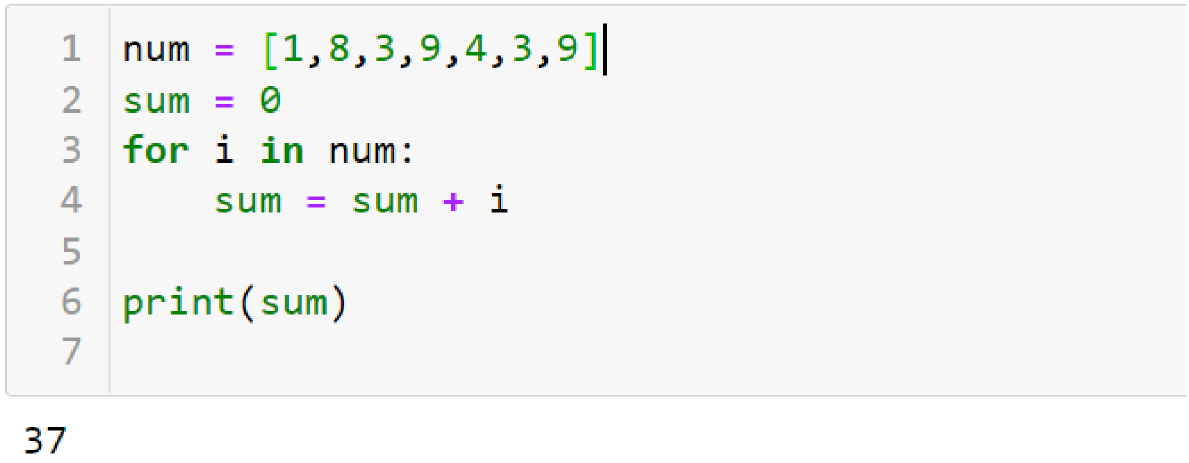
LIST Questions

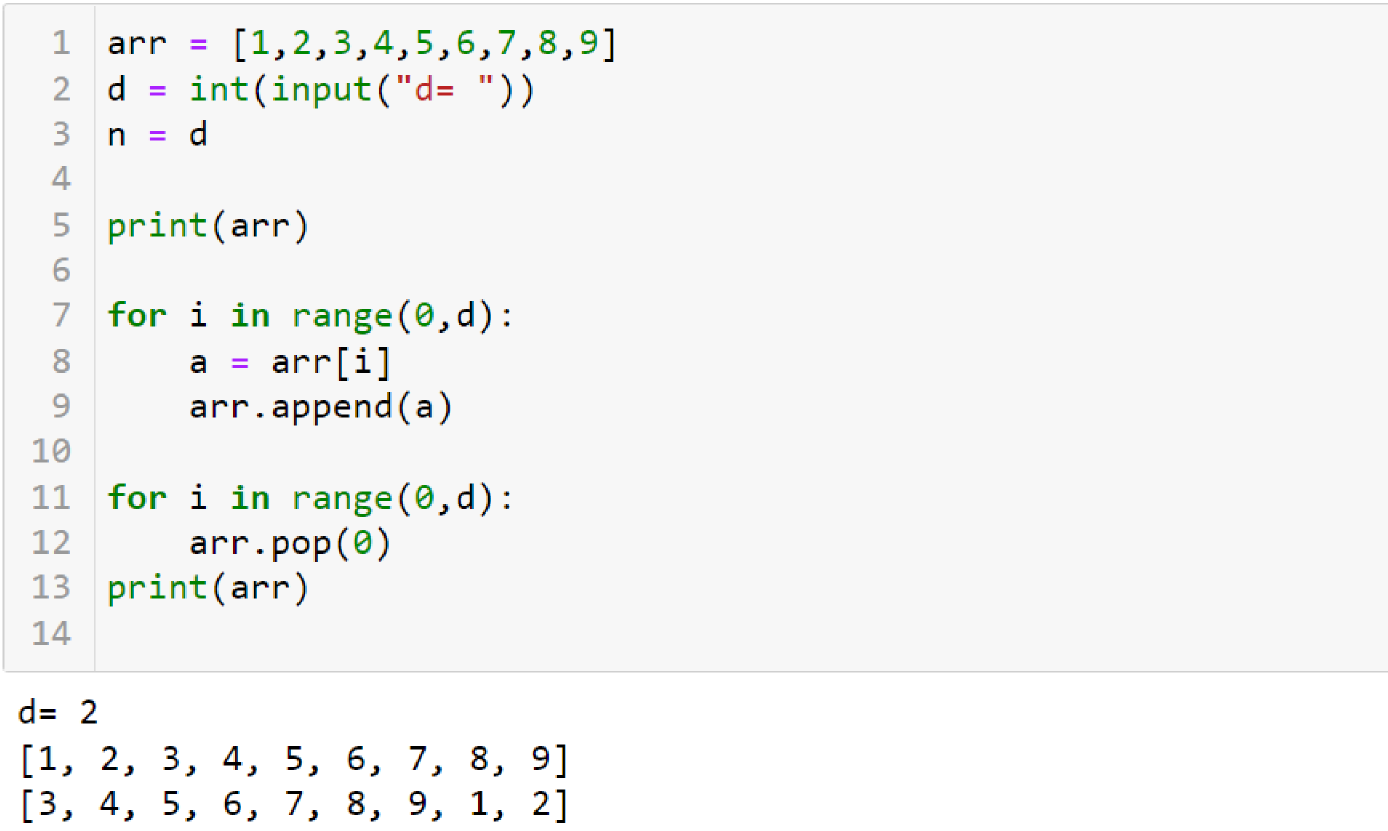
1. Given a list of numbers, write a Python program to find the sum of all the elements in the list.?



```
num = [1,8,3,9,4,3,9]
sum = 0
for i in num:
    sum = sum + i

print(sum)

```

```
37
```

2. Given an array of integers arr[ ] of size N and an integer, the task is to rotate the array elements to the left by d positions.?



