```
#1
#Input  - [1,2,3,4]
#Output - [1,4,9.16]

number = [1,2,3,4]

def square(num):
   return num ** 2
op = []

for n in number:
   res = square(n)
   op.append(res)
print(op) #[1,4,9.16]

#Using Lambda

square1 = lambda num: num ** 2

op = []

for n in number:
   res = square1(n)
   op.append(res)
print(op)#[1,4,9.16]

#Using Map
l = [1,2,3,4]
res1 = map(square1, l)
```

```
print(list(res1))#[1,4,9.16]



#**********************************************************************************
******************
#2
#Create a list strating with names of vowels
names = ['steve','eve','alex','john','alexa']

op1 = []

for name in names:
   if name[0].lower() in 'aeiou':
      print(op1.append(name))
print(op1)

names = ['steve','eve','alex','john','alexa']

vowels = lambda name : name[0].lower() in 'aeiou'
print(list(filter(vowels,names)))

#**********************************************************************************
******************
#3
#  write a program to merge two list ?
l1 = [1, 2, 3]
l2 = [4, 5, 6]
print(l1 + l2)
l3 = [*l1, *l2]
```

```
print(l3)


#4
#  WAP to get the elements that are in list b not in list a ?


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


for item in b :
    if item not in a:
        print(item, end= ' ')
print() #6 7 9


#5
# WAP to built a list with only the items having even number of character ?


names = ['amazon', 'gmail', 'yahoo', 'walmart', 'flipkart', 'rediff']
new = []
for item in names:
    if len(item)%2 == 0:
        new.append(item)
print(new) #['amazon', 'flipkart', 'rediff']


#6
#Wap to print all the maximum numbers present in the list python


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


max_num = max(numbers)
```

```
print(max_num )


#7
#Wap to print all the maximum words present in the list python
# Program - 1st

words = ['apple', 'walmart', 'flipkart', 'flipkart', 'apple']

max_len = max(words, key=len)

for word in words:
   if len(max_len) == len(word):
      print(word)

#8
# Python code to demonstrate
# sum of list of list using
# zip and list comprehension

# Declaring initial list of list
List = [[1, 2, 3],
      [4, 5, 6],
      [7, 8, 9]]

# Printing list of list
print("Initial List - ", str(List))

# Using list comprehension
res = [sum(i) for i in zip(*List)]
```

```
# printing result
print("final list - ", str(res))

#9
# Python program to print even Numbers in a List

# list of numbers
list1 = [10, 21, 4, 45, 66, 93]

# using list comprehension
even_nos = [num for num in list1 if num % 2 == 0]

print("Even numbers in the list: ", even_nos)

#O/p-
#Even numbers in the list:  [10, 4, 66]

#10
#Python program to print
# duplicates from a list
# of integers
def Repeat(x):
    _size = len(x)
    repeated = []
    for i in range(_size):
        k = i + 1
        for j in range(k, _size):
            if x[i] == x[j] and x[i] not in repeated:
```

```
            repeated.append(x[i])
    return repeated

# Driver Code
list1 = [10, 20, 30, 20, 20, 30, 40,
        50, -20, 60, 60, -20, -20]
print (Repeat(list1))

#Output
#[20, 30, -20, 60]

#11
# program to print duplicate numbers in a given list
# provided input
list = [1, 2, 1, 2, 3, 4, 5, 1, 1, 2, 5, 6, 7, 8, 9, 9]

new = []  # defining output list

# condition for reviewing every
# element of given input list
for a in list:

    # checking the occurrence of elements
    n = list.count(a)

    # if the occurrence is more than
    # one we add it to the output list
    if n > 1:
```

```
    if new.count(a) == 0:  # condition to check

        new.append(a)

print(new)

#12
# Python code to count the number of occurrences
def countX(lst, x):
    return lst.count(x)

# Driver Code
lst = [8, 6, 8, 10, 8, 20, 10, 8, 8]
x = 8
print('{} has occurred {} times'.format(x, countX(lst, x)))

#13
# Python program to find the k most frequent words
# from data set
from collections import Counter

data_set = "Welcome to the world of Geeks " \
"This portal has been created to provide well written well" \
"thought and well explained solutions for selected questions " \
"If you like Geeks for Geeks and would like to contribute " \
"here is your chance You can write article and mail your article " \
" to contribute at geeksforgeeks org See your article appearing on " \
"the Geeks for Geeks main page and help thousands of other Geeks. " \
```

```
# split() returns list of all the words in the string
split_it = data_set.split()

# Pass the split_it list to instance of Counter class.
Counter = Counter(split_it)

