**General category programs**

1. Write a program to print all the Non-Prime numbers between A and B? Sample Input: A = 12 B = 19

Sample Output:

14, 15, 16, 18

Answer:

lower = int(input("enter a number : "))

upper = int(input("enter a number : "))

print("Prime numbers between", lower, "and", upper, "are:")

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

for i in range(2, num):

if (num % i) == 0:

print(num)

break

1. Find the year of the given Anniversary is leap year or not. If leap year then print the next Anniversary, if not leap year then print the previous Anniversary.

Sample Input:

Enter Date: 04/11/1947 Sample Output:

Given Anniversary Year: Non Leap Year. Anniversary Date: 04/11/1946

Answer:

date=input("enter date:")

d1=date.split('/')

l1=list(map(int,d1))

year=l1[2]

if year%4==0 or year%100!=0 and year%400==0:

print("it is an leap year")

print("the next leap year is:",year+4)

else:

print("it is not a leap year")

for i in range(1,5):

year-=i

if year%4==0 or year%100!=0 and year%400==0:

print("the previous leap year is:",year)

1. Write a program to print the given number is Perfect number or not?

Sample Input: Given Number: 6

Sample Output: Its a Perfect Number

Answer:

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

f=[]

s=0

for i in range(1,num):

if(num%i==0):

f.append(i)

s=s+i

if (s==num):

print("perfect number")

else:

print("not a perfect number")

1. Write a program to check if a given year is leap year or not. If it is leap year then print the next leap year, if it is non - leap year then prints the previous leap year.

Sample Input:

Enter Date: 1947 Sample Output:

Given year is Non Leap Year Leap Year: 1944

Answer:

year=int(input("enter year "))

if year%4==0 or year%100!=0 and year%400==0:

print("it is an leap year")

print("the next leap year is:",year+4)

else:

print("it is not a leap year")

for i in range(1,5):

year-=i

if year%4==0 or year%100!=0 and year%400==0:

print("the previous leap year is:",year)

1. Write a program to find the sum of digits of N digit number (sum should be single digit)

Sample Input:

Enter N value : 3

Enter 3 digit number: 143 Sample Output:

Sum of 3 digit number: 8

Answer:

n=int(input("enter the digits : "))

num=int(input("enter the number : "))

num1=str(num)

if(len(num1)!=n):

print("invalid")

else:

temp=num

sum=0

while(temp>0):

digit=temp%10

sum=sum+digit

temp=temp//10

print("Total = ",sum)

1. Program to find whether the given number is Armstrong number or not

Sample Input: Enter number: 153

Sample Output: Given number is Armstrong number

Answer:

n=int(input("enter the digits : "))

num=int(input("enter the number : "))

num1=str(num)

if(len(num1)!=n):

print("invalid")

else:

temp=num

sum1=0

while(temp>0):

digit=temp%10

sum1=sum1+digit\*digit\*digit

temp=temp//10

if num==sum1:

print("armstrong")

else:

print("not armstrong")

1. Program to find whether the given number is Harshad number or not

Sample Input: Enter number: 21

Sample Output: Given number is Harshad number

Answer:

n=int(input("enter the digits : "))

num=int(input("enter the number : "))

num1=str(num)

if(len(num1)!=n):

print("invalid")

else:

temp=num

sum=0

while(temp>0):

digit=temp%10

sum=sum+digit

temp=temp//10

if num%sum==0:

print("harshad")

else:

print("not harshad")

1. Program to find whether the given number is Happy number or not

Sample Input: Enter number: 19

Sample Output: Given number is happy number

Answer:

def happy(n):

temp=n

sum=0

while temp>0:

digit=temp%10

sum=digit\*\*2+sum

temp=temp//10

return sum

n=int(input("enter a number :"))

result=n

while result!=1 and result!=4:

result=(happy(result))

if result==1:

print("true")

elif result==4:

print("false")

1. Program to find whether the given number is Tech number or not

Sample Input: Enter number: 3025

Sample Output: Given number is Tech number

Answer:

n=input("number:")

dig=len(n)

if(dig%2==0):

half=int(dig/2)

n1=int(n[:half])

n2=int(n[half:])

sum=n1+n2

sq=sum\*\*2

num=int(n)

if(num==sq):

print("tech")

else:

print("not tech")

else:

print("not tech")

1. Write a program using function to calculate the simple interest. Suppose the customer is a senior citizen. She is being offered 15 percent rate of interest; he is being offered 12 percent rate of interest for all other customers, the ROI is 10 percent.