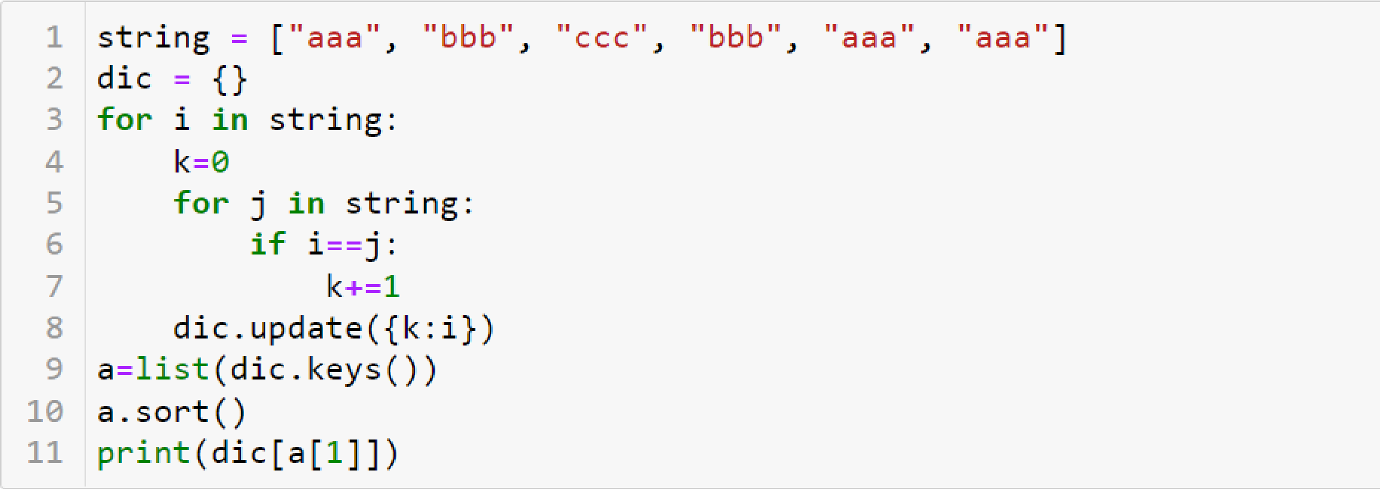
```
arr = [1,2,3,4,5,6,7,8,9]
d = int(input("d= "))
n = d

print(arr)

for i in range(0,d):
    a = arr[i]
    arr.append(a)

for i in range(0,d):
    arr.pop(0)
print(arr)

```

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

3. Second most repeated word in a sequence in Python?

Given a sequence of strings, the task is to find out the second mostrepeated (or frequent) string in the given sequence.

Input : ["aaa", "bbb", "ccc", "bbb", "aaa", "aaa"]
Output : bbb

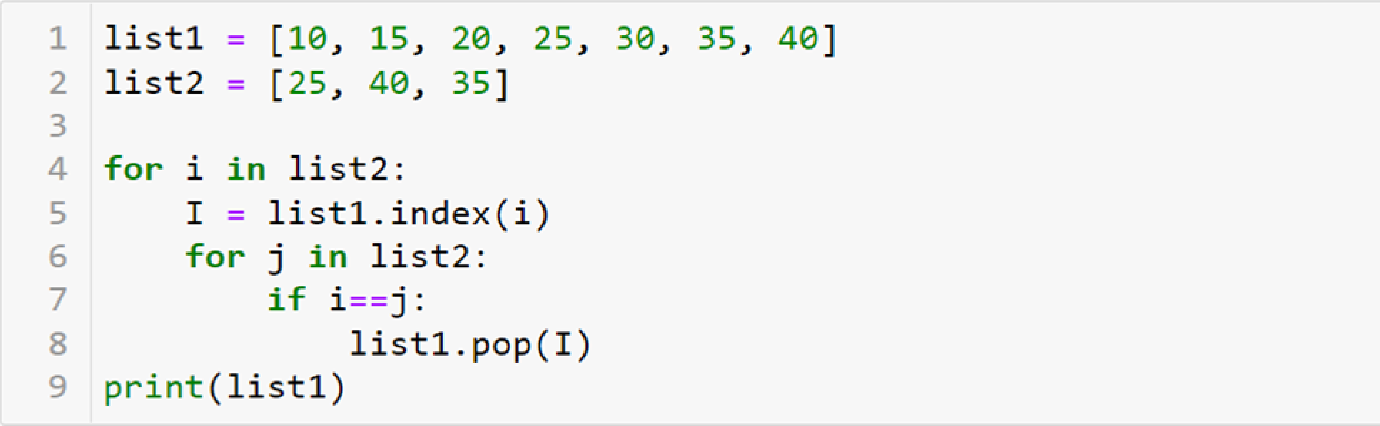


```python
string = ["aaa", "bbb", "ccc", "bbb", "aaa", "aaa"]
dic = {}
for i in string:
    k=0
    for j in string:
        if i==j:
            k+=1
    dic.update({k:i})
a=list(dic.keys())
a.sort()
print(dic[a[1]])
```

```
bbb
```

4. Difference between two lists ?

Input:
list1 = [10, 15, 20, 25, 30, 35, 40]
list2 = [25, 40, 35]
Output:
[10, 20, 30, 15]

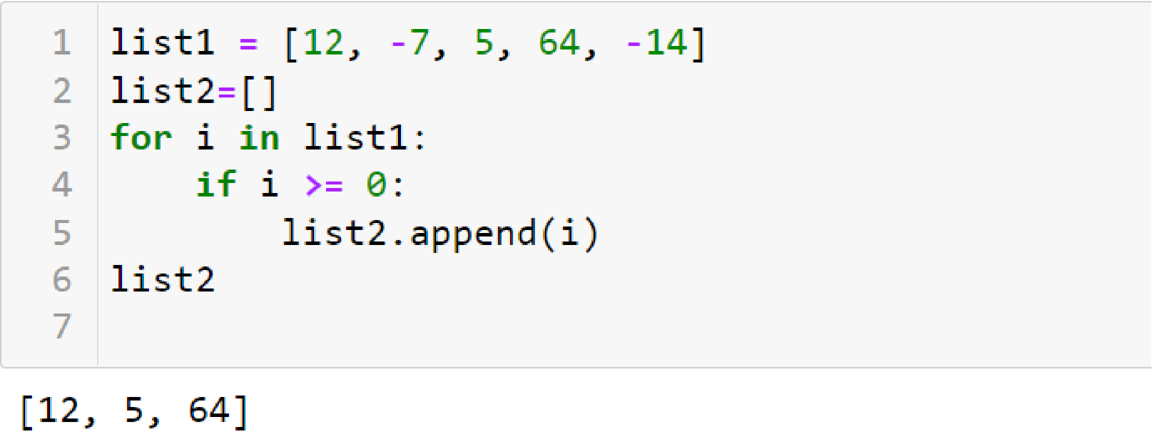


```python
list1 = [10, 15, 20, 25, 30, 35, 40]
list2 = [25, 40, 35]

for i in list2:
    I = list1.index(i)
    for j in list2:
        if i==j:
            list1.pop(I)
print(list1)
```

```
[10, 15, 20, 30]
```

5. Print all positive numbers from given list using for loop Iterate each element in the list using for loop and check if number is greater than or equal to 0. If the condition satisfies, then only print the number.?

