Value Entered!")
            continue

        t = Time(hours, minutes, seconds)
        print("Entered Time: ", t.print_time())

        t = t.pure_increment(t, sec)
        print("Incremented Time: ", t.print_time())

        print("*************************************************************************************
*******************")

    elif choice == 4:
        n = int(input("Enter The Value By Which You Wish To Multiply The Given Time: "))

        hours = int(input("Enter The Value Of Hours For Your Desired Time: "))
        if hours < 0 or hours > 23:
            print("Invalid Hour Value Entered!")
            continue

        minutes = int(input("Enter The Value Of Minutes For Your Desired Time: "))
        if minutes < 0 or minutes > 59:
            print("Invalid Minute Value Entered!")
            continue

        seconds = int(input("Enter The Value Of Seconds For Your Desired Time: "))
        if seconds < 0 or seconds > 59:
            print("Invalid Seconds Value Entered!")
            continue

        t = Time(hours, minutes, seconds)
        print("Entered Time: ", t.print_time())

        t = t.mul_time(t, n)
        print("Multiplied Time: ", t.print_time())

        print("*************************************************************************************
*******************")

    elif choice == 5:
        print("Thank You!!")
        exit(0)

    else:
        print("Invalid Input Entered. Please Try Again")



**********************************************************************************************
**********************************************************************************************
```

***