LIST Questions

1. Given a list of numbers, write a Python program to find the sum of all the elements in the list.?

Input: arr = [2,4,5,10], i = 1, j = 3

Output: 19

***Code:-***

arr = [2,4,5,10] i = 4j = 3sum = 0for idx in range(len(arr)): if idx >= i and idx <=j: sum += arr[idx]print(“Ans is:- ”,sum)

Graphical user interface, text, application

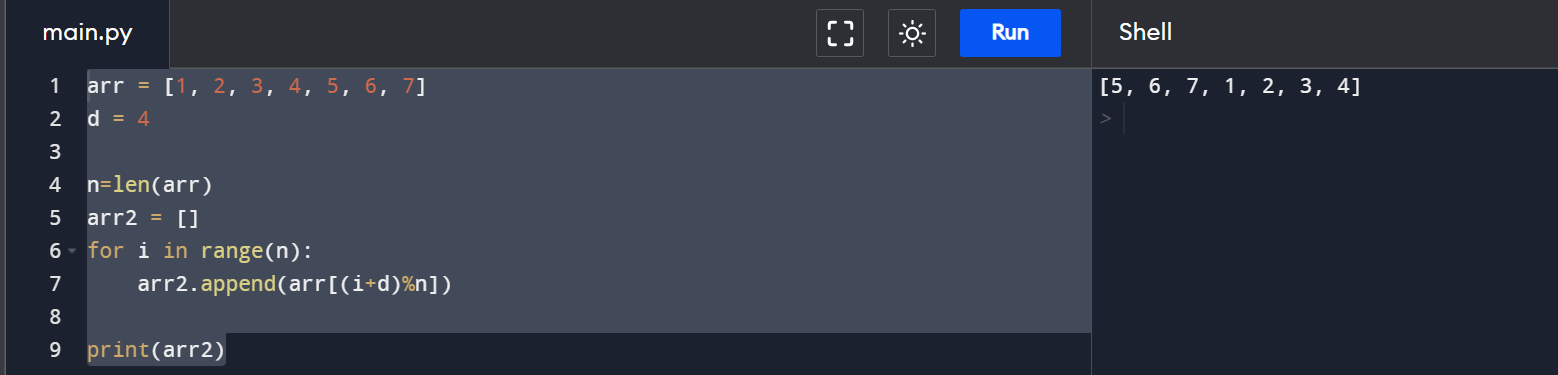
Description automatically generated

2. Given an array of integers arr[] of size N and an integer, the task is to rotate the array elements to the left by d positions.?

***Code:-***

arr = [1, 2, 3, 4, 5, 6, 7]d = 4

n=len(arr)arr2 = []for i in range(n): arr2.append(arr[(i+d)%n])print(arr2)



3. Second most repeated word in a sequence in Python.

Given a sequence of strings, the task is to find out the second most repeated (or frequent) string in the given sequence.

Input : ["aaa", "bbb", "ccc", "bbb", "aaa", "aaa"]

Output : bbb

***Code:-***

arr = ["aaa", "bbb", "ccc", "bbb", "aaa", "aaa"]

freq\_dic = {}

for i in arr:

if i in freq\_dic:

freq\_dic[i] += 1

else:

freq\_dic[i] = 1

sec\_max = 0

a = list(freq\_dic.values())

a.sort(reverse=True)

max = a[1]

for key,val in freq\_dic.items():

if val == max:

sec\_max = key

print(sec\_max)

Graphical user interface, text, application

Description automatically generated

4. Difference between two lists ?

Input: list1 = [10, 15, 20, 25, 30, 35, 40]

list2 = [25, 40, 35]

Output: [10, 20, 30, 15]

***Code:-***

list1 = [10, 15, 20, 25, 30, 35, 40]

list2 = [25, 40, 35]

list2 = [i for i in list1 + list2 if i not in list1 or i not in list2]

print(list2)

Graphical user interface, text

Description automatically generated

5. Print all positive numbers from given list using for loop Iterate each element in the list using for loop and check if number is greater than or equal to 0. If the condition satisfies, then only print the number.?Input: list1 = [12, -7, 5, 64, -14]

Output: 12, 5, 64

***Code***:-

list1 = [10, 15, -20, -25, 30, -35, 40]

list2 = [i for i in list1 if i > 0]

print(list2)

Graphical user interface, application

Description automatically generated

6. Write a Python program to flatten a given nested list structure.?

Original list: [0, 10, [20, 30], 40, 50, [60, 70, 80], [90, 100, 110, 120]]

Flatten list: [0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120]

***Code:-***

list1 = [0, 10, [20, 30], 40, 50, [60, 70, 80], [90, 100, 110, 120]]

flat\_list = []

for sublist in list1:

if type(sublist) == list:

for num in sublist:

flat\_list.append(num)

else:

flat\_list.append(sublist)

print(flat\_list)

Graphical user interface, text, application

Description automatically generated

7. Given an array and a value, find if there is a triplet in array whose sum is equal to the given value. If there is such a triplet present in array, then print the triplet and return true. Else return false.?

Input: array = [12, 3, 4, 1, 6, 9]

sum = 24

Output: 12, 3, 9

Explanation: There is a triplet (12, 3 and 9) present in the array whose sum is 24.

***Code:-***

array = [12, 3, 4, 1, 6, 9]sum = 24array.sort()n = len(array)for i in range(0, n-2): l = i + 1 r = n-1 while (l < r): if( array[i] + array[l] + array[r] == sum): print("Triplet is", array[i],', ', array[l], ', ', array[r]); break elif (array[i] + array[l] + array[r] < sum): l += 1 else: r -= 1

Text

Description automatically generated

Strings

1. Missing characters to make a string Pangram.?

Input : welcome to geeksforgeeks

Output : abdhijnpquvxyz

***Code:-***

Str = "The quick brown fox jumps over the dog"

present = [False for i in range(26)]

for i in range(len(Str)):

if (Str[i] >= 'a' and Str[i] <= 'z'):

present[ord(Str[i]) - ord('a')] = True

elif (Str[i] >= 'A' and Str[i] <= 'Z'):

present[ord(Str[i]) - ord('A')] = True

res = ""

for i in range(26):

if (present[i] == False):

res += chr(i + ord('a'))

print(res)



1. Find total number of non-empty substrings of a string with N characters.

Input : str = “abc”

Output : 6

Every substring of the given string : “a”, “b”, “c”, “ab”, “bc”, “abc”

***Code:-***

Str = "abc"

n = len(Str);

ans = int(n \* (n + 1) / 2);

print(ans)

Background pattern

Description automatically generated

1. Given a string containing lowercase and uppercase letters. Sort it in such a manner that the uppercase and lowercase letters come in an alternate manner but in a sorted way.?

Input : bAwutndekWEdkd

Output :AbEdWddekkntuw

Explanation:

Here we can see that letter ‘A’, ’E’, ’W’ are sorted

as well as letters “b, d, d, d, e, k, k, n, t, u, w” are sorted

but both appears alternately in the string as far as possible.

***Code:-***

Str = "bAwutndekWEdkd"

n = len(Str)

lCount = [0 for i in range(26)]

uCount = [0 for i in range(26)]

Str = list(Str)

for i in range(n):

if(Str[i].isupper()):

uCount[ord(Str[i]) - ord('A')] += 1

else:

lCount[ord(Str[i]) - ord('a')] += 1

i = 0

j = 0

k = 0

while(k < n):

while(i < 26 and uCount[i] == 0):

i += 1

if(i < 26):

Str[k] = chr(ord('A') + i)

k += 1

uCount[i] -= 1

while(j < 26 and lCount[j] == 0):

j += 1

if(j < 26):

Str[k] = chr(ord('a') + j)

k += 1

lCount[j] -= 1

print("".join(Str))

Text

Description automatically generated

1. Write a Python program that accepts a comma separated sequence of words as input and prints the unique words in sorted form (alphanumerically). ?