# Python program to print positive Numbers in a List
Input: list1 = [12, -7, 5, 64, -14]
Output: 12, 5, 64
Input: list2 = [12, 14, -95, 3]
Output: [12, 14, 3]

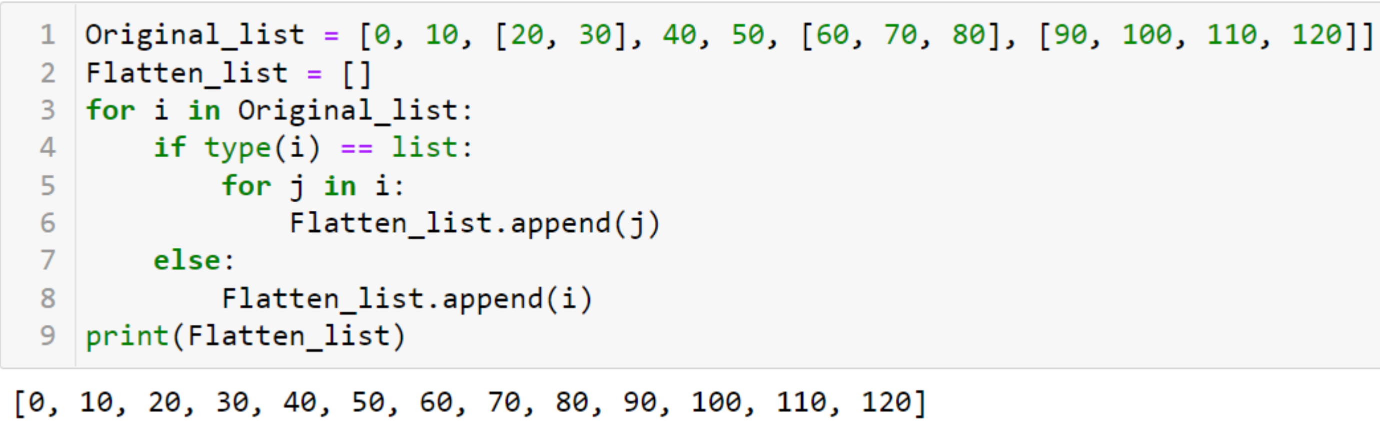


```
list1 = [12, -7, 5, 64, -14]
list2=[]
for i in list1:
    if i >= 0:
        list2.append(i)
list2

```

```
[12, 5, 64]
```

6. Write a Python program to flatten a given nested list structure.?

Original list: [0, 10, [20, 30], 40, 50, [60, 70, 80], [90, 100, 110, 120]]
Flatten list:
[0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120]

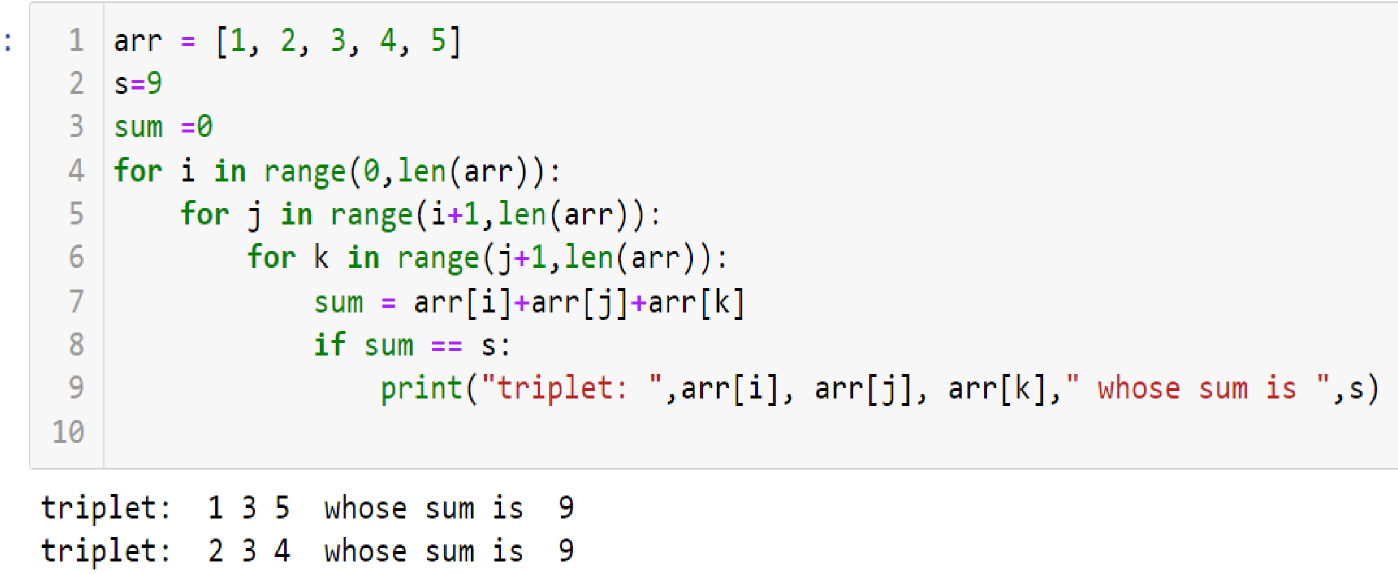


```
Original_list = [0, 10, [20, 30], 40, 50, [60, 70, 80], [90, 100, 110, 120]]
Flatten_list = []
for i in Original_list:
    if type(i) == list:
        for j in i:
            Flatten_list.append(j)
    else:
        Flatten_list.append(i)
print(Flatten_list)
```

```
[0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120]
```

7. Given an array and a value, find if there is a triplet in array whose sum is equal to the given value. If there is such a triplet present in array, then print the triplet and return true. Else return false.?

Input: array = [12, 3, 4, 1, 6, 9], sum = 24;
Output: 12, 3, 9
Explanation: There is a triplet (12, 3 and 9) present
in the array whose sum is 24.
Input: array = [1, 2, 3, 4, 5], sum = 9
Output: 5, 3, 1
Explanation: There is a triplet (5, 3 and 1) present
in the array whose sum is 9.