Sample Input:

Enter the principal amount: 200000 Enter the no of years: 3

Gender (m/f): m

Is customer senior citizen (y/n): n Sample Output:

Interest: 60000

Answer:

p=int(input("enter rate "))

y=int(input("enter year "))

c=input("citizen (s/n) ")

g=input("citizen (m/f) ")

if (c=="s" or g=="m"):

roi=p\*y\*12/100

print(roi)

elif (c=="s" or g=="f"):

roi=p\*y\*15/100

print(roi)

else:

roi=p\*y\*10/100

print(roi)

1. Find the number of factors for the given number and print the 1st N factors of the given number.

Sample Input: Given number: 100

N: 4

Sample Output: Number of factors = 9

1st 4 factors are: 1, 2, 4, 5

Answer:

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

f=[]

for i in range(1,num):

if(num%i==0):

f.append(i)

print("The factor of ",num," = ",f)

length=len(f)

print("The length of the list is ",length+1)

n=int(input("Enter inside list : "))

print(\*f[0:n])

1. Write a program to print number of factors and to print nth factor of the given number.

Sample Input: Given Number: 100

N = 4

Sample Output:

Number of factors = 9 4th factor of 100 = 5

Answer:

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

f=[]

for i in range(1,num):

if(num%i==0):

f.append(i)

print("The factor of ",num," = ",f)

length=len(f)

print("The length of the list is ",length)

n=int(input("Enter inside list : "))

if(n>length):

print("invalid ! enter new number ")

n1=int(input("Enter the nth number :"))

print(n," factor is = ",f[n-1])

else:

print(n," factor is = ",f[n-1])

1. Write a program to print unique permutations of a given number Sample Input:

Given Number: 143 Sample Output:

Permutations are:

134

143

314

341

413

431

Answer:

import itertools

n=input("enter number : ")

n1=list(itertools.permutations(n))

b=(["".join(a) for a in n1])

print(\*b)

1. Write a program to find the square, cube of the given decimal number Sample Input:

Given Number: 0.6

Sample Output: Square Number: 0.36 Cube Number:0.216

Answer:

n=float(input("enter number "))

sq=n\*n

cube=n\*n\*n

print(round(sq,2))

print(round(cube,2))

1. Write a program to convert the Binary to Decimal, Octal Sample Input:

Given Number: 1101 Sample Output:

Decimal Number: 13 Octal: 15

1. Add Binary

Given two binary strings a and b, return their sum as a binary string.

a and b consist only of '0' or '1' characters.

Each string does not contain leading zeros except for the zero itself.

**Test cases:**

1.Input: a = "11", b = "1"

1. Output: "100"

Answer:

b1=input("")

b2=input("")

sum=bin(int(b1,2)+int(b2,2))

print(sum[2:])

1. Write a program for matrix multiplication?

Sample Input:

Mat1 =

Mat2 =

Sample Output:

Mat Sum =

Answer:

a=[[2,1,4],

[2,1,2],

[3,4,3]]

b=[[1,2,3],

[4,5,6],

[7,8,9]]

e=[[0,0,0],

[0,0,0],

[0,0,0]]

for i in range(len(a)):

for j in range(len(b)):

for k in range(len(b)):

e[i][j]+=a[i][k]\*b[k][j]

print("multipication.....")

for n in e:

print(n)

1. Write a program for matrix addition?

Sample Input:

Mat1 =

Mat2 =

Sample Output:

Mat Sum =

Answer:

a=[[2,1,4],

[2,1,2],

[3,4,3]]

b=[[1,2,3],

[4,5,6],

[7,8,9]]

c=[[0,0,0],

[0,0,0],

[0,0,0]]

for i in range(len(a)):

for j in range(len(b)):

c[i][j]=a[i][j]+b[i][j]

print("addtion.....")

