1. **Write a python program to sum all the items in the list**

| def sumlist(lst): sumn=0 for x in lst: sumn=sumn+x return sumnprint(sumlist([10,20,30,40])) |
| --- |

Output

100

1. **Write a python program to multiply all the items in a list**

| def multlist(lst): tot=1 for x in lst: tot=tot \* x return totprint(multlist([10,20,30,40])) |
| --- |

Output

240000

1. **Write a Python program to get the largest number from a list.**

| def max\_value(lst): maxv=lst[0] for a in lst: if a>maxv: maxv=a return maxvprint(max\_value([23,56,11,-4,-9,0,47])) |
| --- |

Output

56

1. **Finding consonents and vowels in the given string**

| def countvowels(val): n\_vowel=0 n\_consonent=0 for i in range(len(val)): if val[i] in ['a','e','i','o','u']: n\_vowel+=1 else: n\_consonent+=1 print("number of vowels is =",n\_vowel) print("number of consonents is = ",n\_consonent)x=input("enter the word")countvowels(x) |
| --- |

1. **Find the factorial of a number**

| def factorial(x): fact=1 for i in range(1,x+1): fact=fact\*i return(fact)x=int(input("enter the number"))y=factorial(x)print("the factorial of ",x," is ",y) |
| --- |

### **Python program to convert lower letter to upper and upper letter to lower in a string.**

| input\_str=input("enter a string")x=[]for i in range(len(input\_str)): if input\_str[i].isupper(): x.append(input\_str[i].lower()) elif input\_str[i].islower(): x.append(input\_str[i].upper())for i in range(len(x)): print(x[i],end="") |
| --- |

1. **Search word from a list**

| input\_str=input("enter the string : ")x=input\_str.split()search=input("enter a word to search")if search in input\_str: print("the search string exists")else: print("the search string does not exists")  |
| --- |

Output

enter the string : welcome to python world

enter a word to searchto

the search string exists

### **Write a program in Python to count lower, upper, numeric and special characters in a string.**

| input\_str="@#Welcome To PyThon WorLd 2022&"sum\_num=0sum\_lower=0sum\_upper=0sum\_special=0for i in range(len(input\_str)): if(input\_str[i].isnumeric()): sum\_num+=1 elif(input\_str[i].islower()): sum\_lower+=1 elif(input\_str[i].isupper()): sum\_upper+=1 else: sum\_special+=1print("number counts =",sum\_num)print("lower case count =",sum\_lower)print("upper case count =",sum\_upper)print("special count =",sum\_special)  |
| --- |

Output

number counts = 4

lower case count = 14

upper case count = 6

special count = 7

1. **program to check for palindrome**

| # program to check for palindromestr1=input("enter the string")str1=str1.casefold()rev\_str1=reversed(str1)if list(str1)==list(rev\_str1): print("the string is palindrome")else: print("the string is not a palindrome") |
| --- |

output

enter the string : liril

the string is palindrome

enter the string : welcome

the string is not a palindrome

1. **Matrix reading**

| R=int(input("enter the number of rows : "))C=int(input("enter the number of columns"))print("enter the entries rowwise")mat=[[int(input()) for x in range(C)] for y in range(R)]print(mat)for i in range(R): for j in range(C): print(mat[i][j],end=" ") print() |
| --- |

Output

enter the number of rows : 2

enter the number of columns3

enter the entries rowwise

1

2

3

4

5

6

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

1 2 3

4 5 6

dictionaries

1. **write a program to concatenate dictionaries**

| dicPC={"Dell":10,"Acer":7,"Lenova":17,"HP":11}dicPhone={"Samsung":22,"Apple":9,"Oppo":13}dicTablet={"Samsung":15,"Other":13}dicTotal={}for d in [dicPC,dicTablet,dicPhone]: dicTotal.update(d)print(dicTotal) |
| --- |

output

{'Dell': 10, 'Acer': 7, 'Lenova': 17, 'HP': 11, 'Samsung': 22, 'Other': 13, 'Apple': 9, 'Oppo': 13}

