**LIST PROGRAMS**

1. Program to remove duplicates numbers entirely from the sorted array

Sample Input:

Array = {15, 14, 25, 14, 32, 14, 31}

Sample Output:

Sorted Array = {15, 25, 31, 32}

Test cases:

1. {16, 16, 16 16, 16}

2. {0, 0, 0, 0}

3. {-12, -78, -35, -42}

4. {1,2,3,7,8,9,4,5,6}

5. {1-2,2-3,3-4,4-5,5-6}

Program:

l=[1,1,2,3]

u=[]

for i in l:

if i not in u and l.count(i)==1:

u.append(i)

print(list(u))

2. Find the Mean, Median and Mode of the array of numbers? Sample Input:

Array of elements = {16, 18, 27, 16, 23, 21, 19} Sample Output:

Mean = 20

Median = 19

Mode = 16

Test cases:

1. Array of elements = {26, 28, 37, 26, 33, 31, 29}

2. Array of elements = {1.6, 1.8, 2.7, 1.6, 2.3, 2.1, .19}

3. Array of elements = {0, 160, 180, 270, 160, 230, 210, 190, 0}

4. Array of elements = {20, 18, 18, 27, 16, 27, 27, 19, 20}

5. Array of elements = {1000, 100, 1000, 100, 1000, 100, 1000, 100, 1000}

Program:

import statistics

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

print("mean:",statistics.mean(l))

print("median:",statistics.median(l))

print("mode:",statistics.mode(l))

3. Python Program to create a list of all numbers in a range which are perfect squares and the sum of the digits of the number is less than 10.

Sample Input & Output:

Enter lower range: 1

Enter upper range: 40 [1, 4, 9, 16, 25, 36]

Test case:

1. Enter lower range: 50 Enter upper range: 100

2. Enter lower range: 5 Enter upper range: 8

3. Enter lower range: 10 Enter upper range: 5

4. Enter lower range: 500 Enter upper range: 500

5. Enter lower range: 0 Enter upper range: -100

Program:

lower\_range = int(input("Enter lower range: "))

upper\_range = int(input("Enter upper range: "))

result = []

for num in range(lower\_range, upper\_range + 1):

sqrt = int(num \*\* 0.5)

if sqrt \* sqrt == num:

digit\_sum = sum(map(int, str(num)))

if digit\_sum < 10:

result.append(num)

print(result)

4. Python Program to Find the Nth Largest Number in a List

Sample Input:

List : {14, 67, 48, 23, 5, 62}

N = 4

Sample Output:

4th Largest number: 23

Test cases: 1. N = 0

2. N = -5

3. N = 1

4. N = M 5. N = %

Program:

lst = [14, 67, 48, 23, 5, 62]

N = int(input("which largest number:"))

if N <= 0 or N > len(lst):

print("Invalid input for N.")

else:

sorted\_list = sorted(lst, reverse=True)

nth\_largest = sorted\_list[N-1]

print(f"{N}th Largest number: {nth\_largest}")

5. Python Program to Create a List of Tuples with the First Element as the Number and Second Element as the Square of the Number.

Sample Input:

Enter the lower range:45 Enter the upper range:49

Sample Output:

[(45, 2025), (46, 2116), (47, 2209), (48, 2304), (49, 2401)]

Test case:

1. Enter lower range: 50 Enter upper range: 100

2. Enter lower range: 5 Enter upper range: 8

3. Enter lower range: 10 Enter upper range: 5

4. Enter lower range: 500 Enter upper range: 500

5. Enter lower range: 0 Enter upper range: -100

Program:

lower\_range = int(input("Enter the lower range: "))

upper\_range = int(input("Enter the upper range: "))

if lower\_range > upper\_range:

print("Invalid input: Lower range is greater than upper range.")

else:

result = [(num, num\*\*2) for num in range(lower\_range, upper\_range + 1)]

print(result)

6. Write a program to find the number of composite numbers in an array of elements

Sample Input;:

Array of elements = {16, 18, 27, 16, 23, 21, 19} Sample Output:

Number of Composite Numbers = 5 Test cases:

1.Array of elements = {26, 28, 37, 26, 33, 31, 29}

2. Array of elements = {1.6, 1.8, 2.7, 1.6, 2.3, 2.1, .19}

3. Array of elements = {0, 160, 180, 270, 160, 230, 210, 190, 0}

4. Array of elements = {200, 180, 180, 270, 270, 270, 190, 200}

5. Array of elements = {100, 100, 100, 100, 100, 100, 100, 100}

Program:

# Sample input

array = [16, 18, 27, 16, 23, 21, 19]

# Count the number of composite numbers

count = 0

for num in array:

if num < 4:

continue

for i in range(2, num):

if num % i == 0:

count += 1

break

# Output the result

print("Number of Composite Numbers =", count)

7. Write a program to reverse an array Sample Input;:

Array of elements = {16, 18, 27, 16, 23, 21, 19} Sample Output:

Reverse Array elements = {19, 21 23, 16, 27, 18, 16} Test cases:

1.Array of elements = {26, 28, 37, 26, 33, 31, 29}

2. Array of elements = {1.6, 1.8, 2.7, 1.6, 2.3, 2.1, .19}

3. Array of elements = {0, 160, 180, 270, 160, 230, 210, 190, 0}

4. Array of elements = {200, 180, 180, 270, 270, 270, 190, 200}

5. Array of elements = {100, 100, 100, 100, 100, 100, 100, 100}

Program:

# Sample input

array = [16, 18, 27, 16, 23, 21, 19]

# Reverse the array using reverse indexing

reversed\_array = array[::-1]

# Output the result

print("Reverse Array elements =", reversed\_array)

8. Write a program to find the Non composite number in the array of numbers Sample Input:

Array of elements = {26, 28, 47, 26, 43, 51, 29} Sample Output:

Prime numbers in Array elements = {47, 43, 29} Test cases:

1.Array of elements = {26, 28, 37, 26, 33, 31, 29}

2. Array of elements = {1.6, 1.8, 2.7, 1.6, 2.3, 2.1, .19}

3. Array of elements = {0, 160, 180, 270, 160, 230, 210, 190, 0}

4.Array of elements = {20, 18, 18, 27, 27, 27, 190, 20}

5. Array of elements = {100, 100, 100, 100, 100, 100, 100, 100}

Program:

# Sample input

array = [26, 28, 47, 26, 43, 51, 29]

# Find the non-composite numbers

non\_composite\_numbers = []

for num in array:

if num < 2:

continue

count = 0

for i in range(2, num):

if num % i == 0:

count += 1

break

if count == 0:

non\_composite\_numbers.append(num)

# Output the result

print("Non-composite numbers in Array elements =", non\_composite\_numbers)

9. Write a program to print the number of negative numbers in the list of numbers

Sample Input:

List of elements = {16, -18, 27, -16, 23, -21, 19} Sample Output:

Number of negative numbers in List of elements = 3 Test cases:

1. List of elements = {-26, 28, 37, -26, 33, -31, -29}

2. List of elements = {1.6, 1.8, 2.7, -1.6, 2.3, -2.1, .19}

3. List of elements = {0, 160, 180, 270, 160, 230, 210, 190, 0}

4. List of elements = {-16, 2.8, -7, -1.5, 2.8, -2.8, -.19}

5. List of elements = {-160, -160, -180, -270, -160, -230, -210, 1-90, 0}

Program:

numbers = [16, -18, 27, -16, 23, -21, 19]

count = 0

for num in numbers:

if num < 0:

count += 1

print("Number of negative numbers in the list:", count)

10. Find the Mth maximum number and Nth minimum number in an array and then find the sum of it, difference of it and product of it.

Sample Input:

Array of elements = {14, 16, 87, 36, 25, 89, 34}

M = 1

N = 3

Sample Output:

1st Maximum Number = 89 3rd Minimum Number = 25 Sum = 114

Difference = 64

Product = 2225 Test cases:

1. {16, 16, 16 16, 16}, M = 0, N = 1

2. {0, 0, 0, 0}, M = 1, N = 2

3. {-12, -78, -35, -42, -85}, M = 3 , N = 3

4. {15, 19, 34, 56, 12}, M = 6 , N = 3

5. {85, 45, 65, 75, 95}, M = 5 , N = 7

Program:

l=[14,67,48,23,5,62]

asc=sorted(l)

dsc=asc[::-1]

n=int(input("enter which largest number:"))

m=int(input("enter which smallestnumber:"))

print(n,"largest:",dsc[n-1])

print(m,"smallest:",asc[m-1])

print("sum:",dsc[n-1]+asc[m-1])

print("difference:",dsc[n-1]-asc[m-1])

print("product:",dsc[n-1]\*asc[m-1])

11. Write a program to merge two sorted lists to the third list.

Input: list1 = [1,2,4], list2 = [1,3,4]

Output: [1,1,2,3,4,4]

Program:

list1=[1,2,4]

list2=[1,2,4]

output=list1+list2

print(output)