for l in c:

print(l)

1. Find the LCM and GCD of n numbers?

Sample Input:

N value = 2

Number 1 = 16

Number 2 = 20

Sample Output: LCM = 80 GCD = 4

Answer:

import math

num1 = 16

num2 = 20

print("The lcm is ",math.lcm(num1,num2))

print("The gcd is ",math.gcd(num1,num2))

20. Program to find day of the week

Code:

def dayOfTheWeek(d, m, y):

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

from datetime import datetime

return days[datetime(y, m, d).weekday()]

date=input("Enter the date:")

month=input("Enter the month:")

year=input("Enter the year:")

print(dayOfTheWeek(date,month,year))

21.Write a Python program to find the largest palindrome made from the product of two 3-digit numbers.

# Largest Palindrome

n = 0

for a in range(999, 100, -1):

for b in range(a, 100, -1):

x = a \* b

if x > n:

s = str(a \* b)

if s == s[::-1]:

n = a \* b

print(n)

NOTE: IF 4 DIGIT NUMBER CHANGE TO 9999 AND 1000

22. Calculate the number of days, month and year from date of birth to date of joining program

Sol;

from datetime import datetime

from dateutil import relativedelta

# get two dates

d1 = '17/7/1980'

d2 = '16/3/2007'

# convert string to date object

start\_date = datetime.strptime(d1, "%d/%m/%Y")

end\_date = datetime.strptime(d2, "%d/%m/%Y")

# Get the relativedelta between two dates

delta = relativedelta.relativedelta(end\_date, start\_date)

print('Years, Months, Days between two dates is')

print(delta.years, 'Years,', delta.months, 'months,', delta.days, 'days')

# To check for leap year and Years>19

d3=d1.split('/')

d4=d2.split('/')

BY=int(d3[2])

JY=int(d4[2])

if(delta.years>=19 and BY%4==0):

print("I m a lucky adult")

elif delta.years<19:

print("I m aspiring to become adult")

if BY%4==0:

print("Birth year is leap Year")

else:

print("Birth year is not a leap Year")

if JY%4==0:

print("Joining year is leap Year")

else:

print("Joining year is not a leap Year")

23. John has m number of vehicles. He has n children. He agreed with peter to give x number of remaining vehicles after distributing equal and maximum number of vehicles to the children. How many vehicles will peter get? Also if vehicles count is even, Mr.Peter seems so lucky.

Sol:

# Vehicles and children program

M=int(input("Enter the number of vehicles:"))

N=int(input("Enter number of children: "))

x=M%N

if x==0:

print("You are so lucky")

elif x!=0 and x%2==0:

print("Mr.Peter gets", x, "Vehicles. He is lucky")

Sample input and output:

Enter the number of vehicles:10

Enter number of children: 2

You are so lucky

24. Given n non-negative integers a1,a2,a3,…an where each represents a point at coordinate (i, ai) . „ n „ vertical lines are drawn such that the two endpoints of line i is at (i, ai) nd (i,0). Find two lines, which together with x-axis forms a container, such that the container contains the most water. The program should return an integer which corresponds to the maximum area of water that can be contained (maximum area instead of maximum volume sounds weird but this is the 2D plane we are working with for simplicity).

Sol:

def maxArea(A, Len) :

area = 0

for i in range(Len) :

for j in range(i + 1, Len) :

# Calculating the max area

area = max(area, min(A[j], A[i]) \* (j - i))

return area

# Driver code

a = [ 1, 5, 4, 3 ]

len1 = len(a)

print(maxArea(a, len1))

Output:

6

25.You are climbing a staircase. It takes n steps to reach the top. Each time you can either climb 1 or 2 steps. In how many distinct ways can you climb to the top?