Sample Words : red, white, black, red, green, black

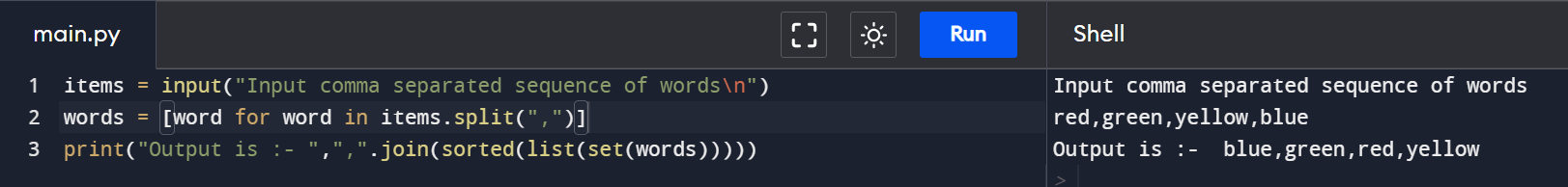
Expected Result : black, green, red, white,red

***Code:-***

items = input("Input comma separated sequence of words\n")

words = [word for word in items.split(",")]

print("Output is :- ",",".join(sorted(list(set(words)))))



1. Write a Python program to count the number of characters (character frequency) in a string. ?

Sample String : google.com'

Expected Result : {'g': 2, 'o': 3, 'l': 1, 'e': 1, '.': 1, 'c': 1, 'm': 1}

***Code:-***

input\_str = "www.google.com"

freq\_dict = {}

for i in input\_str:

if i in freq\_dict:

freq\_dict[i] += 1

else:

freq\_dict[i] = 1

print(freq\_dict)

Graphical user interface, application, Teams

Description automatically generated

1. find the frequency of minimum occurring character in a python string ?

The original string is : iNeuronNet.com

The minimum of all characters in GeeksforGeeks is : i

***Code:-***

input\_str = "iNeuronNet.com "

freq\_dict = {}

for i in input\_str:

if i in freq\_dict:

freq\_dict[i] += 1

else:

freq\_dict[i] = 1

res = min(freq\_dict, key = freq\_dict.get)

print(res)

Graphical user interface, text, application

Description automatically generated

1. Write a program to extract all the string characters which have odd number of occurrences

The original string is : geekforgeeks is best for geeks

The Odd Frequency Characters are : ['k', 'i', 't', 'g', 'e', 'b']

***Code:-***

input\_str = "geekforgeeks is best for geeks"

freq\_dict = {}

a = []

for i in input\_str:

if i in freq\_dict:

freq\_dict[i] += 1

else:

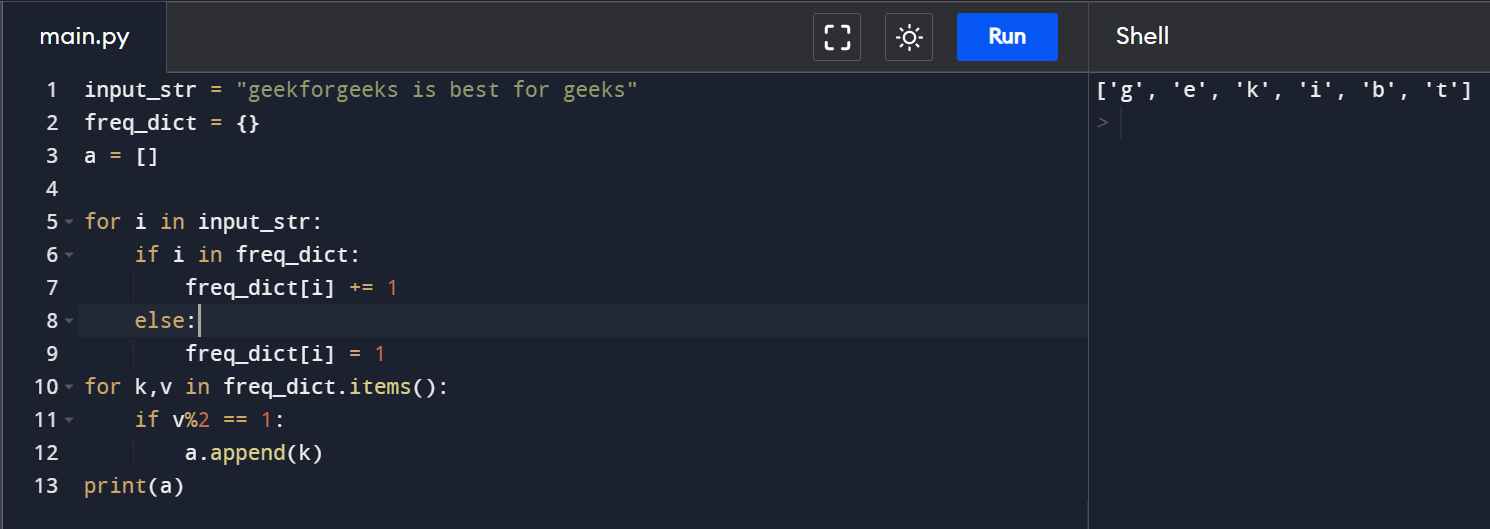
freq\_dict[i] = 1

for k,v in freq\_dict.items():

if v%2 == 1:

a.append(k)

print(a)



Dictionary

1. Given an input string and a pattern, check if characters in the input string follows the same order as determined by characters present in the pattern.

Assume there won’t be any duplicate characters in the pattern.?

Input:

string = "engineers rock"

pattern = "er";

Output: true

Explanation:

All 'e' in the input string are before all 'r'.

Input:

string = "engineers rock"

pattern = "gsr";

Output: false

Explanation:

There are one 'r' before 's' in the input string.

***Code:-***

string = "engineers rock"

pattern = "er"

l = len(pattern)

ans = True

if len(string) < l:

ans = False

for i in range(l - 1):

x = pattern[i]

y = pattern[i + 1]

last = string.rindex(x)

first = string.index(y)

if last == -1 or first == -1 or last > first:

ans = False

print(ans)

Text

Description automatically generated

2. Given a list and dictionary, map each element of list with each item of dictionary, forming nested dictionary as value.?

Input : test\_dict = {‘Gfg’ : 4, ‘best’ : 9}, test\_list = [8, 2]

Output : {8: {‘Gfg’: 4}, 2: {‘best’: 9}}

Explanation : Index-wise key-value pairing from list [8] to dict {‘Gfg’ : 4} and so on.

***Code:-***

test\_dict = {'Gfg' : 4, 'best' : 9}

test\_list = [8, 2]

res = {}

for key, ele in zip(test\_list, test\_dict.items()):

res[key] = dict([ele])

print(res)

Graphical user interface, text, application

Description automatically generated

3. Sort Dictionary key and values List ?

Input : test\_dict = {‘c’: [3], ‘b’: [12, 10], ‘a’: [19, 4]}

Output : {‘a’: [4, 19], ‘b’: [10, 12], ‘c’: [3]}

Input : test\_dict = {‘c’: [10, 34, 3]}

Output : {‘c’: [3, 10, 34]}

***Code:-***

test\_dict = {'c': [3], 'b': [12, 10], 'a': [19, 4]}

ans = dict()

for key in sorted(test\_dict):

ans[key] = sorted(test\_dict[key])

print(ans)

Graphical user interface, application

Description automatically generated

4. Remove all duplicates words from a given sentence ?

Input : Python is great and Java is also great

Output : is also Java Python and great

Code:-

Str = "Python is great and Java is also great"

inpt\_arr = Str.split(" ")

print(" ".join(list(set(inpt\_arr))))

Graphical user interface, application

Description automatically generated

5. Inversion in nested dictionary?

Input : test\_dict = {“a” : {“b” : {}}, “d” : {“e” : {}}, “f” : {“g” : {}}

Output : {‘b’: {‘a’: {}}, ‘e’: {‘d’: {}}, ‘g’: {‘f’: {}}

Explanation : Nested dictionaries inverted as outer dictionary keys and viz-a-vis.