1. **write a program to partitions given dictionary to two subdictionary based on admitted and non-admitted students. We consider the following dictionary whose keys are the names of the students and the key values are the general averages obtained by passing the final exam**:

| students={"student1":13,"student2":17,"student3":9,"student4":16,"student5":15,"student6":20, "student7":10,"student8":7,"student9":9,"student10":4,"student11":15,"student12":20, "student13":10,"student14":6,"student15":19,"student16":12,"student17":10,"student18":9, "student19":9,"student20":15,"student21":9,"student22":11,"student23":14,"student24":10, "student25":13,"student26":17,"student27":10,"student28":12,"student29":11,"student30":17}#create two empty dictionariesadmitted={}notadmitted={}for k,v in students.items(): if(v<10): notadmitted[k]=v else: admitted[k]=vdef displayprint(): print("-"\*50)print("admitted students : ",admitted)displayprint()print("non admitted students : ",notadmitted) |
| --- |

output

admitted students : {'student1': 13, 'student2': 17, 'student4': 16, 'student5': 15, 'student6': 20, 'student7': 10, 'student11': 15, 'student12': 20, 'student13': 10, 'student15': 19, 'student16': 12, 'student17': 10, 'student20': 15, 'student22': 11, 'student23': 14, 'student24': 10, 'student25': 13, 'student26': 17, 'student27': 10, 'student28': 12, 'student29': 11, 'student30': 17}

--------------------------------------------------

non admitted students : {'student3': 9, 'student8': 7, 'student9': 9, 'student10': 4, 'student14': 6, 'student18': 9, 'student19': 9, 'student21': 9}

1. **write a program to create a dictionary with integer and its square**

| n=int(input("enter the value of n"))d={}for i in range(1,n+1): d[i]=i\*iprint(d) |
| --- |

output

enter the value of n7

{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49}

1. **create a dictionary which has key as integer and value as prime number**

| def isPrime(n): n\_divisor=0 for i in range(1,n+1): if(n%i==0): n\_divisor=n\_divisor+1 if (n\_divisor<3): return True else: return Falsed={}for i in range(0,10): n=int(input("type an integer : ")) if(isPrime(n)): d[n]="Prime" else: d[n]="not prime"print(d)  |
| --- |

output

type an integer : 1

type an integer : 4

type an integer : 3

type an integer : 19

type an integer : 17

type an integer : 57

type an integer : 45

type an integer : 79

type an integer : 13

type an integer : 11

{1: 'Prime', 4: 'not prime', 3: 'Prime', 19: 'Prime', 17: 'Prime', 57: 'not prime', 45: 'not prime', 79: 'Prime', 13: 'Prime', 11: 'Prime'}

**15. create a dictionary which has key as integer and value as factorial number**

| def fact(n): f=1 for i in range(1,n+1): f=f\*i return fd={}n=int(input("enter value of integer n "))for i in range(1,n+1): d[i]=fact(i)print("the dictionary is =",d) |
| --- |

output

enter value of integer n 7

the dictionary is = {1: 1, 2: 2, 3: 6, 4: 24, 5: 120, 6: 720, 7: 5040}

**16. create a dictionary which has key as integer and value as their sums (1, 1+2, 1+2+3, ..)**

| def sumInt(n): s=0 for i in range(1,n+1): s=s+i return sd={}n=int(input("enter value for n "))for i in range(1,n): d[i]=sumInt(i)print("the dictionary is = ",d) |
| --- |

output

enter value for n 10

the dictionary is = {1: 1, 2: 3, 3: 6, 4: 10, 5: 15, 6: 21, 7: 28, 8: 36, 9: 45}

**17. Write a Python program that asks the user to enter a text and return a dictionary whose keys are the words of the text entered and the values are the lengths of the words that make up the text. Example for the text T = "Python is a programming language", the program must return the dictionary:**

d = {'Python': 6, 'is': 3, 'a': 3, 'language': 7, 'de': 2, 'programming': 13}

| txt=input("enter the text : ")words=txt.split(' ')d={}for i in words: d[i]=len(i)print("the dictionary is = ",d) |
| --- |

output

enter the text : python is a programming language and is an open source

the dictionary is = {'python': 6, 'is': 2, 'a': 1, 'programming': 11, 'language': 8, 'and': 3, 'an': 2, 'open': 4, 'source': 6}