Sol:

def fib(n):

if n <= 1:

return n

return fib(n-1) + fib(n-2)

# Driver program

s = int(input("Enter the value of n: "))

print ("Number of ways = ", end="")

print (fib(s+1))

Output:

Enter the value of n: 5

Number of ways = 8

**II. STRING OPERATIONS AND METHODS**

1.Write a program to find the number of special characters in the given statement

Sample Input:

Given statement: Modi Birthday @ September 17, #&$% is the wishes code for him.

Sample Output:

Number of special Characters: 5

#program#

ab=input("enter the string")

a,d,s,spl=0,0,0,0

alpha=[]

digit=[]

spal=[]

for i in range(len(ab)):

if(ab[i].isalpha()):

a+=1

alpha.append(ab[i])

elif(ab[i].isdigit()):

d+=1

digit.append(ab[i])

elif(ab[i].isspace()):

s+=1

else:

spal.append(ab[i])

spl+=1

print(spal,spl)

2. Write a program to print the number of vowels and number of consonants in the given statement and which is maximum?

Sample Input:

Saveetha School of Engineering Sample Output:

Number of vowels = 12 Number of Consonants = 15

Test cases:

1. India is my country

2. All are my brothers and sisters

3. Why dry sky

4. Shy Try Cry

5. EDUCATION

#program#

a=input("enter")

b='aAeEiIoOuU'

v,c=0,0

vo=[]

co=[]

for i in range(len(a)):

if a[i] not in b:

co.append(a[i])

c+=1

else:

vo.append(a[i])

v+=1

print(vo,v)

print(co,c)

3. Program to find whether two strings have same character in same index and returns the number of matches

Sample input:

S1=”what”

S2=”watch”

Sample output:

1

#program#

def match(s1,s2):

c=0

for i in range(min(len(s1),len(s2))):

if(s1[i].lower()==s2[i].lower()):

c+=1

return c

s11=input("enter")

s22=input("enter")

print(match(s11,s22))

4. Program to print number of words in a line and number of lines in a para

Sample input:

'''This is the most straightforward way to count the number

of lines in a text file in Python. The readlines() method reads all

lines from a file and stores it in a list. Next, use the len() function

to find the length of the list which is nothing but total lines present in a file.'''

Sample output:

Number of lines: 3

Number of words in each line:

Line 1 18

Line 2 15

Line 3 22

#program#

a='''Python is a high-level, general-purpose programming language.

Its design philosophy emphasizes code readability.

with the use of significant indentation.'''

b=a.split(".")

b.pop()

print(len(b))

for i in range(len(b)):

words=b[i].split()

print("no of words",len(words))

5. Program to find number of sentences starts with "B"

Sample input:

'''The apple doesn't fall. ...

All that glitters are not gold. ...

A picture is worth a thousand words. ...

Beggers can't be choosers. ...

A bird in the hand. ...

Better safe than sorry. ...

An apple a day keeps doctor away. ...

Blood is thicker than water. ...'''

Sample output:

Total number of lines: 8

Number of Sentences that start with letter B : 3

#program#

str='''The apple doesn't fall. ...

All that glitters are not gold. ...

A picture is worth a thousand words. ...

Beggers can't be choosers. ...

A bird in the hand. ...

Better safe than sorry. ...

An apple a day keeps doctor away. ...

Blood is thicker than water. ...'''

str1=str.split('...')

str1.pop()

c=0

d=[]

print("no of lines",len(str1))

for i in str1:

str2=i.split()

for j in str2:

if j[0]=='B':

c+=1

print("with b",c)

6. Write a program that finds whether a given character is present in a string or not. In case it is present it prints the index at which it is present. Do not use built-in find functions to search the character.

Sample Input:

Enter the string: I am a programmer Enter the character to be searched: p

Sample Output:

P is found in string at index: 8

Note: Check for non-available Character in the given statement as Hidden Test case.