:

```
arr = [1, 2, 3, 4, 5]
s=9
sum =0
for i in range(0,len(arr)):
    for j in range(i+1,len(arr)):
        for k in range(j+1,len(arr)):
            sum = arr[i]+arr[j]+arr[k]
            if sum == s:
                print("triplet: ",arr[i], arr[j], arr[k]," whose sum is ",s)

```

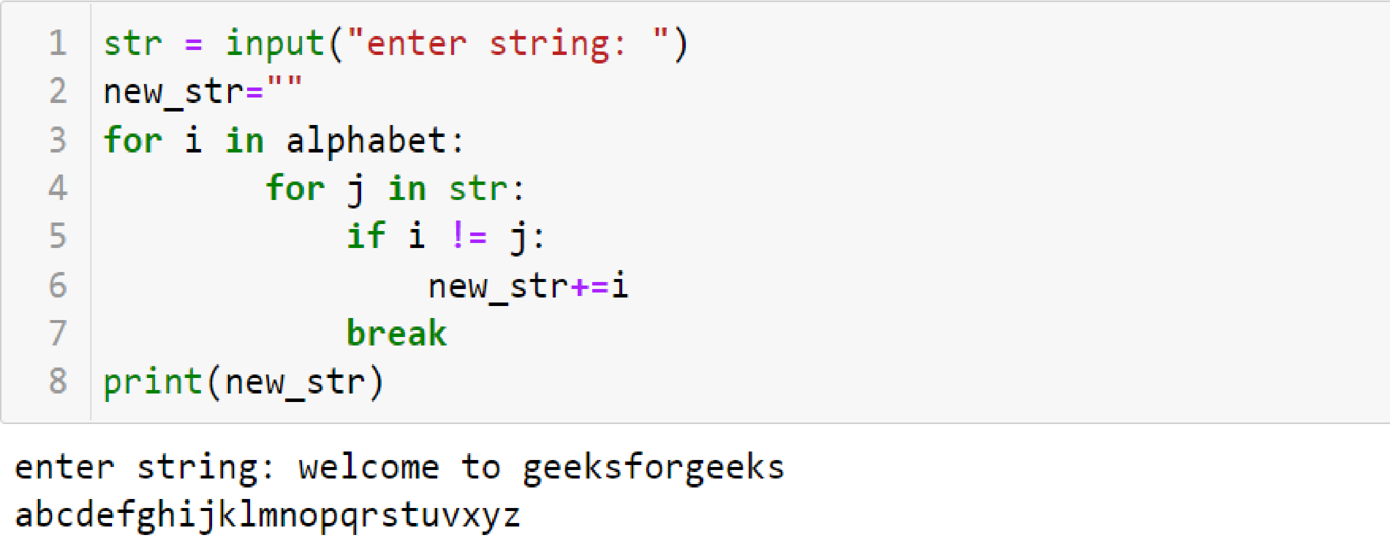
```
triplet:  1 3 5  whose sum is  9
triplet:  2 3 4  whose sum is  9
```

Strings

1. Missing characters to make a string Pangram.?

Pangram is a sentence containing every letter in the English alphabet.
Given a string, find all characters that are missing from the string, i.e.,
the characters that can make the string a Pangram.
We need to print output in alphabetic order.

Input : welcome to geeksforgeeks
Output : abdhijnpquvxyz
Input : The quick brown fox jumps
Output : adglvyz

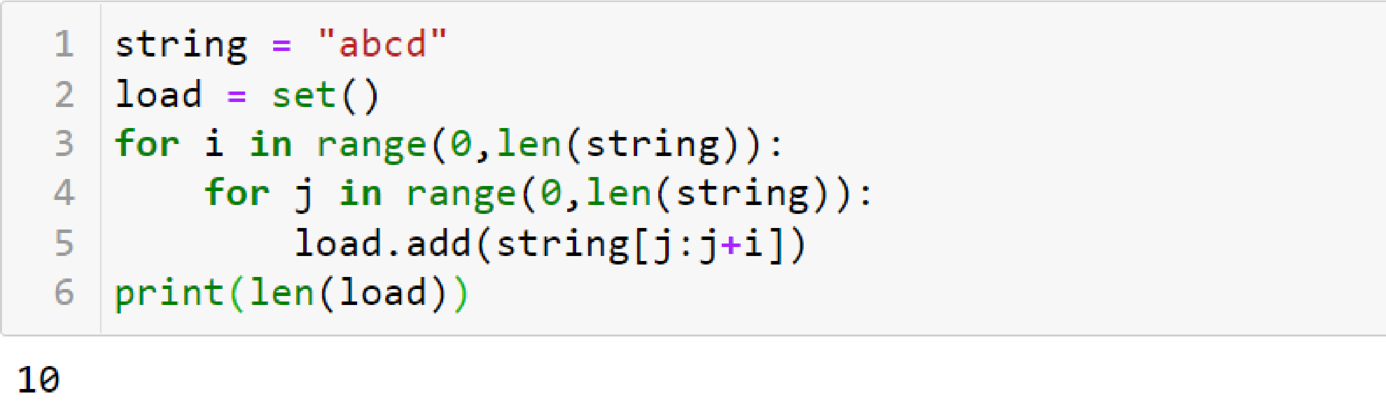


```
str = input("enter string: ")
new_str=""
for i in alphabet:
    for j in str:
      if i != j:
        new_str+=i
      break
print(new_str)
```

```
enter string: welcome to geeksforgeeks
abcdefghijklmnopqrstuvxyz
```

2. Find total number of non-empty substrings of a string with N characters.?

Input : str = “abc”
Output : 6
Every substring of the given string : “a”, “b”, “c”, “ab”, “bc”, “abc”
Input : str = “abcd”
Output : 10
Every substring of the given string : “a”, “b”, “c”, “d”, “ab”,
“bc”, “cd”, “abc”, “bcd” and “abcd”

