pa

List Duplicates

```
a=[1,2,3,4,5,2,3,4,7,9,5]
b=[]
for i in a:
   if i not in b:
      b.append(i)
   else:
      print(i,end=" ")

# print(b)
```

2. find the unique characters in given string :

```
a="bdbac353@#2c&191#"
unique_char=[]
for character in a:
   if character not in unique_char:
      unique_char.append(character.lower())
   else:
      unique_char.remove(character.lower())
print(format(unique_char))
```

show only duplicates output in list

```
list = [10, 20, 30, 40, 50, 10, 20, 10]
new = []
for i in list:
 n = list.count(i)
 if n > 1:
  if new.count(i) == 0:
   new.append(i)
print(new)
```

3. insertion sort

```
n=[1,35,6,3,7,3,8,2,0]
for i in range(len(n)):
   for j in range(i+1,len(n)):
      if n[i]<n[j]:
         n[i],n[j]=n[j],n[i]
         print(n)
```

11.Bubble sorting :-

```
def bubble_sort(array):
   n=len(array)
   for i in range(n):
      for j in range(n-i-1):
         if array[j]>array[j+1]:
            array[j],array[j+1]=array[j+1],array[j]
array=[2,4,7,4,8,9,3]
bubble_sort(array)
print(array)
```

odd & even separation :-

```
list1 = [3, 5, 4, 9, 8, 5, 7, 8, 12]
odd = []
even = []
j = 0
for i in list1:
   if list1[j] % 2 == 0:
      even.append(i)
   else:
      odd.append(i)
   j = j + 1
print(even)
print(odd)
```

5. #second largest number of list :-

```
list = [20, 30, 40, 25, 10]
list_val.sort()
print("The second large number :", list[-2])
```

6. print the elements of an array :-

```
arr = [1, 2, 3, 4, 5,6,7]
 # Loop through the array by incrementing the value
of i
for i in range(0, len(arr)):
   print(arr[i],end='')
```

Even at front and odd at back

```
l = [1,6, 2, 3, 8, 7, 4]
j = 0
for item in l:
 if l[j] % 2 == 0:
 l.append(item)
 l.remove(item)
 j = j+1
print(l)
```

7. elements of an array in reverse order

```
arr = [1, 2, 3, 4, 5];
for i in range(0, len(arr)):
   print(arr[i],end='')
print("\nreversal array:  ");
for i in range(len(arr)-1,-1,-1):
   print(arr[i],end='')
```

Merge two dictionaries

```
dict1 = {'a': 10, 'b': 8}
dict2 = {'c': 6, 'd': 4}
def Merge(dict1, dict2):
 return(dict1.update(dict2)) # using update method
Merge(dict1, dict2)
print(dict1)

dict1 = {'a': 10, 'b': 8}
dict2 = {'c': 6, 'd': 4}
print(dict(dict1 | dict2)) #using union
```

Duplicate words from sentence

```
s = ["I am very happy"]
word = ' '.join([str(elem) for elem in s])
words = word.split(" ")
for i in range(0, len(words)):
   count = 1
   for j in range(i+1, len(words)):
      if(words[i] == (words[j])):
         count = count + 1
         words[j] = "0"
   if(count > 1 and words[i] != "0"):
      print(words[i])
```

9. Lower to upper by using decorator :-

```
def uppercase(fun):
   def abc():
      text = fun()
      modified = text.upper()
      return modified
   return abc

@uppercase
def gen_message():
   return 'Hello my name is'
msg = gen_message()
print(msg)
```

10.add update insert, in dictionary :-

```
a={1:'zttz',2:'aaaa'}
a.update({3:'zzzz'})
print(a)

a=["a_1","b_2","c_3"]
dict={}
for item in a:
   var=item.split('_')
   dict[var[0]]=int(var[1])
print(dict)
 output={'a': 1, 'b': 2, 'c': 3}
```

duplicates output in dictionary format :-

```
a = [10, 11, 12, 15, 20, 21]
b = []
res = {}
for i in a:
   res[i] = a.count(i)
   if i not in b:
      b.append(i)
   else:
      print(i)
print(list(b))
print(res)
```