12. A peak element is an element that is strictly greater than its neighbours. Given a **0-**

**indexed** integer array nums, find a peak element, and return its index. If the array contains

multiple peaks, return the index to **any of the peaks**

Input: nums = [1,2,3,1]

Output: 2

Program:

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

peaks = []

# Check if the first or last element is a peak

if len(arr) == 1 or arr[0] >= arr[1]:

peaks.append(arr[0])

if arr[-1] >= arr[-2]:

peaks.append(arr[-1])

# Iterate through the array and check for peaks

for i in range(1, len(arr) - 1):

if arr[i] >= arr[i - 1] and arr[i] >= arr[i + 1]:

peaks.append(arr[i])

if peaks:

print("Peak elements:", peaks)

else:

print("No peak elements found")

13. Write a program to read the numbers until -1 is encountered. Find the average of positive numbers and negative numbers entered by user. Restrict the decimal up to 2 digits.

Sample Input:

Enter -1 to exit… Enter the number: 7 Enter the number: -2 Enter the number: 9 Enter the number: -8 Enter the number: -6 Enter the number: -4 Enter the number: 10 Enter the number: -1

Sample Output:

The average of negative numbers is: -5.00 The average of positive numbers is: 8.67

Test cases:

1. -1,43, -87, -29, 1, -9

2. 73, 7-6,2,10,28,-1

3. -5, -9, -46,2,5,0

4. 9, 11, -5, 6, 0,-1

5. -1,-1,-1,-1,-1

Program:

l=[]

while True:

num=int(input("enter list elements:"))

if num==-1:

break

else:

l.append(num)

pos\_nums,neg\_nums=[],[]

pos\_avg,neg\_avg=0.0,0.0

for i in range(len(l)):

if l[i]>0:

pos\_nums.append(l[i])

pos\_avg+=l[i]

else:

neg\_nums.append(l[i])

neg\_avg+=l[i]

print("positive avg:",pos\_avg/len(pos\_nums))

print("negative avg:",neg\_avg/len(neg\_nums))

14.Write a Python function sumsquare(l) that takes a nonempty list of integers and returns a list [odd, even], where odd is the sum of squares of all the odd numbers in l and even is the sum of squares of all the even numbers in l.

Sample Input:

Enter the number of elements:7

Enter the element: 18

Enter the element:9

Enter the element:1

Enter the element:12

Enter the element:13

Enter the element:4

Enter the element:30

Output:

[251,1384]

Program:

def sumsquare(l):

odd\_sum = 0

even\_sum = 0

for num in l:

if num % 2 == 0:

even\_sum += num \*\* 2

else:

odd\_sum += num \*\* 2

return [odd\_sum, even\_sum]

n = int(input("Enter the number of elements: "))

l = []

for i in range(n):

l.append(int(input("Enter the element: ")))

output = sumsquare(l)

print(output)

15. Given an array of integers nums, return the number of good pairs.

A pair (i, j) is called good if nums[i] == nums[j] and i < j.

Input: nums = [1,2,3,1,1,3]

Output: 4

Explanation: There are 4 good pairs (0,3), (0,4), (3,4), (2,5) 0-indexed.

Program:

l=[1,2,3,1,1,3]

c=0

for i in range(0,len(l)):

for j in range(i+1,len(l)):

if l[i]==l[j]:

print("(",i,j,")")

c+=1

print("number of good pairs:",c)

16. How Many Numbers Are Smaller Than the Current Number

Given the array nums, for each nums[i] find out how many numbers in the array are smaller

than it. That is, for each nums[i] you have to count the number of valid j's such that j !=

i and nums[j] < nums[i].

Input: nums = [8,1,2,2,3]

Output: [4,0,1,1,3]

Program:

l=[8,1,2,2,3]

l1=[]

for i in range(len(l)):

c=0

for j in range(len(l)):

if l[i]>l[j]:

c+=1

l1.append(c)

print(l1)

17. A party has been organised on a cruise. The party is organised for a limited time (T). The

number of guests entering (E[i]) and leaving (L[i]) the party at every hour is represented as

elements of the array. The task is to find the maximum number of guests present on the cruise

at any given instance within T hours.

Sample Input:

5 ---> Value of T

[7,0,5,1,3] ---> E[], element of E[0] to E[N-1], where input each element is separated by new

line

[1,2,1,3,4] -----> L[],element of L[0] to L[N-1], where input each element is separated by

new line

Sample Output:

8 -----> Maximum number of guests on cruise at an instance.