Code:-

from collections import defaultdict

test\_dict = {'a' : {'b' : {}}, 'd' : {'e' : {}}, 'f' : {'g' : {}}}

flipped = defaultdict(dict)

for key, val in test\_dict.items():

for subkey, subval in val.items():

flipped[subkey][key] = subval

print(dict(flipped))

Graphical user interface, application, Teams

Description automatically generated

6. Given an array of n string containing lowercase letters. Find the size of largest subset of string which are anagram of each others. An anagram of a string is another string that contains same characters, only the order of characters can be different. For example, “abcd” and “dabc” are anagram of each other.?

Input: ant magenta magnate tan gnamate

Output: 3

Explanation

Anagram strings(1) - ant, tan

Anagram strings(2) - magenta, magnate,

gnamate

Thus, only second subset have largest

size i.e., 3

Input:

cars bikes arcs steer

Output: 2

***Code:-***

Str = "ant magenta magnate tan gnamate"

arr = Str.split(" ")

n = len(arr)

maxSize = 0

count = {}

for i in range(n):

freq=[0 for i in range(26)]

for ch in arr[i]:

freq[ord(ch) - ord('a')] += 1

temp = "".join(str(i) for i in freq)

if temp not in count:

count[temp] = 1

else:

count[temp] += 1

maxSize = max(maxSize, count[temp])

print(maxSize)

Text

Description automatically generated

Sets

1. Given two lists a, b. Check if two lists have at least one element common in them.

Input : a = [1, 2, 3, 4, 5], b = [5, 6, 7, 8, 9]

Output : True

Input : a=[1, 2, 3, 4, 5] b=[6, 7, 8, 9]

Output : False

***Code:-***

a = [1, 2, 3, 4, 5]

b = [5, 6, 7, 8]

a\_set = set(a)

b\_set = set(b)

if (a\_set & b\_set):

ans = True

else:

ans = False

print(ans)

Graphical user interface, application

Description automatically generated

2. Return a new set of identical items from two sets

set1 = {10, 20, 30, 40, 50} set2 = {30, 40, 50, 60, 70}

Expected output: {40, 50, 30}

***Code:-***

set1 = {10, 20, 30, 40, 50}

set2 = {30, 40, 50, 60, 70}

print(set1.intersection(set2))

Graphical user interface, background pattern

Description automatically generated with medium confidence

3. Maximum and Minimum in a Set without use of inbuild max/min functions?

Input : set = ([8, 16, 24, 1, 25, 3, 10, 65, 55])

Output : max is 65

Input : set = ([4, 12, 10, 9, 4, 13])

Output : min is 4

***Code:-***

arr = [1000, 11, 445, 1, 330, 3000]

n = len(arr)

mx=0

mn=0

if(n % 2 == 0):

mx = max(arr[0], arr[1])

mn = min(arr[0], arr[1])

i = 2

else:

mx = mn = arr[0]

i = 1

while(i < n - 1):

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

mx = max(mx, arr[i + 1])

mn = min(mn, arr[i])

else:

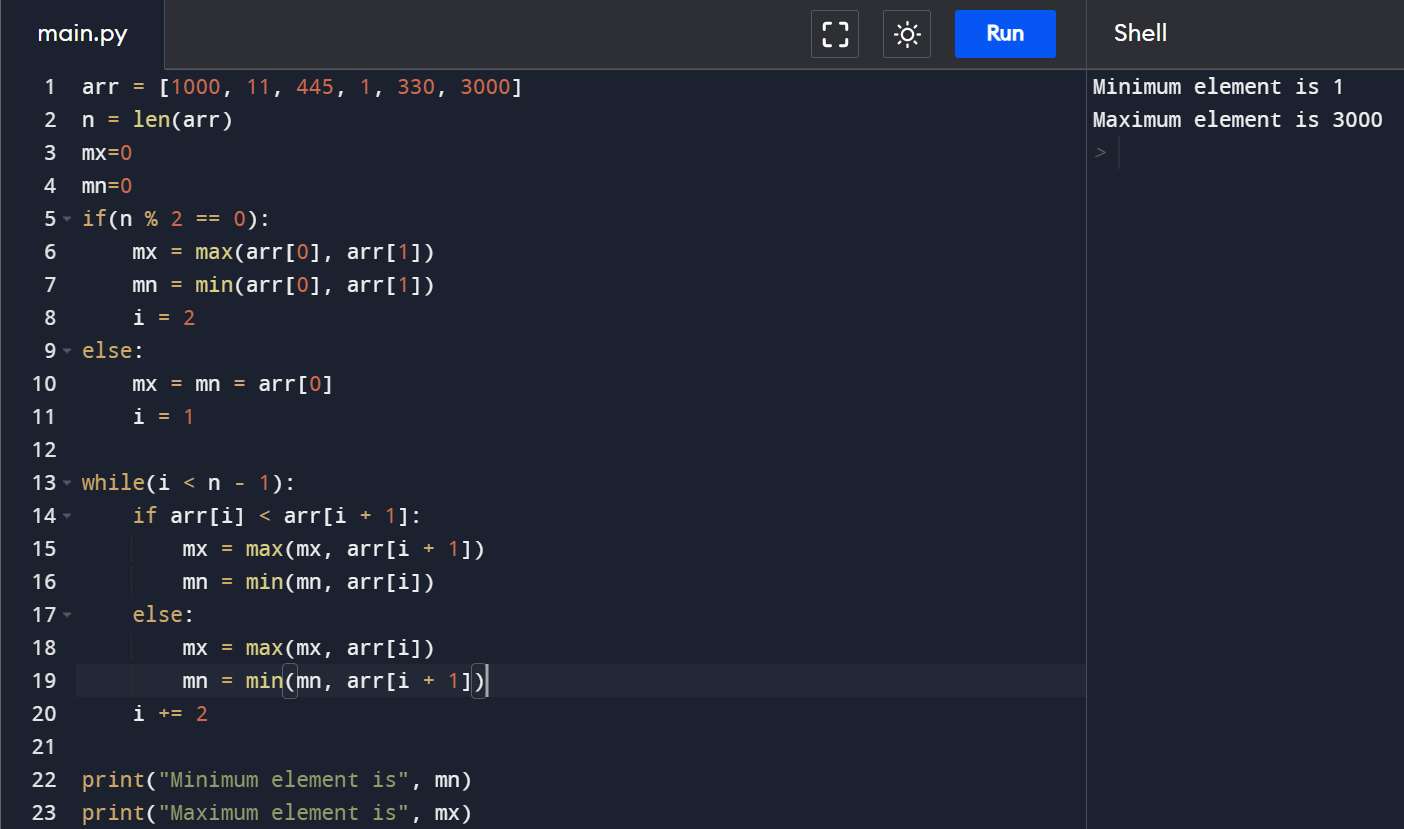
mx = max(mx, arr[i])

mn = min(mn, arr[i + 1])

i += 2

print("Minimum element is", mn)

print("Maximum element is", mx)



