Python programming

1.write a python program to check whether a given year is a leap year or not?

n=int(input(“enter a number”))

if(n%4==0):

print(“leap year”)

else:

print(“not a leap year”)

2.write a python program to merge to sorted arrays in non increasing order?

def merge\_sorted\_arrays(arr1, arr2):

result = [0] \* (len(arr1) + len(arr2))

i = len(arr1) - 1

j = len(arr2) - 1

k = len(result) - 1

while i >= 0 and j >= 0:

if arr1[i] > arr2[j]:

result[k] = arr1[i]

i -= 1

else:

result[k] = arr2[j]

j -= 1

k -= 1

while i >= 0:

result[k] = arr1[i]

i -= 1

k -= 1

while j >= 0:

result[k] = arr2[j]

j -= 1

k -= 1

return result

arr1 = [9, 7, 5, 3]

arr2 = [10, 8, 6, 4]

print("Array 1:", arr1)

print("Array 2:", arr2)

merged\_array = merge\_sorted\_arrays(arr1, arr2)

print("Merged Array:", merged\_array)

3.Define a class taximeter having the following description?

class Taximeter:

def \_\_init\_\_(self):

self.base\_fare = 5

self.per\_km\_rate = 2.5

self.per\_min\_rate = 0.5

def calculate\_fare(self, distance, time):

distance\_fare = distance \* self.per\_km\_rate

time\_fare = time \* self.per\_min\_rate

total\_fare = self.base\_fare + distance\_fare + time\_fare

return total\_fare

def print\_receipt(self, distance, time):

total\_fare = self.calculate\_fare(distance, time)

print("Taxi Receipt")

print("----------------")

print(f"Base Fare: ${self.base\_fare:.2f}")

print(f"Distance: {distance:.2f} km")

print(f"Distance Fare: ${distance \* self.per\_km\_rate:.2f}")

print(f"Time: {time:.2f} minutes")

print(f"Time Fare: ${time \* self.per\_min\_rate:.2f}")

print(f"Total Fare: ${total\_fare:.2f}")

taxi = Taximeter()

distance = float(input("Enter distance traveled (km): "))

time = float(input("Enter time elapsed (minutes): "))

taxi.print\_receipt(distance, time)

**4.**write a python program to find the number of sentences starts with alphabets B?

def count\_sentences\_starting\_with\_b(text):

sentences = text.replace('?', '.').replace('!', '.').split('.')

count = 0

for sentence in sentences:

sentence = sentence.strip()

if sentence and sentence[0].lower() == 'b':

count += 1

return count

text = input("Enter the text: ")

print("Number of sentences starting with B:", count\_sentences\_starting\_with\_b(text))

``)

5.write a python program to display the diagonal elements in a matrix array and also find the sum of them?

def display\_diagonal\_elements(matrix):

rows = len(matrix)

cols = len(matrix[0]

if rows != cols:

print("Matrix is not square.")

return

diagonal\_elements = []

for i in range(rows):

diagonal\_elements.append(matrix[i][i])

print("Diagonal Elements:", diagonal\_elements)

diagonal\_sum = sum(diagonal\_elements)

print("Sum of Diagonal Elements:", diagonal\_sum)

matrix = [

[1, 2, 3],

[4, 5, 6],

[7, 8, 9]

]

print("Matrix:")

for row in matrix:

print(row)

display\_diagonal\_elements(matrix)

``]

6.write a python program which checks the given number is luck or not?

def is\_lucky(n):

return '4' not in str(n) and '13' not in str(n)

num = int(input("Enter a number: "))

if is\_lucky(num):

print(f"{num} is a lucky number.")

else:

print(f"{num} is not a lucky number.")

7.A peak elements is an element that is strictly greater than its neighbours?

def find\_peak\_elements(arr):

peak\_elements = []

n = len(arr)

if n == 1 or arr[0] > arr[1]:

peak\_elements.append(arr[0])

for i in range(1, n-1):

if arr[i-1] < arr[i] > arr[i+1]:

peak\_elements.append(arr[i])

if n > 1 and arr[n-1] > arr[n-2]:

peak\_elements.append(arr[n-1])

return peak\_elements

arr = [1, 3, 5, 3, 2, 1]

print("Peak Elements:", find\_peak\_elements(arr))

8.write a python program to find the maximum total from top to bottom of the triangle below?

def max\_total(triangle):

rows = len(triangle)

for i in range(rows - 2, -1, -1):

for j in range(len(triangle[i])):

triangle[i][j] += max(triangle[i + 1][j], triangle[i + 1][j + 1])

return triangle[0][0]

triangle = [

[3],

[7, 5],

[1, 6, 7],

[8, 2, 1, 0]

]

print("Maximum Total:", max\_total(triangle))