12.int and string separation :-

```
a = "MH10CD7172"
num  =  ""
word = ""
for c in a:
   if c.isdigit():
      num = num + c
   else:
      word = word + c
print(num)
print(word)
```

14.legnth of without space string

```
a = 'indexial solution is the best'
 # printing original string
print("original length is : " + str(len(a)))
 # isspace() checks for space
result = sum(not chr.isspace() for chr in a)
 # printing result
print("without space length is : " + str(result))
```

Reverse list using recursion function

```
list1 = [1, 2, 3, 4, 5]
def reverse_fun(numbers):
if len(numbers) == 1:
   return numbers
# Otherwise
return
reverse_fun(numbers[1:])+numbers[0:1]
print(reverse_fun(list1))
```

13.To print even length words in string

```
n="This is a python language"
#splitting the words in a given string
s=n.split(" ")
for i in s:
  #checking the length of words
  if len(i)%2==0:
    print(i)
```

4. addition of two list and output dictionary:-

```
a=[3,7,8,4,2,8,9]
b=[3456,7356,5768588,4363,35632,83536,2329]
ab={a[i]:b[i] for i in range(len(a))}
print(ab)
```

16.Counter function in string/array

```
from collections import Counter
a=[1,2,3,4,2,5,2,5,2]
#b=a.count(2,5)
print(Counter(a))
```

Find the middle of list

```
def findMiddle(list1):
    middle = float(len(list1))/2
    if middle % 2 != 0:
     return list1[int(middle - .5)]
    else:
     return (list1[int(middle-1)],
list1[int(middle)])
list1 = [4, 3, 2, 7, 10, 44, 22]
print(findMiddle(list1))
```

15.key value separate in dicationary

```
my_dict = {"one": 1,"two":2,"three":3,"four":4}

1. for item in my_dict:
    print("Key : {} , Value :
{}".format(item,my_dict[item]))

2. for k in my_dict:
print(k)

3. keys= my_dict.keys()
4.values= my_dict.value()
5.items= my_dict.item()
```

super() – provides parent class functionality

```
class Parent:
    def__init_(self, txt):
        self.message = txt
    def printmessage(self):
        print(self.message)
class Child(Parent):
    def__init_(self, txt):
        super()._init_(txt)
x = Child("Hello, and welcome!")
x.printmessage()
```

Each vowel in string

```
def Check_Vow(string, vowels):
    string = string.casefold()
    count = {}.fromkeys(vowels, 0)
    for character in string:
        if character in count:
            count[character] += 1
    return count
vowels = 'aeiou'
string = "Hi, i love eating ice cream and junk
food"
print (Check_Vow(string, vowels))
```

17.Factorial Numbers

```
def factorial(n):
    if n==0:
        result=1
    else:
        result=n*factorial(n-1)
    return result
print(factorial(2))
print(factorial(3))
print(factorial(4))
print(factorial(0))
print(factorial(5))
```

18. Instance variable / self /__init___:-

```
class student:
    def___init_(self,name,age,marks):
        self.name=name
        self.age=age
        self.marks=marks

s1=student("xyz",20,77)
print('object 1')
print('name:',s1.name)
print('age:',s1.age)
print('marks:',s1.marks)
```

Overriding method

```
class Parent():
    def__init__(self):
        self.value = "Inside Parent"
    def show(self):
        print(self.value)
class Child(Parent):
    def___init__(self):
        self.value = "Inside Child"
    def show(self):
        print(self.value)
obj1 = Parent()
obj2 = Child()
obj1.show()
obj2.show()
```

Map

```
def func(a):
    return a * a
x = map(func, (1,2,3,4))
print(list(x))
```

Filter

```
def func(x):
 if x>=3:
  return x
y = filter(func, (1,2,3,4))
print(list(y))
```

Reduce

```
from functools import reduce
x = reduce(lambda a,b:
a+b,[23,21,45,98])
print(x)
```