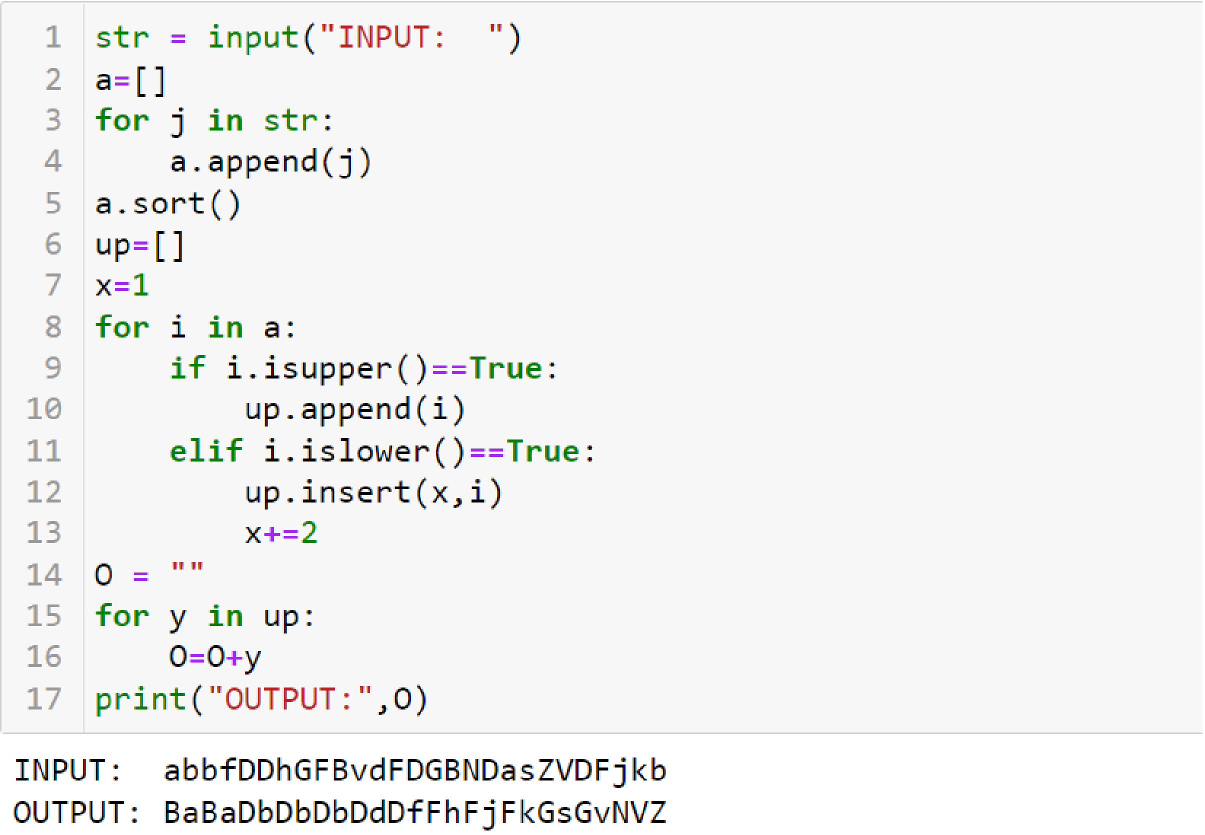


```python
string = "abcd"
load = set()
for i in range(0,len(string)):
    for j in range(0,len(string)):
        load.add(string[j:j+i])
print(len(load))
```

```
10
```

3. Given a string containing lowercase and uppercase letters. Sort it in such a manner that the uppercase and lowercase letters come in an alternate manner but in a sorted way.?

Input : bAwutndekWEdkd
Output :AbEdWddekkntuw
Explanation:
Here we can see that letter 'A', 'E', 'W' are sorted
as well as letters "b, d, d, d, e, k, k, n, t, u, w" are sorted
but both appears alternately in the string as far as possible.
Input :abbfDDhGFBvdFDGBNDasZVDFjkb
Output :BaBaDbDbDbDdDfFhFjFkGsGvNVZ

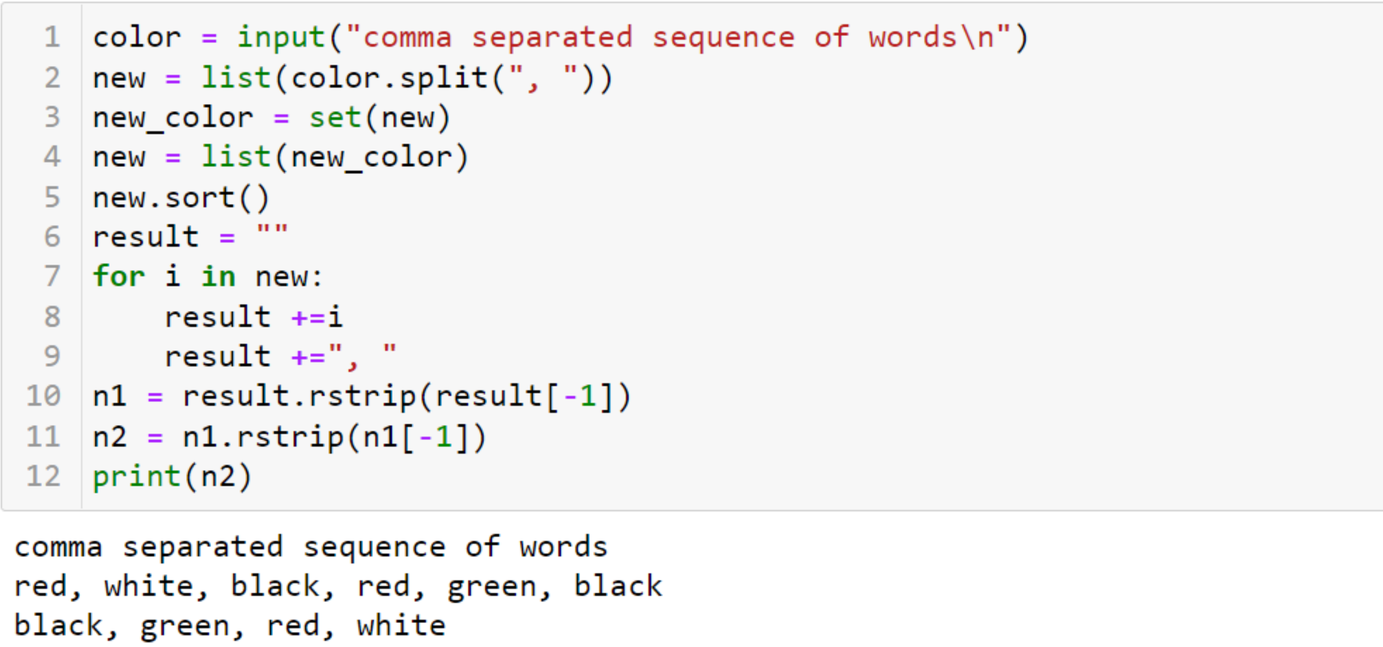


```
str = input("INPUT:  ")
a=[]
for j in str:
    a.append(j)
a.sort()
up=[]
x=1
for i in a:
    if i.isupper()==True:
        up.append(i)
    elif i.islower()==True:
        up.insert(x,i)
        x+=2
O = ""
for y in up:
    O=O+y
print("OUTPUT:",O)
```

```
INPUT:   abbfDDhGFBvdFDGBNDasZVDFjkb
OUTPUT:  BaBaDbDbDbDdDfFhFjFkGsGvNVZ
```

4. Write a Python program that accepts a comma separated sequence of words as input and prints the unique words in sorted form (alphanumerically). ?