18. python code to create a dictionary from typed text with word as keys and reverses words as values

| txt=input("enter the text : ")words=txt.split(' ')d={}for i in words: # creating reverse r\_word=i[::-1] d[i]=r\_wordprint("the dictionary is = ",d) |
| --- |

output

enter the text : python is an open source

the dictionary is = {'python': 'nohtyp', 'is': 'si', 'an': 'na', 'open': 'nepo', 'source': 'ecruos'}

**19. Given a dictionary d whose key values are lists. Write a Python program that transforms the dictionary d by sorting the lists. Example for the dictionary:**

d = {'a1': [21, 17, 22, 3], 'a2': [11, 15, 8, 13], 'a3': [7, 13, 2, 11], 'a4': [22,14,7,9]}

The program should return the dictionary:

d = {'a1': [3, 17, 21, 22], 'a2': [8, 11, 13, 15], 'a3': [2, 7, 11, 13], 'a4': [7, 9, 14, 22]}

| d={'a1':[21,17,22,3],'a2':[11,15,8,13],'a3':[7,13,2,11],'a4': [22,14,7,9]}for key,value in d.items(): value.sort() d[key]=valueprint("the sorted dictionary = ",d) |
| --- |

output

the sorted dictionary = {'a1': [3, 17, 21, 22], 'a2': [8, 11, 13, 15], 'a3': [2, 7, 11, 13], 'a4': [7, 9, 14, 22]}

**20. write a python program that verify if a key is present in a given dictionary or not**

| d={'a':13,'b':37,'c':3,'d':45,'e':23,'f':42}def is\_key(key): if key in d: return True else: return Falseprint(is\_key('c'))print(is\_key('g')) |
| --- |

output

True

False

**21. The following dictionary contains a student's math scores:**

scores = {'score1' : 16, 'score2' : 14, 'score3' : 17}

update the notes dictionary by adding the average of scores.

| scores={'score1':16,'score2':14,'score3':20}average=0for key,value in scores.items(): average=average+scores[key]average=average/3scores['average']=averageprint(scores) |
| --- |

{'score1': 16, 'score2': 14, 'score3': 20, 'average': 16.666666666666668}

**22. convert two list into dictionary**

| stname=["anitha","aarthi","banu","bhavana","chitra"]rollno=[100,101,102,103,104]st\_database=dict(zip(stname,rollno))print(st\_database) |
| --- |

output

{'anitha': 100, 'aarthi': 101, 'banu': 102, 'bhavana': 103, 'chitra': 104}

solution 2

| stname=["anitha","aarthi","banu","bhavana","chitra"]rollno=[100,101,102,103,104]st\_database=dict()for i in range(len(stname)): st\_database.update({stname[i]:rollno[i]})print(st\_database) |
| --- |

output

{'anitha': 100, 'aarthi': 101, 'banu': 102, 'bhavana': 103, 'chitra': 104}

**23. merge two dictionary**

| dict1={'Ten':10,'Twenty':20,'Thirty':30}dict2={'Fourty':40,'Fifty':50,'SIxty':60}dict3={\*\*dict1,\*\*dict2}print(dict3) |
| --- |

output

{'Ten': 10, 'Twenty': 20, 'Thirty': 30, 'Fourty': 40, 'Fifty': 50, 'SIxty': 60}

solution 2

| dict1={'Ten':10,'Twenty':20,'Thirty':30}dict2={'Fourty':40,'Fifty':50,'SIxty':60}dict3=dict1.copy()dict3.update(dict2)print(dict3) |
| --- |