4. Write a Python program to check if a set is a subset of another set

Input :-

x: {'mango', 'apple'}

y: {'mango', 'orange'}

z: {'mango'}

output :-

If x is subset of y

False

False

If y is subset of x

False

False

If y is subset of z

False

False

If z is subset of y

Basic Questions 7

True

True

***Code:-***

setx = set(["apple", "mango"])

sety = set(["mango", "orange"])

setz = set(["mango"])

print("x: ",setx)

print("y: ",sety)

print("z: ",setz,"\n")

print("If x is subset of y")

print(setx <= sety)

print(setx.issubset(sety))

print("If y is subset of x")

print(sety <= setx)

print(sety.issubset(setx))

print("\nIf y is subset of z")

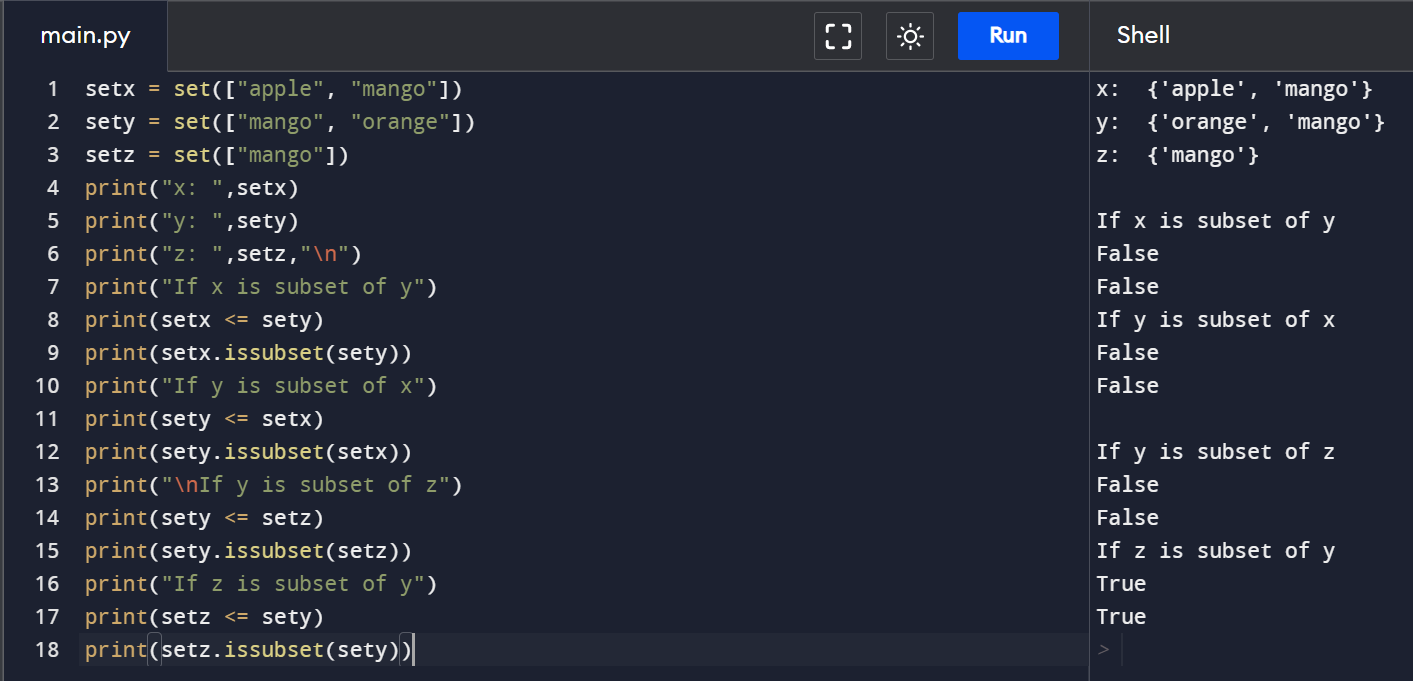
print(sety <= setz)

print(sety.issubset(setz))

print("If z is subset of y")

print(setz <= sety)

print(setz.issubset(sety))



5. Write a Python program to remove the intersection of a 2nd set from the 1st set

input:-

Original sets:

{1, 2, 3, 4, 5} {4, 5, 6, 7, 8}

Output:-

sn1: {1, 2, 3} sn2: {4, 5, 6, 7, 8}

***Code:-***

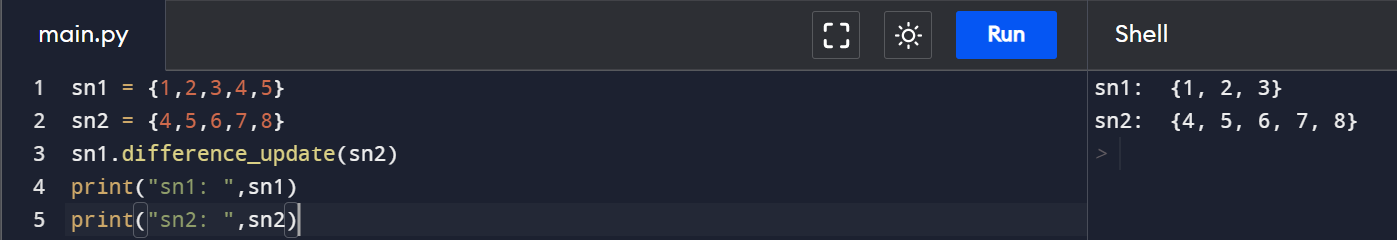
sn1 = {1,2,3,4,5}

sn2 = {4,5,6,7,8}

sn1.difference\_update(sn2)

print("sn1: ",sn1)

print("sn2: ",sn2)



Tuple

1. Remove Tuples of Length K ?

Input : test\_list = [(4, 5), (4, ), (8, 6, 7), (1, ), (3, 4, 6, 7)], K = 2

Output : [(4, ), (8, 6, 7), (1, ), (3, 4, 6, 7)]

Explanation : (4, 5) of len = 2 is removed.

***Code:-***

test\_list = [(4, 5), (4, ), (8, 6, 7), (1, ), (3, 4, 6, 7)]

K = 1

res = [ele for ele in test\_list if len(ele) != K]

print("List : " + str(res))

Graphical user interface, text, application

Description automatically generated

2. Removing duplicates from tuple ?

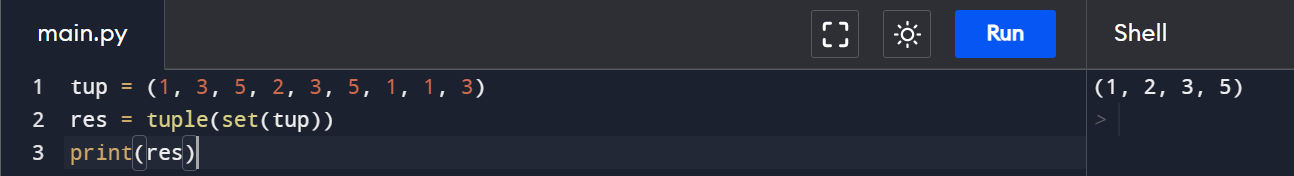
The original tuple is : (1, 3, 5, 2, 3, 5, 1, 1, 3)

The tuple after removing duplicates : (1, 3, 5, 2)***Code:-***

tup = (1, 3, 5, 2, 3, 5, 1, 1, 3)

res = tuple(set(tup))

print(res)



3. Flatten tuple of List to tuple ?

Input : test\_tuple = ([5], [6], [3], [8]) Output : (5, 6, 3, 8)