Sample Words : red, white, black, red, green, black
Expected Result : black, green, red, white,red

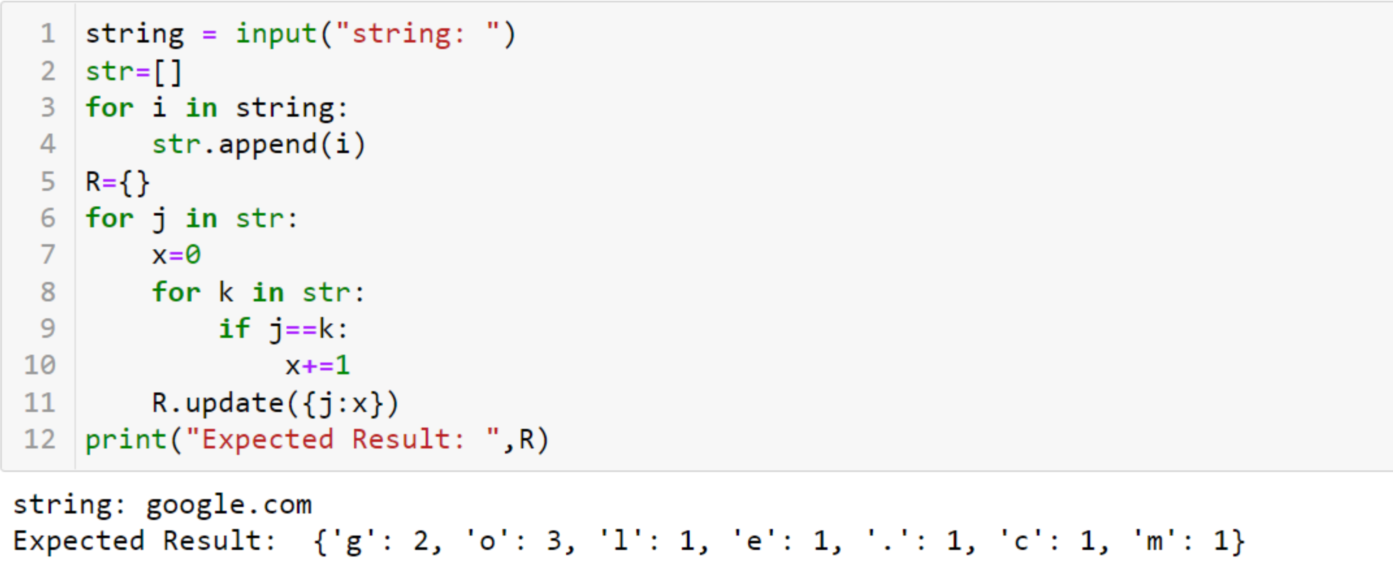


```
color = input("comma separated sequence of words\n")
new = list(color.split(", "))
new_color = set(new)
new = list(new_color)
new.sort()
result = ""
for i in new:
    result +=i
    result +=", "
n1 = result.rstrip(result[-1])
n2 = n1.rstrip(n1[-1])
print(n2)
```

```
comma separated sequence of words
red, white, black, red, green, black
black, green, red, white
```

5. Write a Python program to count the number of characters (character frequency) in a string. ?

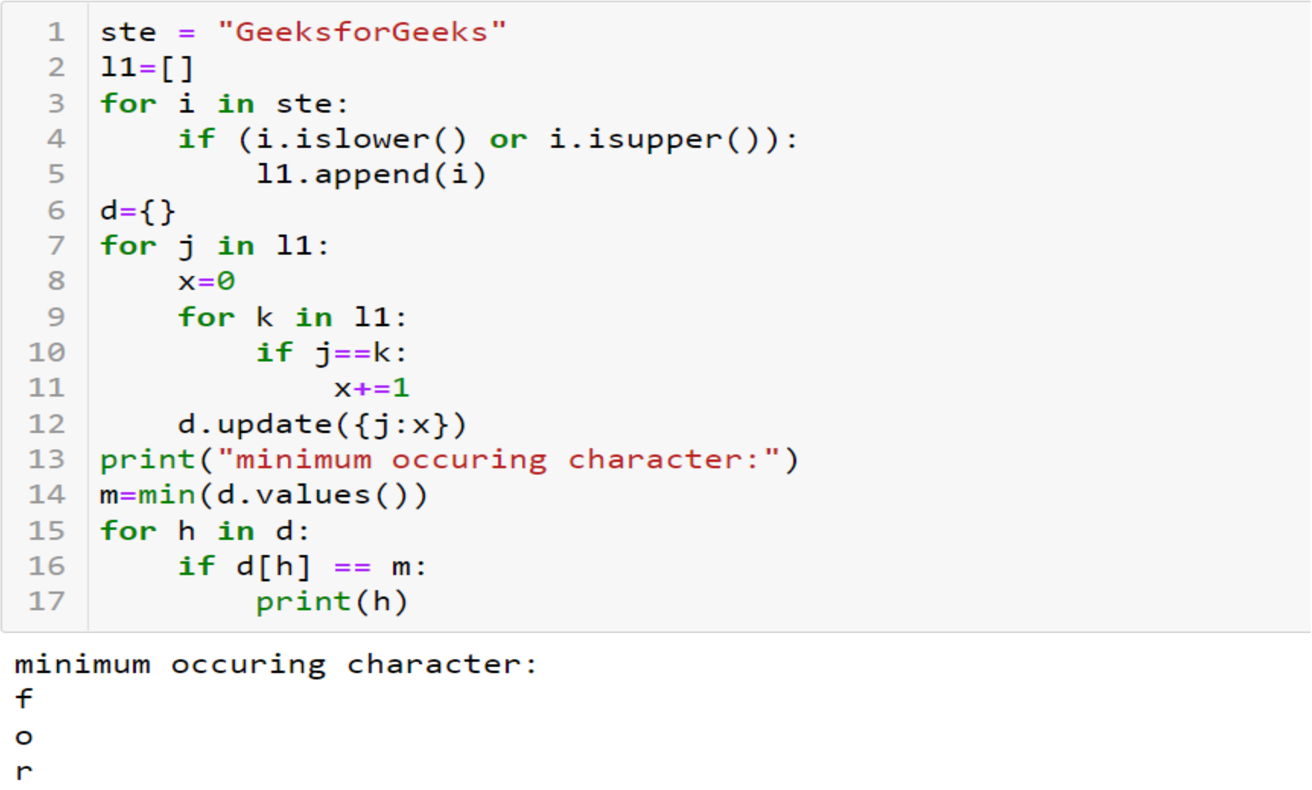
Sample String : google.com'
Expected Result : {'g': 2, 'o': 3, 'l': 1, 'e': 1, '.': 1, 'c': 1, 'm': 1}



```
string = input("string: ")
str=[]
for i in string:
    str.append(i)
R={}
for j in str:
    x=0
    for k in str:
        if j==k:
            x+=1
    R.update({j:x})
print("Expected Result: ",R)
```

```
string: google.com
Expected Result:  {'g': 2, 'o': 3, 'l': 1, 'e': 1, '.': 1, 'c': 1, 'm': 1}
```