Program:

t=int(input("enter instance time:"))

e=[7,0,5,1,3]

l=[1,2,1,3,4]

x=[0,0,0,0,0]

for i in range(t):

if t>len(e) or t>len(l):

print("out of index")

else:

x[i]=(x[i-1]+e[i])-l[i]

print(x[i],end=" ")

print("\nmax:",max(x))

18. Permutations

Given a collection of numbers, nums, that might contain duplicates, return all possible unique

permutations in any order.

**Test cases:**

1.Input: nums = [1,1,2]

Output:

[[1,1,2],

[1,2,1],

[2,1,1]]

Program:

import itertools

p = itertools.permutations([1, 1, 2])

unique = list(dict.fromkeys(list(p)))

output = [list(perm) for perm in unique]

print(output)

19. Given an integer n, return *a string array* answer *(****1-indexed****) where*:

answer[i] == "FizzBuzz" if i is divisible by 3 and 5.

answer[i] == "Fizz" if i is divisible by 3.

answer[i] == "Buzz" if i is divisible by 5.

answer[i] == i (as a string) if none of the above conditions are true.

Input: n = 5

Output: ["1","2","Fizz","4","Buzz"]

Program:

n = 5

result = []

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

if i % 3 == 0 and i % 5 == 0:

result.append("FizzBuzz")

elif i % 3 == 0:

result.append("Fizz")

elif i % 5 == 0:

result.append("Buzz")

else:

result.append(str(i))

print(result)

20. Python Program to Remove the Duplicate Items from a List

Sample Input:

Enter the number of elements in list:7

Enter element1:10

Enter element2:20

Enter element3:20

Enter element4:30

Enter element5:40

Enter element6:40

Enter element7:50

Sample Output:

Non-duplicate items: [10, 20, 30, 40, 50]

Program:

n = int(input("Enter the number of elements in the list: "))

lst = []

for i in range(n):

element = int(input(f"Enter element{i+1}: "))

lst.append(element)

non\_duplicate = list(set(lst))

print("Non-duplicate items:", non\_duplicate)

21. Suppose an array of length n sorted in ascending order is rotated between 1 and n times. For example, the array nums = [0,1,2,4,5,6,7] might become:

[4,5,6,7,0,1,2] if it was rotated 4 times.

[0,1,2,4,5,6,7] if it was rotated 7 times.

Notice that rotating an array [a[0], a[1], a[2], ..., a[n-1]] 1 time results in the array [a[n-1], a[0], a[1], a[2], ..., a[n-2]].

Given the sorted rotated array nums of unique elements, return the minimum element of this array.

Input: nums = [3,4,5,1,2]

Output: 1

Explanation: The original array was [1,2,3,4,5] rotated 3 times.

Program:

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

left = 0

right = len(nums) - 1

while left < right:

mid = left + (right - left) // 2

if nums[mid] > nums[right]:

left = mid + 1

else:

right = mid

min\_element = nums[left]

print(min\_element)

22. Given an array of integers nums sorted in non-decreasing order, find the starting and ending position of a given target value. If target is not found in the array, return [-1, -1].

Input: nums = [5,7,7,8,8,10], target = 8

Output: [3,4]

Program:

nums = [5, 7, 7, 8, 8, 10]

target = 7

start = -1

end = -1

for i in range(len(nums)):

if nums[i] == target:

if start == -1:

start = i

end = i

result = [start, end]

print(result)

23. Write a python program to insert an element in a specific index.

Sample input:

Enter the number of elements=5

Enter the elements: 47,34,21,89,12

Enter the element to be Inserted: 100

Position: 4

Sample output: [12,21,34,100,47,89]

Program:

elements = [47, 34, 21, 89, 12]

element\_to\_insert = 100

position = 4

elements.insert(position, element\_to\_insert)

print("Modified list:", elements)

24. Given a date, return the corresponding day of the week for that date.

The input is given as three integers representing the day, month and year respectively.

Return the answer as one of the following values {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"}.

Input: day = 31, month = 8, year = 2019

Output: "Saturday"

Program:

import datetime

def findDay(day, month, year):

# Create a datetime object for the given date

date = datetime.datetime(year, month, day)

# Get the weekday as an integer (0 = Monday, 1 = Tuesday, ..., 6 = Sunday)

weekday = date.weekday()

# Define a list of weekday names

weekdays = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"]

# Return the corresponding weekday name based on the weekday index

return weekdays[weekday]

# Example usage

day = 31

month = 8

year = 2019

result = findDay(day, month, year)

print(result)

25. Write a Python program to display the MSB and LSB of a number.

Sample input: 1267899

Sample output: LSB: 1 MSB: 9

Program:

s=input("enter number:")

print("lsb:",s[0])

print("msb:",s[-1])

26. Given a 1-indexed array of integers numbers that is already sorted in non-decreasing order, find two numbers such that they add up to a specific target number. Let these two numbers be numbers[index1] and numbers[index2] where 1 <= index1 < index2 <= numbers.length.