7. Write a program to arrange the letters of the word alphabetically in Normal order and reverse order

Sample Input:

Enter the word: MOSQUE Sample Output:

Alphabetical Order Normal: E M O Q S U Alphabetical Order Reverse: U S Q O M E Test Case:

1. SAPONIFICATION

2. MEMORANDUM

3. DISTRIBUTION

4. SATISFACTION

5. PROPAGATION

#program#

a=input("enter")

b=sorted(a)

print(b)

e="".join(b)

c=e.upper()

d=c[::-1]

print(c)

print(d)

8. Write a program to find the number of letters repeatedly present in the given word and print the Repeated letters.

Sample Input:

Enter the word: TEMPLE Sample Output:

Number of repeated letters = 1 Repeated letter = E

Test Case:

1. HYPOTHECATION

2. MATRICULATION

3. MANIPULATION

4. SIMPLIFICATION

5. DEDICATION

#program#

a=input("enter")

c=0

for i in range(len(a)):

if a[i] in a[i+1:]:

c=c+1

print(c)

9. Write a program to print vowels and consonants from the given word in alphabetical order and print which is maximum, if both vowel count and consonant count is equal then prints Equal?

Sample Input:

Enter the word: EDUCATION Sample Output:

Vowels in alphabetical order: A, E, I, O, U Consonants in alphabetical order: C, D, N, T Maximum Count: Vowels

Test cases:

1. HYPOTHECATION

2. MATRICULATION

3. MANIPULATION

4. SEDIMENTATION

5. EXPERIMENTATION

#program#

a=input("enter")

b='aAeEiIoOuU'

v,c=0,0

vo=[]

co=[]

for i in range(len(a)):

if a[i] not in b:

co.append(a[i])

c+=1

else:

vo.append(a[i])

v+=1

print(vo)

print(co)

If(v==c):

Print(“equal”)

10. Write a program that accepts a string from user and re displays the same string after removing vowels from it.

Sample Input & Output:

Enter a string: we can play the game The string without vowels is: w cn ply th gm

#program#

a=input("enter")

b='aAeEiIoOuU'

v,c=0,0

vo=[]

co=[]

for i in range(len(a)):

if a[i] not in b:

co.append(a[i])

c+=1

else:

continue

d="".join(co)

print(d)

11. Given two strings “s” and “t”, determine if they are isomorphic.

Input: s = "egg", t = "add"

Output: true

#program#

def isisomorphic(str1, str2):

if len(str1) != len(str2):

return False

else:

map1, map2 = {}, {}

for i in range(len(str1)):

ch1, ch2 = str1[i], str2[i]

if ch1 not in map1:

map1[ch1] = ch2

if ch2 not in map2:

map2[ch2] = ch1

if map1[ch1] != ch2 or map2[ch2] != ch1:

return False

return True

str1 = input("String 1=")

str2 = input("String 2=")

print(isisomorphic(str1, str2))

12. Given an integer n, return the number of strings of length n that consist only of vowels

(a, e, i, o, u) and are lexicographically sorted.

Input: n = 2

Output: 15

#program#

def countstrings(n, start):

if n == 0:

return 1

cnt = 0

for i in range(start, 5):

cnt += countstrings(n - 1, i)

return cnt

def countVowelStrings(n):

return countstrings(n, 0)

n = int(input("n="))

print(countVowelStrings(n))

13. Given a string S consisting of N lowercase alphabets, the task is to modify the string S by

replacing each character with the alphabet whose circular distance from the character is equal

to the frequency of the character in S.

Input: S=“ghee”

Output: higg

14. Given two strings S1 and S2, representing sentences, the task is to print both sentences after removing all words which are present in both sentences

Input: S1 = “sky is blue in color”, S2 =”Raj likes sky blue color “

Output: is in

Raj likes

#program#

def words(a,b):

a1=list(a.split())

a2=list(b.split())

for i in a1:

if i in a2:

a1.remove(i)

a2.remove(i)

print(a1)

print(a2)

n=input("enter the string")

m=input("enter the second string")

words(n,m)

15. Given a string s consisting of words and spaces, return the length of the last word in the

string. A word is a maximal substring consisting of non-space characters only.

Test Case:

Input: s = "Hello World"

Output: 5

#program#

a="programming in python"

c=a.split()

print(len(c[-1]))

16. Given a string s and an integer k, return the length of the longest substring of s such that the frequency of each character in this substring is greater than or equal to k.

s consists of only lowercase English letters.