First non-repeated character from the string

```
s = "calculate"
while s != "":
 slen0 = len(s)
 ch = s[0]
 s = s.replace(ch, "")
 slen1 = len(s)
 if slen1 == slen0 - 1:
  print ("First non-repeating character is: ",ch)
  break
```

Lambda

```
def myfunc(n):
 return lambda a : a * n
x = myfunc(2)
y = myfunc(3)
print(x(11))
print(y(11))

x = lambda a, b: a * b
print(x(5, 6))

x = lambda a, b, c : a + b + c
print(x(5, 6, 2))
```

Reverse string

```
s = 'Pradip'
print(s[::-1])

def reverse(s):
    str = ' '
    for i in s:
        str = i + str
    return str
s = 'Pradip'
print(reverse(s))
```

Program for pattern (1-15)

```
num = 5
for row in range(num):
 val = row + 1
 dec = num - 1
 for col in range (val):
  print(val, end = "")
  val = val + dec
  dec = dec - 1
 print()
```

Alternative numbers :

```
list = [1,2,3,4,5]
alternate = list[::2]
for item in alternate:
print(item)

numbers = [11,13,15,16,17]
# finding alternate elements
result = [numbers[i] for i in
range(len(numbers)) if i % 2 != 0]
# printing the result
print(result)
```

Min number of list

```
list = [1,2,3,4,5,6]
print(min(list))
list = [1,2,3,4,5,6]
min = list[0]
for i in list:
if i < min:
min = i
print(min)
```

Capitalize first character in each word

```
list1 = ["Hello", "how", "are", "you"]
for i in range(len(list1)):
 list1[i] = list1[i].capitalize()
print(list1)
```

ASCII values

```
Character=input('enter the chaaracter')
A=ord(Character)
Print(A)

Character=input('enter the ASCII value')
B=chr(Character)
Print(B)
```

Fibonacci series :-

```
def fib(n):
    a = 1
    b = 0
    for i in range(n):
     print(b)
     a, b = b, b + a
fib(20)
```

8. sum of all elements in an array :-

```
arr = [1, 2, 3, 4, 5]
sum = 0
# Loop through the array to calculate sum of
elements
for i in range(0, len(arr)):
   sum = sum + arr[i]
print("Sum : " + str(sum))
```

Prime or composite number

```
num=int(input('enter the number'))
count=0
i=1
while i<=num:
   if num%i==0:
      count=count+1
   i=i+1
if  count==2:
   print('prime number')
elif count>2:
   print('composite number')
```

Average of overall list

```
n=int(input('enter the range'))
l=[]
for i in range(n):
   ele=int(input('enter the element'))
   l.append(ele)
print(l)
addition=sum(l)
print(addition)
average=addition/n
print(average)
```

Print dict1 keys and dict2 values in one *dictionary*

```
dict1 = {'a': 10, 'b': 8}
dict2 = {'c': 6, 'd': 4}
output=(dict(zip(dict1.keys(),dict2.values())))
```

Squre of each element with list comprehension

```
numbers = [1, 2, 3, 4, 5]
squared = [number ** 2 for number in
numbers]
print(squared)
```

Find only tuple first element

```
a = [(3,4),(1,2),(5,6)]
c=sorted(a)
print(c)
b = [(x[0]) for x in c]
print(b)
```

Nearest number of 0 :

```
def closest(lst, K):
   return lst[min(range(len(lst)), key = lambda i:
abs(lst[i]-K))]
# Driver code
lst = [3,4,-8,3,2,-1,5]
K = 0
print(closest(lst, K))
```

Reversing a numberse

```
num = 123
reversed_num = 0

while num != 0:
   digit = num % 10
   reversed_num = reversed_num * 10 + digit
   num //= 10
print("Reversed Number: " + str(reversed_num))
```

String palindrom or not

```
my_str = '.....'
rev_str = reversed(my_str)
# check if the string is equal to its reverse
if list(my_str) == list(rev_str):
  print("palindrome.")
else:
  print("not palindrome.")
```

#Odd Number Square using Lambda

```
l1 = [4, 2, 13, 21, 5]
l2 = list(map(lambda v: v ** 2, filter(lambda
u: u % 2==1, l1)))
print(l2)
```