6. find the frequency of minimum occurring character in a python string ?

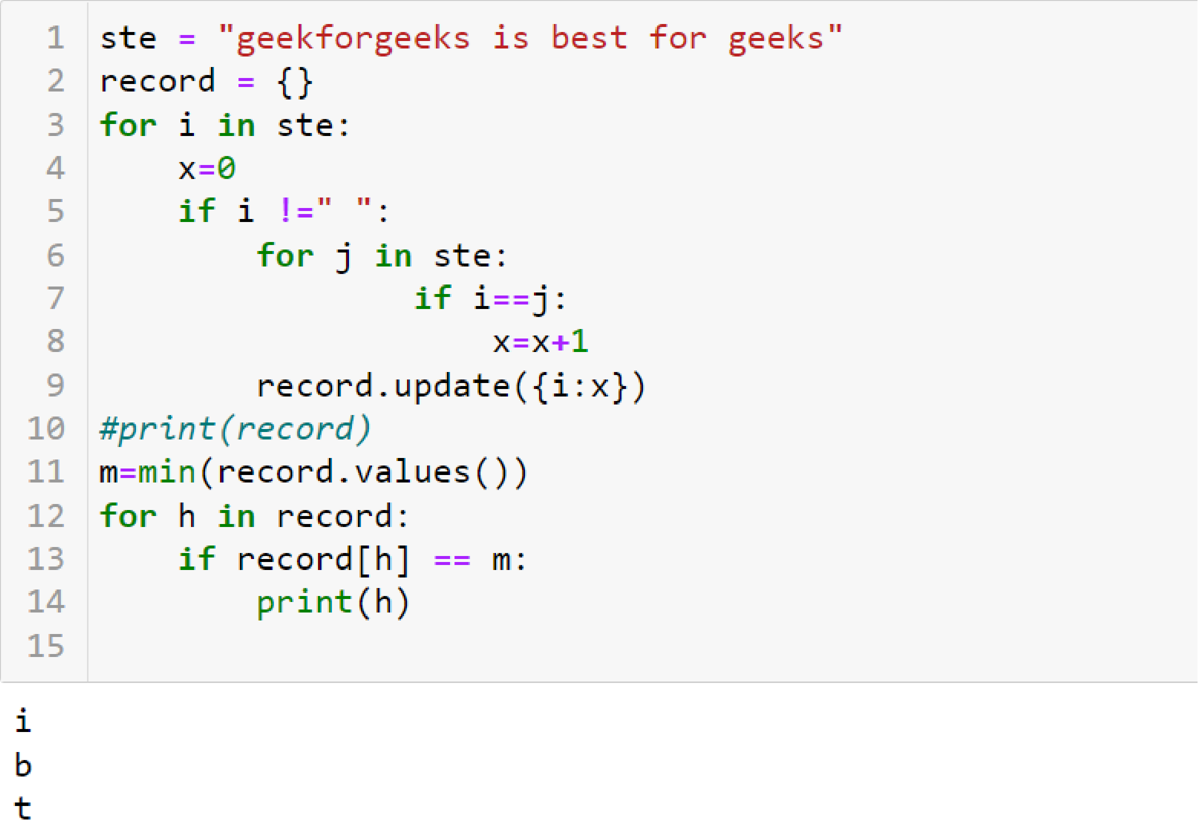


```
ste = "GeeksforGeeks"
l1=[]
for i in ste:
    if (i.islower() or i.isupper()):
        l1.append(i)
d={}
for j in l1:
    x=0
    for k in l1:
        if j==k:
            x+=1
    d.update({j:x})
print("minimum occuring character:")
m=min(d.values())
for h in d:
    if d[h] == m:
        print(h)
```

```
minimum occuring character:
f
o
r
```

7. Write a program extract all the string characters which have odd number of occurrences.

The original string is : geekforgeeks is best for geeks
The Odd Frequency Characters are : ['k', 'i', 't', 'g', 'e', 'b']

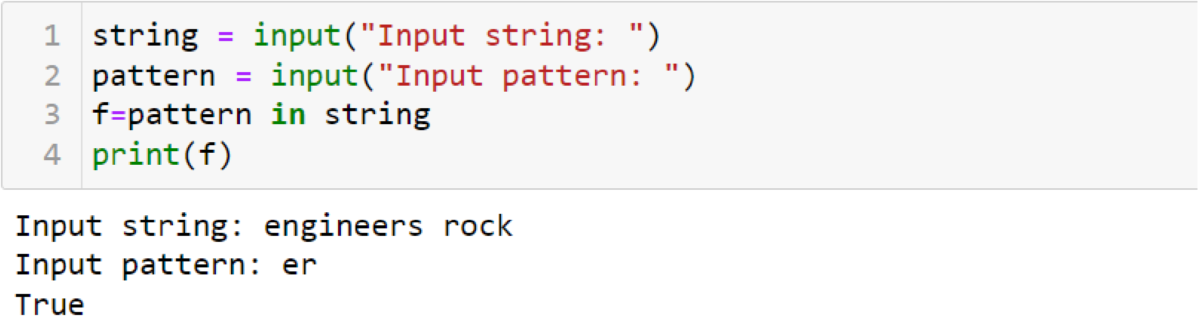


```
ste = "geekforgeeks is best for geeks"
record = {}
for i in ste:
    x=0
    if i !=" ":
        for j in ste:
                if i==j:
                    x=x+1
        record.update({i:x})
#print(record)
m=min(record.values())
for h in record:
    if record[h] == m:
        print(h)

```

```
i
b
t
```

Dictionary

1. Given an input string and a pattern, check if characters in the input string follows the same order as determined by characters present in the pattern. Assume there won't be any duplicate characters in the pattern.?