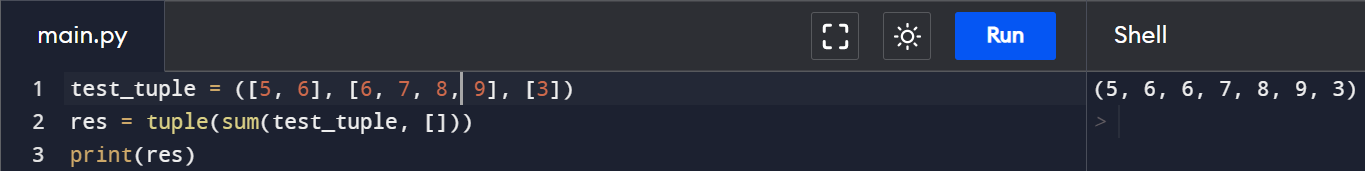
Input : test\_tuple = ([5, 7, 8]) Output : (5, 7, 8)

***Code:-***

test\_tuple = ([5, 6], [6, 7, 8, 9], [3])

res = tuple(sum(test\_tuple, []))

print(res)



4. Remove nested records from tuple?

The original tuple : (1, 5, 7, (4, 6), 10)

Elements after removal of nested records : (1, 5, 7, 10)

***Code:-***

test\_tup = (1, 5, 7, (4, 6), 10)

res = tuple()

for count, ele in enumerate(test\_tup):

if not isinstance(ele, tuple):

res = res + (ele, )

print(res)

Graphical user interface, application

Description automatically generated

5. Convert Binary tuple to Integer

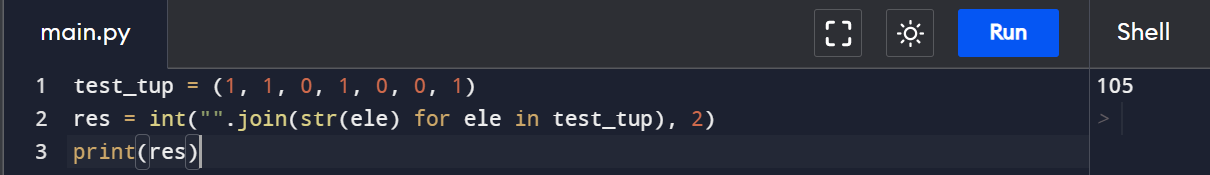
The original tuple is : (1, 1, 0, 1, 0, 0, 1)

Decimal number is : 105***Code:-***

test\_tup = (1, 1, 0, 1, 0, 0, 1)

res = int("".join(str(ele) for ele in test\_tup), 2)

print(res)



6. Sort Tuples by Total digits?

Input : test\_list = [(3, 4, 6, 723), (1, 2), (134, 234, 34)]

Output : [(1, 2), (3, 4, 6, 723), (134, 234, 34)]

Explanation : 2 < 6 < 8, sorted by increasing total digits.

***Code:-***

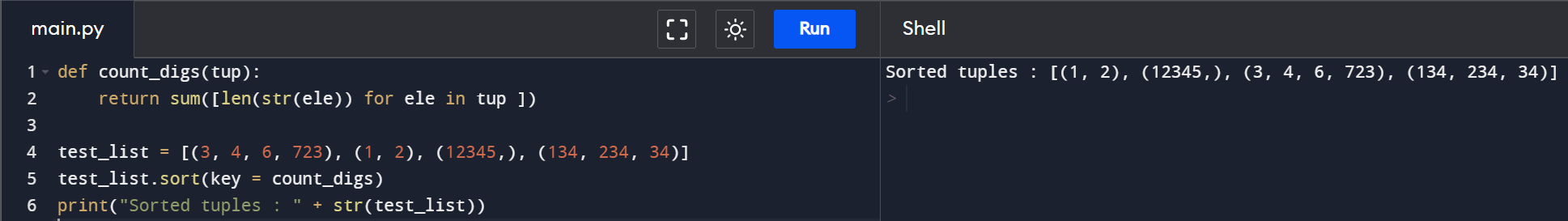
def count\_digs(tup):

return sum([len(str(ele)) for ele in tup ])

test\_list = [(3, 4, 6, 723), (1, 2), (12345,), (134, 234, 34)]

test\_list.sort(key = count\_digs)

print("Sorted tuples : " + str(test\_list))



MAP & Lambda Function

1. Write a Python program to add three given lists using Python map and lambda.?

Original list:

[1, 2, 3]

[4, 5, 6]

[7, 8, 9]

New list after adding above three lists:

[12, 15, 18]

***Code:-***

nums1 = [1, 2, 3]

nums2 = [4, 5, 6]

nums3 = [7, 8, 9]

result = map(lambda x, y, z: x + y + z, nums1, nums2, nums3)

print(list(result))

Graphical user interface, text

Description automatically generated

2. Write a Python program to convert a given list of tuples to a list of strings using map function?

Original list of tuples:

[('red', 'pink'), ('white', 'black'), ('orange', 'green')]

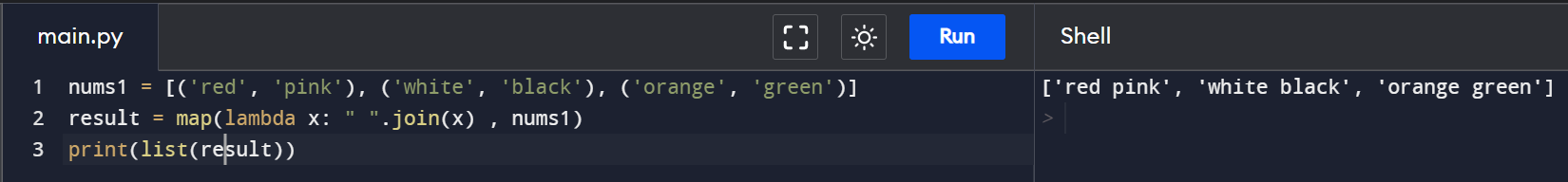
Convert the said list of tuples to a list of strings:

['red pink', 'white black', 'orange green']***Code:-***

nums1 = [('red', 'pink'), ('white', 'black'), ('orange', 'green')]

result = map(lambda x: " ".join(x) , nums1)

print(list(result))



3. Write a Python program to add two given lists and find the difference between

lists. Use map() function?

Original lists:

[6, 5, 3, 9]

[0, 1, 7, 7]

Result:

[(6, 6), (6, 4), (10, -4), (16, 2)]

***Code:-***

def addition\_subtrction(x, y):

return x + y, x - y

nums1 = [6, 5, 3, 9]

nums2 = [0, 1, 7, 7]

result = map(addition\_subtrction, nums1, nums2)

print(list(result))

Graphical user interface, application, Teams

Description automatically generated

4. Write a Python program to create Fibonacci series up to n using Lambda?

Fibonacci series upto 2:

[0, 1]

Fibonacci series upto 5:

[0, 1, 1, 2, 3]

***Code:-***

from functools import reduce

print("Enter n: ")

n = int(input())

fib = lambda n: reduce(lambda x, \_: x+[x[-1]+x[-2]],range(n-2), [0, 1])

print(fib(n))

Text

Description automatically generated

5. Write a Python program to find intersection of two given arrays using Lambda ?

[1, 2, 3, 5, 7, 8, 9, 10]

[1, 2, 4, 8, 9]

Intersection of the said arrays: [1, 2, 8, 9]

***Code:-***

array\_nums1 = [1, 2, 3, 5, 7, 8, 9, 10]

array\_nums2 = [1, 2, 4, 8, 9]

result = list(filter(lambda x: x in array\_nums1, array\_nums2))

print (result)