Test cases:

1.Input: s = "aaabb", k = 3

Output: 3

17. Reverse Words in a String

Given an input string s, reverse the order of the words.

Input: s = "the sky is blue"

Output: "blue is sky the"

#program#

a="hello world"

b=a.split()

print("".join(b[::-1]))

18. Raju, has again started troubling people in your city. The people have turned on to you for

getting rid of Raju. Raju presents to you a number consisting of numbers from 0 to 9

characters. He wants you to reverse it from the final answer such that the number becomes

Mirror number. A Mirror is a number which equals its reverse. The hope of people are on you

so you have to solve the riddle. You have to tell if some number exists which you would

reverse to convert the number into Mirror

Sample input:

Enter the number: 123456

Sample output:

Mirror image: 654321

#program#

N=int(input(“enter the number”))

Print(N[::-1])

19. Given an array of strings strs, group the anagrams together. You can return the answer

in any order.

Input: strs = ["eat","tea","tan","ate","nat","bat"]

Output: [["bat"],["nat","tan"],["ate","eat","tea"]]

#program#

def anagrams(l1):

dictionary={}

for word in l1:

sortedword="".join(sorted(word))

print(sortedword)

if sortedword not in dictionary:

dictionary[sortedword]=[word]

else:

dictionary[sortedword]+=word

return [dictionary[i] for i in dictionary]

l1=['pop','bat','tab','opp']

print(anagrams(l1))

20. Program to print first letters of the word in a sentence separated by dot.

Sample input: "The cat on the wall"

Sample output: T.C.O.T.W.

#program#

str1="this is a cat"

str2=str1.split()

for i in str2:

print(i[0].upper(),end=".")

21. Valid Palindrome

A phrase is a palindrome if, after converting all uppercase letters into lowercase letters and

removing all non-alphanumeric characters, it reads the same forward and backward.

Alphanumeric characters include letters and numbers.

Given a string s, return true if it is a palindrome, or false otherwise.

Test Cases:

1.Input: s = "A man, a plan, a canal: Panama"

Output: true

#program#

a="A man, a plan, a canal: Panama"

alpha=[]

digit=[]

for i in range(len(a)):

if (a[i].isalpha()):

alpha.append(a[i])

elif(a[i].isdigit()):

alpha.append(a[i])

e="".join(alpha)

c=e.upper()

b=c[::-1]

if(c==b):

print("palindrome")

else:

print("not")

22. Write a function delchar(s,c) that takes as input strings s and c, where c has length 1 (i.e., a single character), and returns the string obtained by deleting all occurrences of c in s. If c has a length other than 1, the function should return s.

Sample Input:

Enter the string: Hello world

Enter a character to be deleted: l

Sample output:

String after the character is removed: Heo Word

#program#

a=input("enter")

b=input("enter")

co=[]

for i in range(len(a)):

if a[i]==b:

continue

else:

co.append(a[i])

d="".join(co)

print(d)

23. Given a string s consisting of words and spaces, return the length of the last word in the

string. A word is a maximal substring consisting of non-space characters only.

Test Case:

Input: s = "Hello World"

#program#

a="programming in python"

c=a.split()

print(len(c[-1]))

24. Longest Substring with At Least K Repeating Characters

Given a string s and an integer k, return the length of the longest substring of s such that the

frequency of each character in this substring is greater than or equal to k.

s consists of only lowercase English letters.

Solution:

def Substring(s):

ans, temp = 1, 1

for i in range(1, len(s)):

if (s[i] == s[i - 1]):

temp += 1

else:

ans = max(ans, temp)

temp = 1

ans = max(ans, temp)

return ans

s = input("Enter the string: ")

print(Substring(s))

Test cases:

1.Input: s = "aaabb", k = 3

Output: 3

25. Program to accept the strings which contains all vowels.

Input : ABeeIghiObhkUul

Output : Accepted

All vowels are present

SOL:

#All vowels program

def check(string):

vowels = "aeiou" #storing vowels

if all(vowel in string.lower() for vowel in vowels):

print("All vowels are present")

return "Accepted"

return "Not accepted"