output

{'Ten': 10, 'Twenty': 20, 'Thirty': 30, 'Fourty': 40, 'Fifty': 50, 'SIxty': 60}

<https://www.w3resource.com/python-exercises/math/>

**24. write a program to display a matrix**

| mat1=[[1,2,3,4],[5,6,7,8],[9,10,11,12]]for i in range(len(mat1)): for j in range(len(mat1[0])): print(mat1[i][j],end=" ") print("\n") |
| --- |

output

1 2 3 4

5 6 7 8

9 10 11 12

**25 Program to do matrix addition**

| print("enter the value for first matrix")R1=int(input("enter number of row : "))C1=int(input("enter number of column : ")) mat1=[]for i in range(R1):  a=[] for j in range(C1):  a.append(int(input())) mat1.append(a)print(mat1)print("enter the value for second matrix")R2=int(input("enter number of row : "))C2=int(input("enter number of column : ")) mat2=[]for i in range(R2):  a=[] for j in range(C2):  a.append(int(input())) mat2.append(a)print(mat2)# add the matrixmat3=[]for i in range(R1): a=[] for j in range(C1): a.append(mat1[i][j]+mat2[i][j]) mat3.append(a)print("the added matrix is")print(mat3)# another way to print matrixfor i in range(len(mat3)): for j in range(len(mat3[0])): print(mat3[i][j],end=" ") print("\n") |
| --- |

output

enter the value for first matrix

enter number of row : 2

enter number of column : 3

1

2

3

4

5

6

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

enter the value for second matrix

enter number of row : 2

enter number of column : 3

7

8

9

10

11

12

[[7, 8, 9], [10, 11, 12]]

the added matrix is

[[8, 10, 12], [14, 16, 18]]

8 10 12

14 16 18

**26. program to add matrix using single line input**

| R1=int(input("enter the no of rows for first matrix "))C1=int(input("enter the no of columns for the first matrix"))mat1=[[int(input())for j in range(C1)]for i in range(R1)]print("second matrix")R2=int(input("enter the no of rows for first matrix "))C2=int(input("enter the no of columns for the first matrix"))mat2=[[int(input())for j in range(C2)]for i in range(R2)]print(mat1)print(mat2)mat3=[[mat1[i][j]+mat2[i][j] for j in range(C2) ] for i in range(R2)]print(mat3) |
| --- |

output

enter the no of rows for first matrix 2

enter the no of columns for the first matrix3

1

2

3

4

5

6

second matrix

enter the no of rows for first matrix 2

enter the no of columns for the first matrix3

7

8

9

1

1

3

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

[[7, 8, 9], [1, 1, 3]]

[[8, 10, 12], [5, 6, 9]]

**27. write a program to find the transpose of a matrix**

| R1=int(input("enter the no of rows of matrix "))C1=int(input("enter the no of columns of matrix"))mat1=[[int(input())for j in range(C1)]for i in range(R1)] mat2=[[mat1[j][i] for j in range(len(mat1))] for i in range(len(mat1[0]))] print ("original matrix is")for i in range(len(mat1)):  for j in range(len(mat1[0])):  print(mat1[i][j],end=" ")  print() print(" transposed matrix is ")for i in range(len(mat2)):  for j in range(len(mat2[0])):  print(mat2[i][j],end=" ")  print() |
| --- |