A screenshot of a computer

Description automatically generated with medium confidence

6. Write a Python program to find palindromes in a given list of strings using Lambda ?

Orginal list of strings:

['php', 'w3r', 'Python', 'abcd', 'Java', 'aaa']

List of palindromes:

['php', 'aaa']

***Code:-***

texts = ["php", "w3r", "Python", "abcd", "Java", "aaa"]

result = list(filter(lambda x: (x == "".join(reversed(x))), texts))

print(result)

Graphical user interface, text

Description automatically generated

7. Write a Python program to find the list with maximum and minimum length using lambda ?

Original list:

[[0], [1, 3], [5, 7], [9, 11], [13, 15, 17]]

List with maximum length of lists:

(3, [13, 15, 17])

List with minimum length of lists:

(1, [0])***Code:-***

def max\_list\_fn(input\_list):

max\_length = max(len(x) for x in input\_list )

max\_list = max(input\_list, key = lambda i: len(i))

return(max\_length, max\_list)

def min\_list\_fn(input\_list):

min\_length = min(len(x) for x in input\_list )

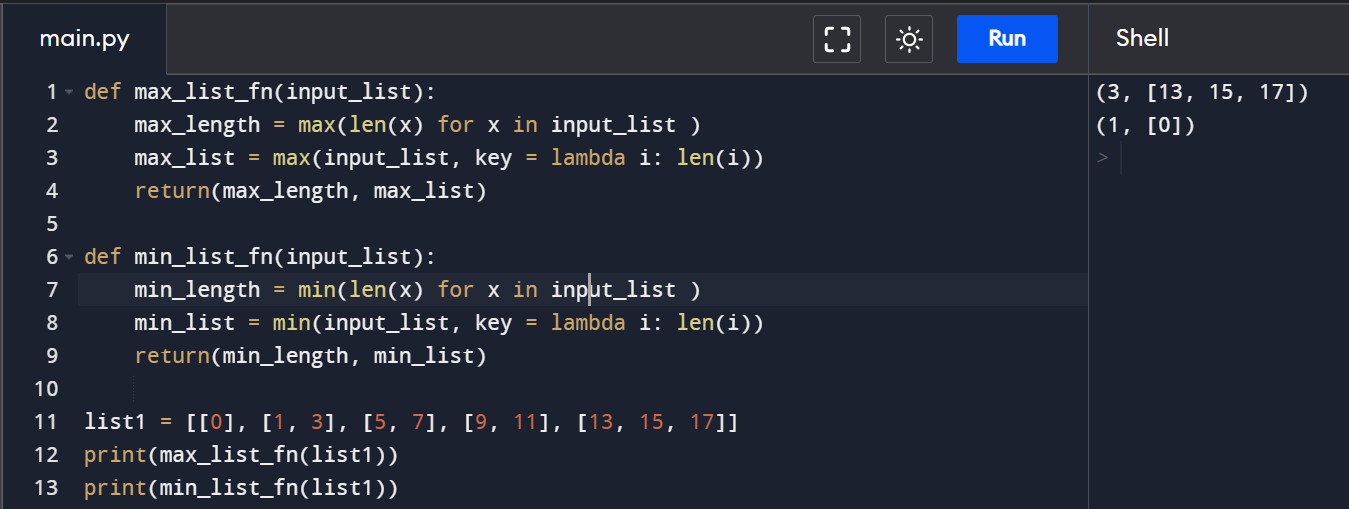
min\_list = min(input\_list, key = lambda i: len(i))

return(min\_length, min\_list)

list1 = [[0], [1, 3], [5, 7], [9, 11], [13, 15, 17]]

print(max\_list\_fn(list1))

print(min\_list\_fn(list1))



8. Write a Python program to triple all numbers of a given list of integers.?

Original list: (1, 2, 3, 4, 5, 6, 7)

Triple of said list numbers:

[3, 6, 9, 12, 15, 18, 21]

***Code:-***

nums = (1, 2, 3, 4, 5, 6, 7)

result = map(lambda x: x \*3, nums)

print(list(result))

Graphical user interface

Description automatically generated

List Comprehension

1. Use a nested list comprehension to find all of the numbers from 1–1000 that are divisible by any single digit besides 1 (2–9)?

***Code:-***

print([\*set([i for i in range(1,1000) for j in [2,3,4,5,6,7,8,9] if i % j == 0]

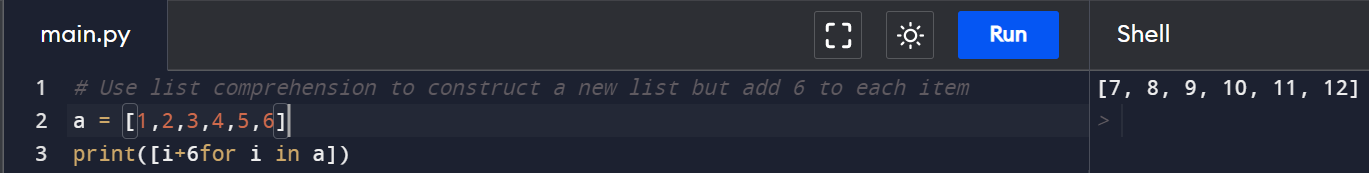
Graphical user interface, text

Description automatically generated

2. Use list comprehension to construct a new list but add 6 to each item?***Code:-***

a = [1,2,3,4,5,6]

print([i+6for i in a])



3. Suppose we want to create an output dictionary which contains only the odd numbers that are present in the input list as keys and their cubes as values. Let’s see how to do this using for loops and dictionary comprehension.?

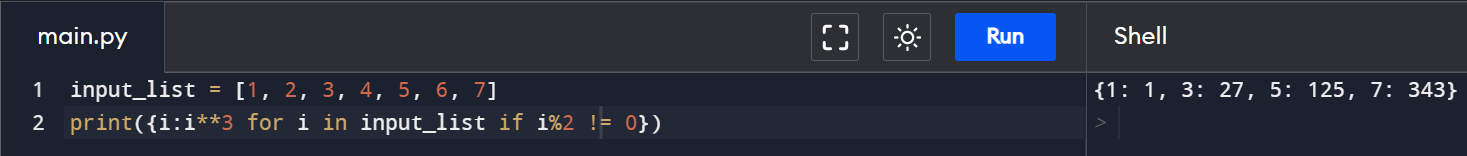
input = [1, 2, 3, 4, 5, 6, 7]

output :- {1: 1, 3: 27, 5: 125, 7: 343}

***Code:-***

input\_list = [1, 2, 3, 4, 5, 6, 7]

print({i:i\*\*3 for i in input\_list if i%2 != 0})



4. Given two lists containing the names of states and their corresponding capitals, construct a dictionary which maps the states with their respective capitals. Let’s see how to do this using for loops and dictionary comprehension.?

state = ['Gujarat', 'Maharashtra', 'Rajasthan']

capital = ['Gandhinagar', 'Mumbai', 'Jaipur']

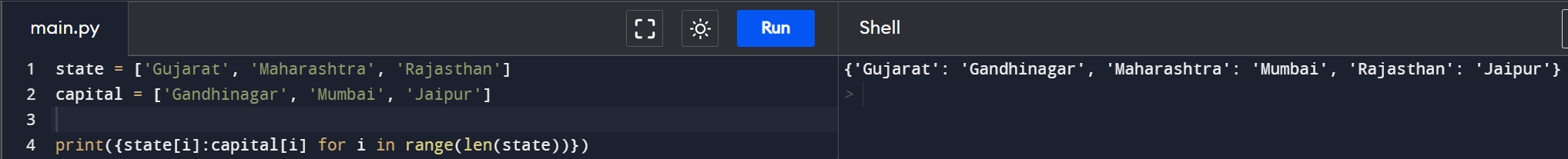
output:- {'Gujarat': 'Gandhinagar','Maharashtra': 'Mumbai', 'Rajasthan': 'Jaipur'}