Return the indices of the two numbers, index1 and index2, added by one as an integer array [index1, index2] of length 2.

Input: numbers = [2,3,4], target = 6

Output: [1,3]

Program:

l = [2, 3, 4]

t = 6

left = 0

right = len(l) - 1

while left < right:

c\_sum = l[left] + l[right]

if c\_sum == t:

print([left + 1, right + 1])

break

elif c\_sum > t:

right -= 1

else:

left += 1

else:

result = []

print(result)

27. You are given with an array which contains integer elements. Sort these elements in ascending order. If any negative number is encountered it has to be replaced with the average of the array.

Input: [9,0,4,5,6] output: [0,4,5,6,9]

l = [-1, 2, 3, -4, 6]

avg = sum(l) / len(l)

for i in range(len(l)):

if l[i] < 0:

l[i] = avg

print(l)

28. Write a python program to compute the sum of all the multiples of 3 and 5 below 200.

c=0

l=[]

for i in range(1,200):

if i%3==0 and i%5==0:

l.append(i)

print(sum(l))

29. You have n jobs and m workers. You are given three arrays: difficulty, profit, and worker where:

For example, if three workers attempt the same job that pays $1, then the total profit will be $3. If a worker cannot complete any job, their profit is $0.

Return the maximum profit we can achieve after assigning the workers to the jobs.

Input: difficulty = [2,4,6,8,10], profit = [10,20,30,40,50], worker = [4,5,6,7]

Output: 100

Explanation: Workers are assigned jobs of difficulty [4,4,6,6] and they get a profit of [20,20,30,30] separately.

def maxProfitAssignment(difficulty, profit, worker):

jobs = sorted(zip(difficulty, profit))

res = i = best = 0

for ability in sorted(worker):

while i < len(jobs) and ability >= jobs[i][0]:

best = max(jobs[i][1], best)

i += 1

print(jobs,best)

res += best

return res

#Main Program

diff = [2,4,6,8,10]

pro = [10,20,30,40,50]

w = [4,5,6,7]

print(maxProfitAssignment(diff, pro, w))

30. Write a python program to find the frequency of all each element present in an array.

import pandas as pd

# declaring the list

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

count = pd.Series(l).value\_counts()

print("Element Count")

print(count)

31. Given an array of integers containing n+1 integers where each integer is in the range of[1,n]inclusive. There is only one repeated number. Return the repeated number.

Input: [1,3,4,2,2] output: 2

l = [1, 3, 4, 2, 2]

l1 = []

for i in l:

if l.count(i) > 1:

l1.append(i)

s=set(l1)

print(list(s))

32. Write an efficient algorithm that searches for a value target in an m x n integer matrix. This matrix has the following properties:

Integers in each row are sorted in ascending from left to right.

Integers in each column are sorted in ascending from top to bottom.

Input: matrix = [[1,4,7,11,15],[2,5,8,12,19],[3,6,9,16,22],[10,13,14,17,24],[18,21,23,26,30]], target = 5

Output: true

def searchMatrix(matrix, target):

for row in matrix:

if target in row:

return True

return False

matrix = [[1,4,7,11,15],[2,5,8,12,19],[3,6,9,16,22],[10,13,14,17,24],[18,21,23,26,30]]

target=5

print(searchMatrix(matrix,target))

33. Write a python program to create and display all the combinations of letters, selecting each letter from a different key in a dictionary.

Input: {1: [a,b], 2: [c,d]}

Output: ac,ad,bc,bd

import itertools

d ={'1':['a','b'], '2':['c','d']}

for combo in itertools.product(\*[d[k] for k in sorted(d.keys())]):

print(''.join(combo))

34. Patterns

|  |  |  |  |
| --- | --- | --- | --- |
| 1  1 2  1 2 3  1 2 3 4  1 2 3 4 5 | 1  2 2  3 3 3  4 4 4 4  5 5 5 5 5 | 10  10 20  10 20 30  10 20 30 40  10 20 30 40 50 | 0.1  0.1 0.2  0.1 0.2 0.3  0.1 0.2 0.3 0.4  0.1 0.2 0.3 0.4 0.5 |

35. Patterns

10

5 5

20 20 20

10 10 10 10