output

enter the no of rows of matrix 3

enter the no of columns of matrix3

1

2

3

4

5

6

7

8

9

original matrix is

1 2 3

4 5 6

7 8 9

 transposed matrix is

1 4 7

2 5 8

3 6 9

**28. write a program to find multiplication of two matrix**

| print("enter the value for first matrix")R1=int(input("enter number of row : "))C1=int(input("enter number of column : ")) mat1=[]for i in range(R1):  a=[] for j in range(C1):  a.append(int(input())) mat1.append(a)print(mat1)print("enter the value for second matrix")R2=int(input("enter number of row : "))C2=int(input("enter number of column : ")) mat2=[]for i in range(R2):  a=[] for j in range(C2):  a.append(int(input())) mat2.append(a)print(mat2)# add the matrixmat3=[[0 for j in range(len(mat2[0]))] for i in range(len(mat1))] #iterate through rows of mat1for i in range(len(mat1)): #iterate through columns of mat2 for j in range(len(mat2[0])): #iterate through rows of mat2 for k in range(len(mat2)): mat3[i][j]=mat3[i][j]+mat1[i][k]\*mat2[k][j]  print("the multipled matrix result is")# another way to print matrixfor i in range(len(mat3)): for j in range(len(mat3[0])): print(mat3[i][j],end=" ") print("\n") |
| --- |

output

enter the value for first matrix

enter number of row : 3

enter number of column : 3

12

7

3

4

5

6

7

8

9

[[12, 7, 3], [4, 5, 6], [7, 8, 9]]

enter the value for second matrix

enter number of row : 3

enter number of column : 4

5

8

1

2

6

7

3

0

4

5

9

1

[[5, 8, 1, 2], [6, 7, 3, 0], [4, 5, 9, 1]]

the multipled matrix result is

114 160 60 27

74 97 73 14

119 157 112 23

**programs using Lambda functions**

**29. Write a Python program to create a lambda function that adds 15 to a given number passed in as an argument, also create a lambda function that multiplies argument x with argument y and print the result.**

| r=lambda x:x+15print(r(5))r1=lambda x,y:x\*yprint(r1(5,10)) |
| --- |

output

20

50

**30. Write a Python program to create a function that takes one argument, and that argument will be multiplied with an unknown given number**.

| def func\_compute(n): return lambda x:x\*nresult=func\_compute(2)print("double the number of 15 = ",result(15))result=func\_compute(3)print("Triple the number of 15 = ",result(15))result=func\_compute(4)print("Quadruple the number of 15 = ",result(15))result=func\_compute(5)print("Quintuple the number of 15 = ",result(15)) |
| --- |

output

double the number of 15 = 30

Triple the number of 15 = 45

Quadruple the number of 15 = 60

Quintuple the number of 15 = 75

**31. Write a Python program to sort a list of tuples using Lambda.**

| tuple1=[('english',88),('science ',90),('maths',55),('physics',34),('chemistry',67)]print("original values is ")print(tuple1)tuple1.sort(key=lambda x:x[1])print(tuple1)tuple1.sort(key=lambda x:x[0])print(tuple1) |
| --- |

output

original values is

[('english', 88), ('science ', 90), ('maths', 55), ('physics', 34), ('chemistry', 67)]

[('physics', 34), ('maths', 55), ('chemistry', 67), ('english', 88), ('science ', 90)]

[('chemistry', 67), ('english', 88), ('maths', 55), ('physics', 34), ('science ', 90)]

32 Write a Python program to sort a list of dictionaries using Lambda.

| student=[{'name':'banu','science ':90},{'name':'anitha','english':88},{'name':'gayathri','chemistry': 67},{'name':'danush','physics':45},{'name':'chitra','maths':55}]print("original values is ")print(student)student\_sort=sorted(student,key=lambda x:x['name'])print("sorted dictionary is ")print(student\_sort) |
| --- |

output

original values is

[{'name': 'banu', 'science ': 90}, {'name': 'anitha', 'english': 88}, {'name': 'gayathri', 'chemistry': 67}, {'name': 'danush', 'physics': 45}, {'name': 'chitra', 'maths': 55}]

sorted dictionary is

[{'name': 'anitha', 'english': 88}, {'name': 'banu', 'science ': 90}, {'name': 'chitra', 'maths': 55}, {'name': 'danush', 'physics': 45}, {'name': 'gayathri', 'chemistry': 67}]

32. Write a Python program to filter a list of integers using Lambda.

| nums=[1,2,3,4,5,6,7,8,9,10]print("original list")print(nums)print("even numbered list is")even\_nums=list(filter(lambda x:x%2==0,nums))print(even\_nums)print("odd numbered list is")odd\_nums=list(filter(lambda x:x%2!=0,nums))print(odd\_nums) |
| --- |

output

original list

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

even numbered list is

[2, 4, 6, 8, 10]

odd numbered list is

[1, 3, 5, 7, 9]