***Code:***-

state = ['Gujarat', 'Maharashtra', 'Rajasthan']

capital = ['Gandhinagar', 'Mumbai', 'Jaipur']

print({state[i]:capital[i] for i in range(len(state))})



5. Transpose of Matrix using Comprehension?

Input :- [[1, 2, 3, 4], [4, 5, 6, 8]]

Output :- [[1, 4], [2, 5], [3, 6], [4, 8]]

***Code:-***

m = [[1,2],[3,4],[5,6]]

res= [[m[j][i] for j in range(len(m))] for i in range(len(m[0]))]

print(res)

Graphical user interface

Description automatically generated

6. We have a string ‘2459a09b’ and we want to extract all integer literals, and use int() to cast them into integers.?

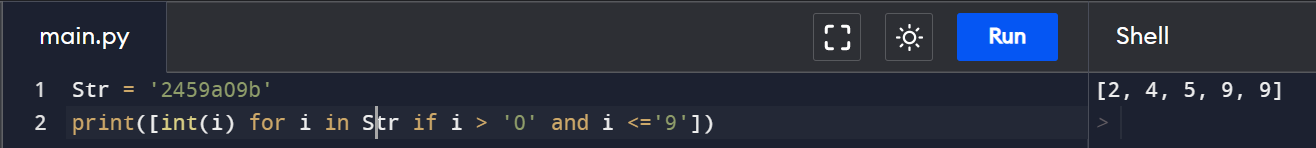
Input- '2459a09b'

output :- [2, 4, 5, 9, 9]

***Code:-***

Str = '2459a09b'

print([int(i) for i in Str if i > '0' and i <='9'])



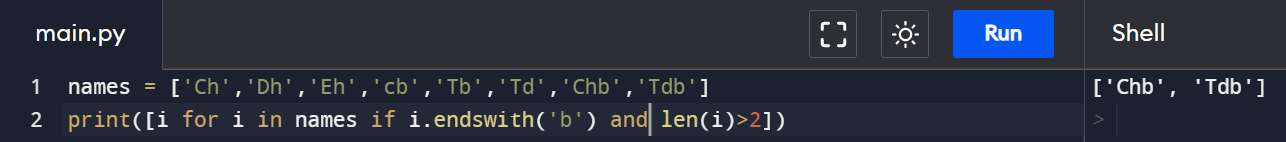
7. Finding the elements in a list in which elements are ended with the letter ‘b’ and the length of that element is greater than 2?

input :- names = ['Ch','Dh','Eh','cb','Tb','Td','Chb','Tdb']

output :- ['Chb', 'Tdb']

Code:-

names = ['Ch','Dh','Eh','cb','Tb','Td','Chb','Tdb']print([i for i in names if i.endswith('b') and len(i)>2])



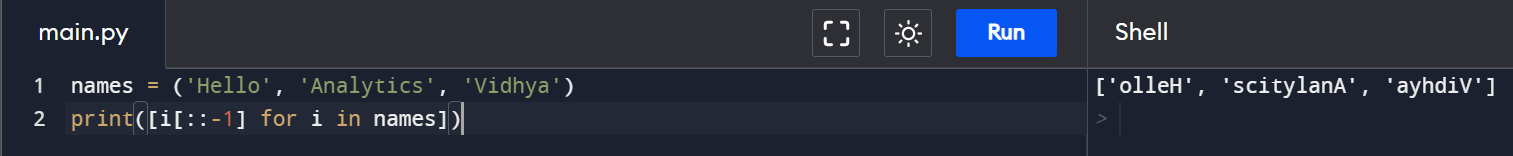
8. Reverse each String in a Tuple using list comprehension ?

Input :- 'Hello', 'Analytics', 'Vidhya'

output :- ['olleH', 'scitylanA', 'ayhdiV']

Code:-

names = ('Hello', 'Analytics', 'Vidhya')print([i[::-1] for i in names])



Object Oriented Programming

1. Define a property that must have the same value for every class instance (object)?

Output :-

Color: White, Vehicle name: School Volvo, Speed: 180, Mileage: 12

Color: White, Vehicle name: Audi Q5, Speed: 240, Mileage: 18

***Code:-***

class Vehicle:

# Class attribute

color = "White"

def \_\_init\_\_(self, name, max\_speed, mileage):

self.name = name

self.max\_speed = max\_speed

self.mileage = mileage

class Bus(Vehicle):

pass

class Car(Vehicle):

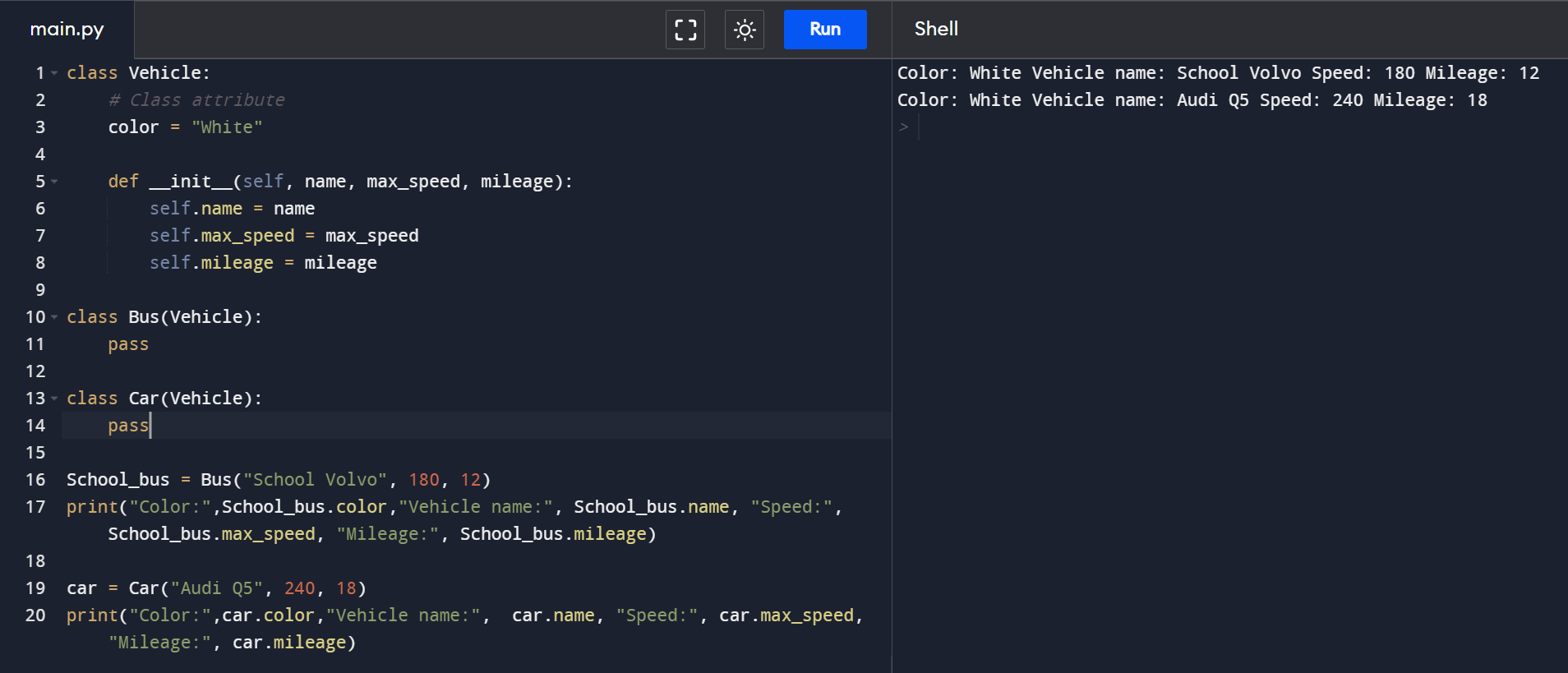
pass

School\_bus = Bus("School Volvo", 180, 12)

print("Color:",School\_bus.color,"Vehicle name:", School\_bus.name, "Speed:", School\_bus.max\_speed, "Mileage:", School\_bus.mileage)

car = Car("Audi Q5", 240, 18)

print("Color:",car.color,"Vehicle name:", car.name, "Speed:", car.max\_speed, "Mileage:", car.mileage)