# most_common() produces k frequently encountered
# input values and their respective counts.
most_occur = Counter.most_common(4)

print(most_occur)
#1
#Sort a list based on the reversed order
names = ['apple', 'google', 'yahoo', 'amazon', 'facebook', 'instagram', 'microsoft']
print(names[::-1])
#['microsoft', 'instagram', 'facebook', 'amazon', 'yahoo', 'google', 'apple']

#**********************************************************************************
**********
#2
#Sort a list based on their length
names = ['apple', 'google', 'yahoo', 'amazon', 'facebook', 'instagram', 'microsoft']
print(names.sort(key=len))

#['microsoft', 'instagram', 'facebook', 'amazon', 'yahoo', 'google', 'apple']

#**********************************************************************************
**********
# sort based on the length
```

```
words = ["yahoo", "instagram", "google", "flipkart", "walmart", "apple"]
print(sorted(words, key=len))

# sort based on last character of the elements in the list
def last_item(word):
    return word[-1]

last = lambda word: word[-1]
print(sorted(words, key=last))

# sorting a dictionary
d = {"4": "walmart", "1": "apple", "7": "instagram", "3": "yahoo"}

# sorting based on keys
print(sorted(d))
print(sorted(d.keys()))
print(sorted(d.items()))

# sorting based on values

print(sorted(d.values()))

def values_(item):
    return item[-1]

print(sorted(d.items(), key=values_))
print(sorted(d.items(), key=lambda item: item[-1]))

# sort the dictionary based on length of the key
```

```
d = {"ACME": 45.23, "AAPL": 612.78, "IBM": 205.55, "FB": 10.75, "HPQ": 37.20}
res = sorted(d.items(), key=lambda item: len(item[0]))
# print(dict(res))

# sort the dictionary based on last character of the key
d = {"ACME": 45.23, "AAPL": 612.78, "IBM": 205.55, "FB": 10.75, "HPQ": 37.20}
res = sorted(d.items(), key=lambda item: item[0][-1])
# print(dict(res))

# sort based on first character of value
d = {"Bangalore": "Traffic", "Mysore": "Palace", "Dharwad": "peda", "Bagalkot": "caves"}
res = sorted(d.items(), key=lambda item: item[1][0])
# print(dict(res))

# sort based on length of values
d = {"Bangalore": "Traffic", "Mysore": "Palace", "Dharwad": "peda", "Bagalkot": "caves"}
res = sorted(d.items(), key=lambda item: len(item[1]))
# print(dict(res))

#Anagram

def anagram(string1, string2):
    return sorted(string1) == sorted(string2)
# print(anagram("tea", "teaa"))



# grouping anagrams
words = ["tea", "eat", "silent", "hello", "listen", "ate"]
```

```
d = {}

for word in words:
    key = ''.join(sorted(word))
    if key not in d:
        d[key] = [word]
    else:
        d[key].append(word)
print(d)



#{'aet': ['eat', 'ate', 'tea'], 'eilnst': ['silent', 'listen'], 'ehllo': ['hello']}

# longest word
words = ["yahoo", "instagram", "google", "flipkart", "walmart", "apple"]
longest_word = ""

for word in words:
    if len(word) > len(longest_word):
        longest_word = word
print(longest_word)

# to print the longest non repeated word in the sentence

sentence = "python is a programming language and programming is fun"
words = sentence.split()
d = {}

d = {word: len(word) for word in words if words.count(word) == 1}
```

```
res = sorted(d.items(), key=lambda item: item[-1])
# print(res[-1])


# create a dictionary with word and length pair and get the longest and smallest words along with its length
sentence = "Today is Holi but still we are in class even afternoon we will be in class"
words = sentence.split()
d = {}

for word in words:
    if len(word) not in d:
        d[len(word)] = [word]
    else:
        d[len(word)] += [word]      # d[len(word)].append(word)
# print(d)

res = sorted(d.items())
smallest, *rest, longest = res
# print(smallest)
# print(longest)

# to print the most common word in the list

words = ["apple", "google", "gmail", "google", "apple", "google", "flipkart"]
d = {}

for word in words:
    if word not in d:
        d[word] = 1
```

```
    else:
        d[word] += 1
print(d)

res = sorted(d.items(), key=lambda item: item[1])
least_common, *rest, most_common = res
print(most_common)

# using Counter
from collections import Counter
words = ["apple", "apple", "google", "gmail", "google", "apple", "google", "flipkart"]
c = Counter(words)
print(c)
print(c.most_common())

                                        #Assignments
#Using Comprehension
#1
#prime numbers
#A number that is divisible only by itself and 1 (e.g. 2, 3, 5, 7, 11).
#Normal program

#n = 50
#primes = []
'''
for i in range(2, n):
        for j in range(2, int(i ** 0.5) + 1):
                if i%j == 0:
                        break
```

```
            else:
                    primes.append(i)

print(primes)

'''

#Using list Comprehension
print([i for i in range(2, 50) if 0 not in [i%n for n in range(2, i)]])

#[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47]



#WAP to check wheather given no is Prime or not.
#Python program to check whether a number is  prime or not
'''
n = 5
if n > 1:
   for i in range(2, n):
      if n % i == 0:
         print("not prime")
   print("prime")



#Using function
def prime(n):
   for i in range(2, n):
      if n % i == 0:
         return f'{n} is not prime'
```

```
        return f'{n} is prime '
print(prime(8))
'''

#*****************************************************************************************
*****************************

#2. Reverse the item of a list if the item is of odd length string
'''
names = ['apple', 'google', 'yahoo', 'facebook', 'yelp', 'flipkart', 'gmail', 'instagram', 'microsoft']

list = [name[::-1] for name in names if len(name)%2 != 0]
print(list)
'''
#['elppa', 'oohay', 'liamg', 'margatsni', 'tfosorcim']



#2.1. Reverse the item of a list if the item is of odd length string otherwise keep the item as is!

#NP
'''
names = ['apple', 'google', 'yahoo', 'facebook', 'yelp', 'flipkart', 'gmail', 'instagram', 'microsoft']

l1 = []
for word in names:
    if len(word) % 2 == 0:
        l1.append(word)
    else:
```

```
        l1.append(word[::-1])
print(l1)

#['elppa', 'google', 'oohay', 'facebook', 'yelp', 'flipkart', 'liamg', 'margatsni', 'tfosorcim']



#Using list Comprehension
l2 = [word if len(word) % 2 == 0 else word[::-1] for word in names]
print(l2)
#['elppa', 'google', 'oohay', 'facebook', 'yelp', 'flipkart', 'liamg', 'margatsni', 'tfosorcim']
'''

#*********************************************************************************
*****************************
```