Divided by zero exception handling with Decorator

```
def Div_by_zero(func):
  def inner(x,y):
    if y ==0:
      return "devided by is zero"
    return func(x,y)
  return inner
@Div_by_zero
def Unitprice (Amount,Quantity):
  return Amount / Quantity

 # Main Program
print (Unitprice(500,12))
```

String ends with character

```
list = ['akash','akki',]
results = [string for string in list if
string.endswith("h")]

print(results)
```

Duplicate Word

```
s = "Python is very easy easy"
l = s.split(" ")
l2 = []
for j in range(0,len(l)):
    for i in range(j+1,len(l)):
        if l[j] == l[i]:
            l2.append(l[j])
        #  print(l[j],end=" ")
print(l2)

s1 = "Akash is a good boy :"
l = s1.find('is')
print(l)
```

String duplicate remove

```
string="akash"
str=""
for char in string:
    if char not in str:
        str=str+char
print(str)
list =list("akash")
print(list)
```

Word Occurance

```
s1 = 'here and there'
s3 = 'here'
def count_substring(s1,s2):
    len1 = len(s1)
    len2 = len(s2)
    j=0
    counter =0
    while(j < len1):
        if(s1[j]==s2[0]):
            if(s1[j:j+len2]==s2):
                counter = counter +1
        j=j+1
    return counter
a = count_substring(s1,s3)
print(a)
```

String Occurrence

```
a = 'quhkjhsjklvsnk'
b = {}
for i in a:
    if i in b.keys():
        b[i] = b[i] + 1
    else:
        b[i] = 1
print(b)
```

Digit Separate

```
inp_str = "Python4Journaldev"

print("Original String : " + inp_str)
num = ""
for c in inp_str:
    if c.isdigit():
        num = num + c
print("Extracted numbers from the list : " + num)
```

String to Digit

```
list = ['1','33','33','22','24','252']
for i in range(len(list)):
    for j in range(i+1,len(list)):
        if int(list[i])>int(list[j]):
            list[i],list[j]=list[j],list[i]
print(list)
```

String Reverse

```
# s = input("Enter the string : ")
# l = s.split()
# j = -1
# list = []
# for item in l:
#     list.insert(j,item)
#     j =j-1
# output =' '.join(list)
# print(output)
```

word Reverse

```
s = "python is very easy"
str = []
s1 = s.split(" ")
for i in s1:
    str.insert(0,i)
print(str)
```

Large nested list

```
l = [[1,44,33],[52,66,44],[10,11,10]]
list = []
for item in l:
    item.sort(reverse = True)
    for j in range(len(item)):
        if j not in list:
            list.append(item[0])
        if j == 0:
            break
#list.sort()
print(list)
```

Second Largest Number in list

```
list = [12,43,42,78,54,3,22]
for i in range(len(list)):
    for j in range(i+1,len(list)):
        if list[i]>list[j]:
            list[i],list[j]=list[j],list[i]
sec = len(list)-2
print(list[sec])
```

Palindrome

```
string = "12321"
j = -1
for i in range(len(string)):
    if i != len(string):
        if string[i] != string[j]:
            print("string is not palindrome")
            break
        else:
            j = j - 1
            if j == -len(string):
              print("string is palindrome")
```

Pyramid

```
rows = int(input("Enter the rows"))
for i in range(0,rows+1):
    for j in range(i):
        print(i,end='')
    print()
# * pyramid
rows = int(input("Enter the rows"))
for i in range(0,rows+1):
    for j in range(i):
        print("*",end='')
 print()
```

Ascending order

```
my_list = [2,3,5,7,9,0,4,8,9]
new_list = []
while my_list:
    min = my_list[0]
    for x in my_list:
        if x < min:     # > use greater than for
descending order
            min = x
    new_list.append(min)
    my_list.remove(min)

print(new_list)
```