1. Create a class with property decorator with setter and getter functions.

***Code:-***

class TP:

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

self.\_age = 0

@property

def age(self):

return self.\_age

@age.setter

def age(self, a):

self.\_age = a

mark = TP()

mark.age = 19

print(mark.age)

Graphical user interface, text, application

Description automatically generated

1. Create a class with Multi-level Inheritance ?

***Code:-***

class Grandfather:

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

self.grandfathername = grandfathername

class Father(Grandfather):

def \_\_init\_\_(self, fathername, grandfathername):

self.fathername = fathername

Grandfather.\_\_init\_\_(self, grandfathername)

class Son(Father):

def \_\_init\_\_(self, sonname, fathername, grandfathername):

self.sonname = sonname

Father.\_\_init\_\_(self, fathername, grandfathername)

def print\_name(self):

print('Grandfather name :', self.grandfathername)

print("Father name :", self.fathername)

print("Son name :", self.sonname)

s1 = Son('Tejas', 'Rajendra', 'Mahadev')

s1.print\_name()

Text

Description automatically generated

1. write a decorator for a class?

Code:-

class MyDecorator:

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

self.function = function

def \_\_call\_\_(self):

self.function()

@MyDecorator

def function():

print("Ineuron.ai")

function()

Graphical user interface, application

Description automatically generated

1. Implement a stack ?

***Code:-***

stack = []

stack.append('a')

stack.append('b')

stack.append('c')

print('Initial stack')

print(stack)

print('Elements popped from stack:')

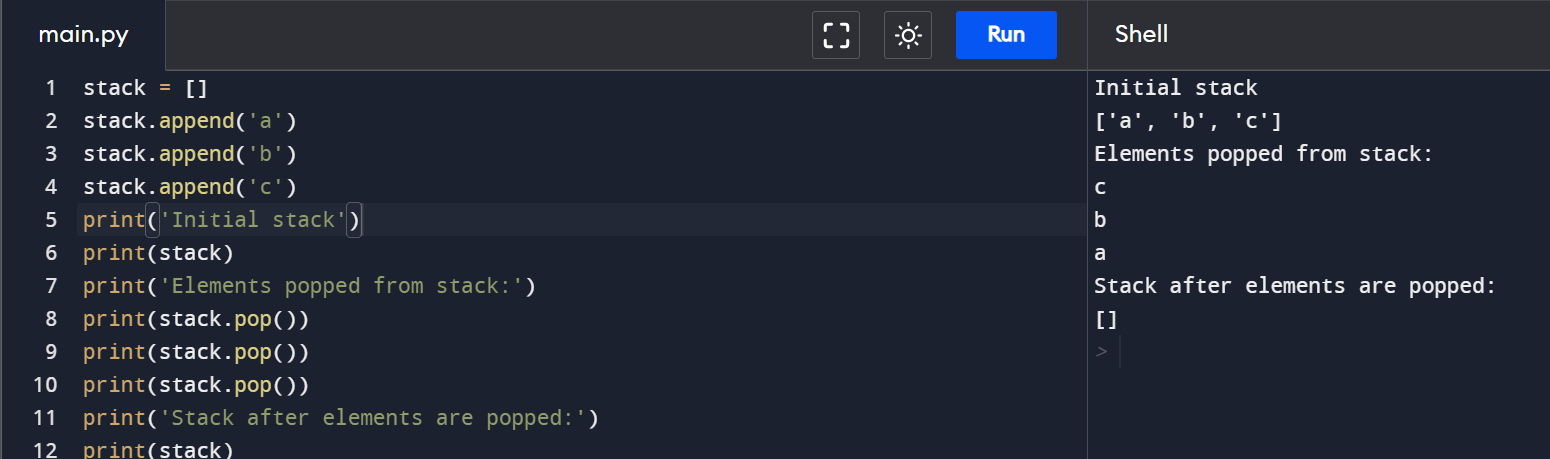
print(stack.pop())

print(stack.pop())

print(stack.pop())

print('Stack after elements are popped:')

print(stack)



1. Implement a Queue

***Code:-***

queue = []

queue.append('a')

queue.append('b')

queue.append('c')

print("Initial queue")

print(queue)

print("\nElements dequeued from queue")

print(queue.pop(0))

print(queue.pop(0))

print("\nQueue after removing elements")

print(queue)

Graphical user interface, application, Teams

Description automatically generated

1. Implement a Linked List

Code:-

class Node:

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

self.data = data

self.next = None

class LinkedList:

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

self.head = None

def printList(self):

temp = self.head

while (temp):

print(temp.data)

temp = temp.next

if \_\_name\_\_ == '\_\_main\_\_':

llist = LinkedList()

llist.head = Node(1)

second = Node(2)

third = Node(3)

llist.head.next = second

second.next = third

llist.printList()

Text

Description automatically generated

Exception Handling

1. Write Custom Exception to handle Zero division Error ?

***Code:-***

a =1

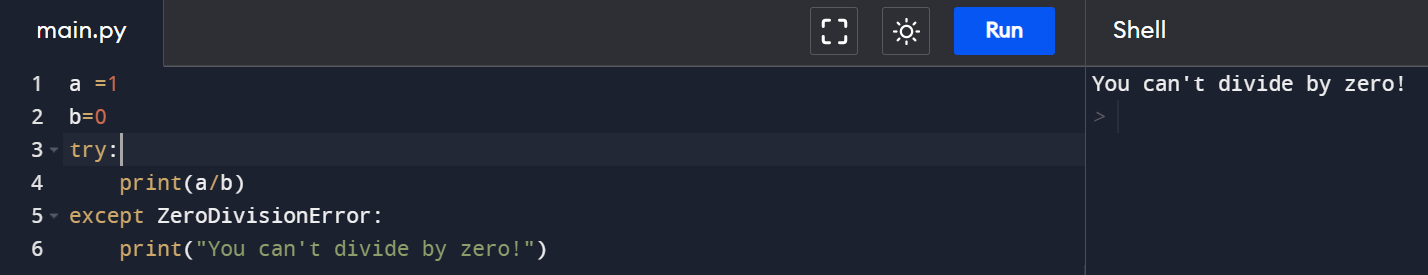
b=0

try:

print(a/b)

except ZeroDivisionError:

print("You can't divide by zero!")



1. Catching specific exception in Python.

def catch(msg):

try:

raise ValueError(msg)

except ValueError as e: # as e syntax added in ~python2.5

if str(e) != "foo":

print('not caught')

else:

print("caught!")

catch("foo")

catch("bar")

Graphical user interface, text

Description automatically generated