33. Write a Python program to square and cube every number in a given list of integers using Lambda.

| nums=[1,2,3,4,5,6,7,8,9,10]print("original list")print(nums)print("square numbered list is")square\_nums=list(map(lambda x:x\*x,nums))print(square\_nums)print("cube numbered list is")cube\_nums=list(map(lambda x:x\*x\*x,nums))print(cube\_nums) |
| --- |

output

original list

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

square numbered list is

[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]

cube numbered list is

[1, 8, 27, 64, 125, 216, 343, 512, 729, 1000]

34. Write a Python program to find whether a given string starts with a given character using Lambda.

| startwith=lambda x: True if x.startswith('P') else Falseprint(startwith("Python"))print(startwith("java")) |
| --- |

output

True

False

35. Write a Python program to count the even, odd numbers in a given array of integers using Lambda.

| nums=[1,2,3,4,5,6,7,8,9,10,15,29,44,35]print("original list")print(nums)print("even numbered count is")even\_count=len(list(filter(lambda x:x%2==0,nums)))print(even\_count)print("odd numbered count is")odd\_count=len(list(filter(lambda x:x%2!=0,nums)))print(odd\_count) |
| --- |

output

original list

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 29, 44, 35]

even numbered count is

6

odd numbered count is

8

36. Write a Python program to calculate the sum of the positive and negative numbers of a given list of numbers using lambda function.

| nums=[2,4,10.-5,-12,6,-34,20,-7]print("original list: ",nums)neg\_list=list(filter(lambda x:x<0,nums))pos\_list=list(filter(lambda x:x>0,nums))print("sum of positive numbers = ",sum(pos\_list))print("sum of negative numbers = ",sum(neg\_list)) |
| --- |

output

original list: [2, 4, 5.0, -12, 6, -34, 20, -7]

sum of positive numbers = 37.0

sum of negative numbers = -53

37. write a program to swap two values .

| a,b=10,20print("before swapping ")print("a= ",a)print("b= ",b)a,b=b,aprint("after swapping ")print("a= ",a)print("b= ",b) |
| --- |

output

before swapping

a= 10

b= 20

after swapping

a= 20

b= 10

**classes and objects**

38. write a class called Number which maintains an integer. it should have the following methods in it to perform various operations on the integer

set\_number(self,n)

get\_number(self)

print\_number(self)

isnegative(self)

isdivisibleby(self,n)

absolute\_value(self)

| class Number: def set\_number(self,n): self.\_\_num=n def get\_number(self): return self.\_\_num def print\_number(self): print(self.\_\_num) def isnegative(self): if self.\_\_num<0: return True else: return False def isdivisibleby(self,n): if n==0: return False if self.\_\_num%n==0: return True else: return False def absolute\_value(self): if self.\_\_num<0: return -(self.\_\_num) else: return self.\_\_numx=Number()x.set\_number(-12)x.print\_number()if x.isdivisibleby(5)==True: print("5 divides",x.get\_number())else: print("5 does not divide",x.get\_number())print("absolute value of ",x.get\_number(),"is",x.absolute\_value()) |
| --- |

**output**

-12

5 does not divide -12

absolute value of -12 is 12

when the value is given is -125

output

-125

5 divides -125

absolute value of -125 is 125

 39. Create a class called complex number and write a program to print complex number

| #complex number classclass complexnumber: def \_\_init\_\_(self,r,i): self.real=r self.img=i def get\_data(self): print(f'{self.real}+{self.img}j')num1=complexnumber(2,3)num1.get\_data()num2=complexnumber(5,3)num2.get\_data()real1=num1.real+num2.realimg1=num1.img+num2.imgprint(f'{real1}+{img1}j') |
| --- |

output

2+3j

5+3j

7+6j