#initializing string

string = "ABeeIghiObhkUul"

# test the function

print(check(string))

**III. 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}

Code:

list1=[2,1,2,3,2,1,1,4,3,5,5,6]

list2=set(list1)

print(list(list2))

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}

Code:

import statistics as stats

list1=[16,18,23,16,15,10,19,22,17]

x=stats.mean(list1)

y=stats.median(list1)

z=stats.mode(list1)

print("mean=",x)

print("median=",y)

print("mode=",z)

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

Code:

l=1

u=10

sq=[]

act=[]

for i in range(l,u+1):

square=(i\*i)

sq.append(square)

for i in sq:

temp=i

sum=0

while temp>0:

digit=temp%10

sum=sum+digit

temp=temp//10

if sum<10:

act.append(i)

else:

pass

print(act)

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 = %

Code:

list1=[7,3,4,2,1,8,5,2,1]

list2=sorted(list1)

n=int(input("enter the nth term"))

print(list2[-n])

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

Code:

l=45

u=49

a=[(x,x\*\*2) for x in range(l,u)]

print(a)

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}

Code:

L1=[16,18,27,16,23,21,19]

prime=[]

composite=[]

for num in L1:

if num==2:

prime.append(num)

else:

is\_prime=True

for i in range(2,num):

if num%i==0:

is\_prime=False

composite.append(num)

break

if is\_prime:

prime.append(num)

print("composite are",composite,len(composite))

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}

Code:

n={16,18,27,16,23,21,19}

m=n[::-1]

print(m)

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}

Code:

L1=[26,28,47,26,43,51,29]

prime=[]

composite=[]

for num in L1:

if num==2:

prime.append(num)

else:

is\_prime=True

for i in range(2,num):

if num%i==0:

is\_prime=False

composite.append(num)

break

if is\_prime:

prime.append(num)

print("non composite are",prime,len(prime))

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}

Code:

l1={16,-18,27,-16,23,-21,19}

negative=[]

for num in l1:

if(num<0):

negative.append(num)

print(negative)

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

Code:

n=int(input("enter the nth term"))

m=int(input("enter the mth term"))

list1=[27,12,15,22,13,11,16,25]

list2=list(set(list1))

list3=sorted(list2)

x=list3[-n]

y=list3[m-1]

print("sum",x+y)

print("sub",x-y)

print("mul",x\*y)

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]

Code:

list1=[1,2,4]

list2=[1,3,4]

list3=sorted(list1)

list4=sorted(list2)

print(sorted(list3+list4))

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

Code:

l = [1,2,3,1]

m = max(l)

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

if l[i] == m:

print("Peak Elemeny Index:",i)

break

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

Code:

neg\_avg=0.0

pos\_avg=0.0

pos\_total=0.0

neg\_total=0.0

list1=[]

while True:

number=int(input("enter any number"))

if number==-1:

break

elif number>=100:

break

else:

list1.append(number)

print((list1))

a=0

b=0

for i in range(0,len(list1)):

if(list1[i]>0):

pos\_total=(pos\_total+list1[i])

a=a+1

elif(list1[i]<0):

neg\_total=(neg\_total + list1[i])

b=b+1

pos\_avg=pos\_total/a

neg\_avg=neg\_total/b

print(pos\_total,neg\_total)

print(pos\_avg,neg\_avg)

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]

Code:

l=[]

a=[]

b=[]

n=int(input("Enter size of list "))

for i in range(0,n):

e=int(input("Enter element of list "))

l.append(e)

for i in l:

if(i%2==0):

a.append(i)

else:

b.append(i)

l1=[]

odd\_squ=[sum(x\*\*2 for x in b)]

l1.append(odd\_squ)

even\_squ=[sum(x\*\*2 for x in a)]

l1.append(even\_squ)

print(l1)

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.

Code:

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

for i in range(len(l)):

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

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

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

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]

Code:

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

l1=[]

for i in range(len(l)):

count=0

for j in range(len(l)):

if l[i]>l[j]:

count+=1

l1.append(count)

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.