Count Check vowels

```
def Check_Vow(string, vowels):
    string = string.casefold()
    count = {}.fromkeys(vowels, 0)

    for item in string:
        if item in count:
            count[item] += 1
    return count

vowels = 'aeiouAEIOU'
string = "Hi, I love eating ice cream and junk
food"
print (Check_Vow(string, vowels))
```

List Duplicate

```
l = [1,2,4,4,5,3,2]
l2 = []
for j in range(0,len(l)):
    for i in range(j+1,len(l)):
        if l[j] == l[i]:
            l2.append(l[j])
            #print(j,end=" ")
 print(l2)
```

 # print first non repeating character

```
def first_non_repeating_char(str1):
    char = ['calculate']
    ctr = {}
    for c in str1:
        if c in ctr:
            ctr[c]+=1
        else:
            ctr[c]=1
            char.append(c)
        for c in char:
            if ctr[c]==1:
                return c
        return None
```

Odd Even Alternet

```
s = "one two three four five six"
l = s.split(" ")
l1 = []
for item in range(0,len(l),2):
    j = item+1
    l1.append(l[j])
    l1.append(l[item])
s = " ".join(l1)
print(s)
```

Vowels Occurance From giver String

```
word=input("Enter any word: ")
vowels={'a','e','i','o','u'}
d={}
for x in word:
    if x in vowels:
        d[x]=d.get(x,0)+1
for k,v in sorted(d.items()):
      print(k,"occurred ",v," times")
```

Factorial Number

```
def fact(num):
    result=1
    while num>=1:
        result=result*num
        num=num-1 ,k
     return result
     for i in range(1,5):
        print("The Factorial of",i,"is
:",fact(i))
```

Factorial Number

```
def factorial(n):
      if n==0:
          result=1
      else:
           result=n*factorial(n-1)
      return result
print("Factorial of 4 is :",factorial(4))
print("Factorial of 5 is :",factorial(5))
```

Greater Number using Lambda

```
S = lambda a,b:a if a>b else b
Print("The biggest number is 10,20",S
(10,20))
```

Two List converted in to Dictionary

```
s1 = [1,2,3,4]
s2 = ["a","b","c","d"]
dic = {}
for k in s1:
    for v in s2:
        dic[k] = v
        s2.remove(v)
        break
print(dic)
```

List Duplicate Shows

```
n=[1,2,3,1,2,4,5]
l = []
for i in range(len(n)):
   for j in range(i+1,len(n)):
      if n[i]==n[j]:
         l.append(n[j])
print(l)
```

List first repeated character

```
def first_repeated(list):
    set1 = set()
    no_rep = 0
    for i in range(len(list)):
       if list[i] in set1:
          return list[i]
       else:
          set1.add(list[i])
    return no_rep
print(first_repeated([1, 2, 3, 4]))
```

Indexing to the list element

```
languages = ['Python', 'Java',
'JavaScript']

enumerate_prime =
enumerate(languages)
# convert enumerate object to list
print(list(enumerate_prime))

# Output: [(0, 'Python'), (1, 'Java'), (2,
'JavaScript')]
```

List Digit Sring Seprate re module isinstance

```
list = [4,'a','b',3,2,'d',6]
list_digit = [x for x in list if isinstance(x,
int)]
print(list_digit)
list_str = [x for x in list if isinstance(x, str)]
print(list_str)
################################
```

Sting = A3b2c3

```
string = "a4b3c2"
output = ""
for i in string:
    if i.isalpha():
        output = output + i
        previous = i
    else:
        output = output +
previous*(int(i)-1)
print(output)
```

List Digit String Seprate without in built

```
list = ['a',1,2,'b']
num = []
word = []
for i in list:
    if type(i) == str:
        word.append(i)
    else:
        num.append(i)
print(num)
print(word)
```

f Stirng uses

```
a = 10
b = 20
print(f" value a is {a} and value b is {b}")
```

Bubble sorting array

```
list = [14,46,43,27,57,41,45,21,70]
for item in range(len(list)-1,0,-1):
    for i in range(item):
        if list[i]>list[i+1]:
            temp = list[i]
            list[i] = list[i+1]
            list[i+1] = temp
print(list)
```