Code:

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

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

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

T=int(input("T="))

for i in range(0,T):

x[i]=(x[i-1]+E[i])-L[i]

print(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]]

Code:

import itertools

digit=eval(input("enter the number of digits:"))

num=input("enter the number")

while True:

x=True

if(len(num)!=digit):

x=False

print("invalid input,out of limit")

elif digit==2:

if num[0]==num[1]:

print("invalid input,unique permutation")

x=false

elif digit==3:

if (num[0]==num[1]==num[2]):

print("invalid input,unique permutation")

x=False

break

if x==True:

nums=list(num)

permutations=list(itertools.permutations(nums))

p=([''.join(permutation) for permutation in permutations])

p.remove(num)

print(p)

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"]

Code:

n=int(input("enter the value of n"))

p=[]

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

if(i%15==0):

p.append("fizzbuzz")

elif(i%5==0):

p.append("buzz")

elif(i%3==0):

p.append("fizz")

else:

p.append(str(i))

print(p)

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]

Code:

list1=[10,20,20,30,40,40,50]

list2=set(list1)

print(sorted(list(list2)))

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.

Code:

def findmin(A):

min\_ele=A[0];

for i in range(len(A)):

if A[i]<min\_ele:

min\_ele=A[i]

return min\_ele

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

print(findmin(arr))

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]

Code:

def chkPair(A,x):

l=[]

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

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

if (A[i] + A[j] == x):

return [i+1,j+1]

return [-1,-1]

L=[1, 2, 3, 5, 10]

target=int(input("Enter target:"))

print(chkPair(L,target))

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]

Code:

list1=[47,34,21,89,12]

n=int(input("enter the position"))

p=int(input("enter the element"))

list1.insert(n-1,p)

print(list1)

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"

Code:

def dayOfTheWeek(d, m, y):

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

from datetime import datetime

return days[datetime(y, m, d).weekday()]

date=int(input("Enter the date:"))

month=int(input("Enter the month:"))

year=int(input("Enter the year:"))

print(dayOfTheWeek(date,month,year))

25.Write a function shuffle(l1,l2) that takes as input two lists, 11 and l2, and returns a list consisting of the first element in l1, then the first element in l2, then the second element in l2, then the second element in l2, and so on. If the two lists are not of equal length, the remaining elements of the longer list are appended at the end of the shuffled output.

Sol:

def shuffle(l1,l2):

new\_list=[]

n=len(l1)

m=len(l2)

x=min(n,m)

for i in range(x):

new\_list.append(l1[i])

new\_list.append(l2[i])

if len(l1)!=len(l2):

if(n>m):

new\_list.append(l1[x:])

elif (m>n):

new\_list.append(l2[x:])

return new\_list

#Driver Program

list1=[1,3,5]

list2=[2,4,6,8,10]

y=shuffle(list1,list2)

print(\*y)

Output:

[1, 2, 3, 4, 5, 6, [8, 10]]

26. Given two integer arrays nums1 and nums2, return an array of their intersection. Each element in the result must appear as many times as it shows in both arrays and you may return the result in any order.

Example 1:

Input: nums1=[1,2,2,1],nums2=[2,2]

Output:[2,2]

Example2:

Input: nums1=[4,9,5],nums2=[9,4,9,8,4]

Code:

def intersect(nums1: list[int], nums2: list[int]) -> list[int]:

ht = dict()

length1 = len(nums1)

length2 = len(nums2)

longer = []

smaller = []

intersection = []

if length1 > length2:

longer = nums1

smaller = nums2

else:

longer = nums2

smaller = nums1

for i in range(len(longer)):

ht[i] = longer[i]

for j in range(len(smaller)):

if smaller[j] in ht:

intersection.append(smaller[j])

else:

j += 1

return intersection

nums1=[1,2,2,1]

nums2=[2,2]

print(intersect(nums1,nums2))

27. 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]

Code:

def chkPair(A,x):

l=[]

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

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

if (A[i] + A[j] == x):

return [i+1,j+1]

return [-1,-1]

L=[1, 2, 3, 5, 10]

target=int(input("Enter target:"))

print(chkPair(L,target))