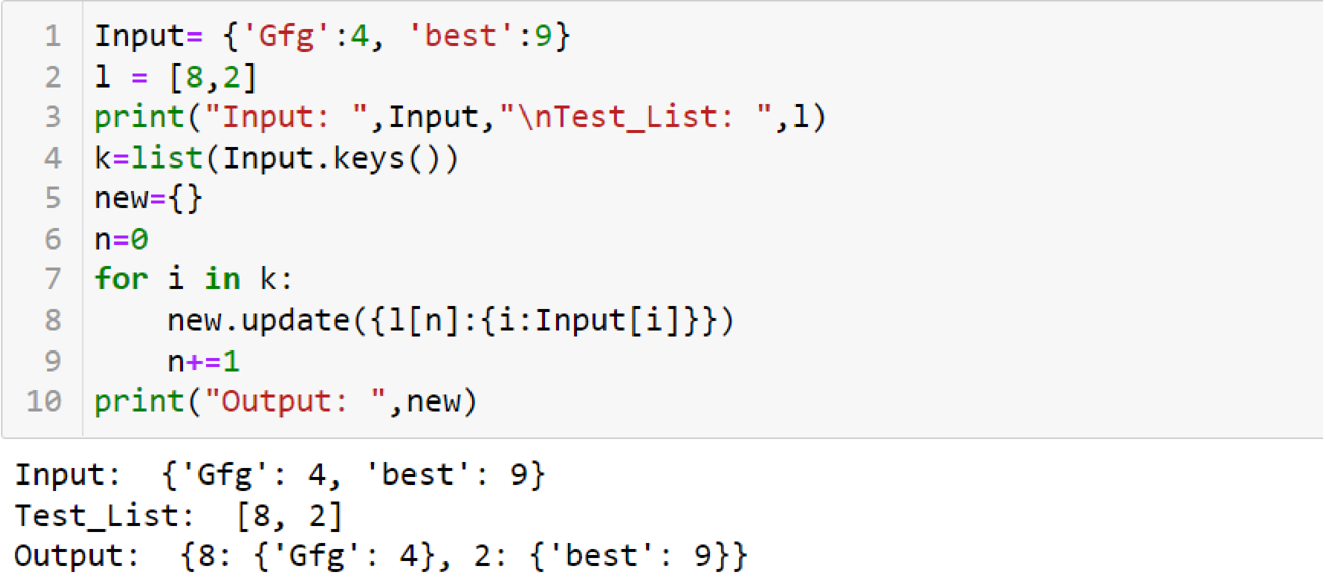


```
string = input("Input string: ")
pattern = input("Input pattern: ")
f=pattern in string
print(f)
```

```
Input string: engineers rock
Input pattern: er
True
```

2. Given a list and dictionary, map each element of list with each item of dictionary, forming nested dictionary as value.?

Input : test_dict = {'Gfg' : 4, 'best' : 9}, test_list = [8, 2]
Output : {8: {'Gfg': 4}, 2: {'best': 9}}

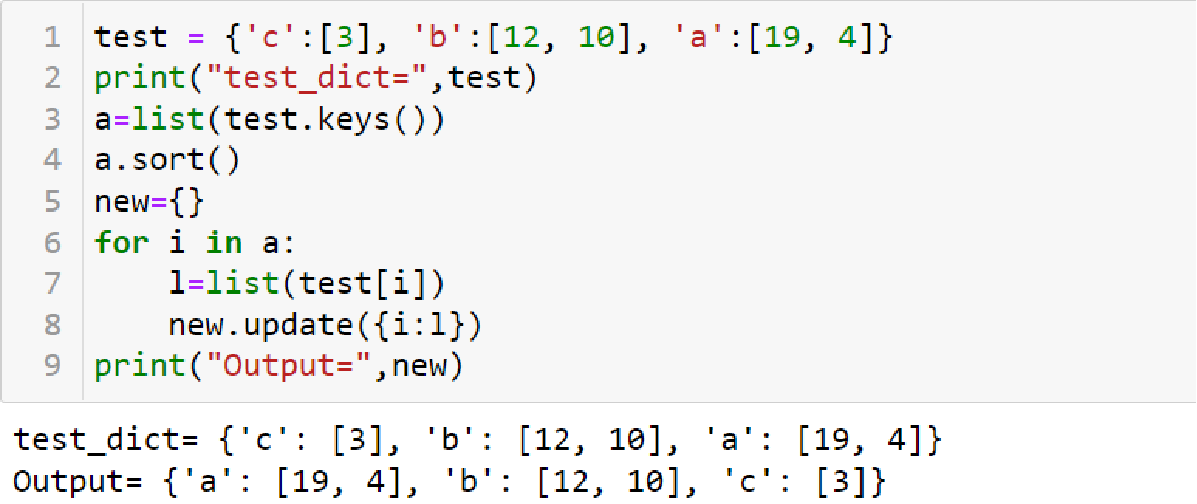


```
Input= {'Gfg':4, 'best':9}
l = [8,2]
print("Input: ",Input,"\nTest_List: ",l)
k=list(Input.keys())
new={}
n=0
for i in k:
    new.update({l[n]:{i:Input[i]}})
    n+=1
print("Output: ",new)
```

```
Input:  {'Gfg': 4, 'best': 9}
Test_List:  [8, 2]
Output:  {8: {'Gfg': 4}, 2: {'best': 9}}
```

3. Sort Dictionary key and values List ?

Input : test_dict = {‘c’: [3], ‘b’: [12, 10], ‘a’: [19, 4]}
Output : {‘a’: [4, 19], ‘b’: [10, 12], ‘c’: [3]}

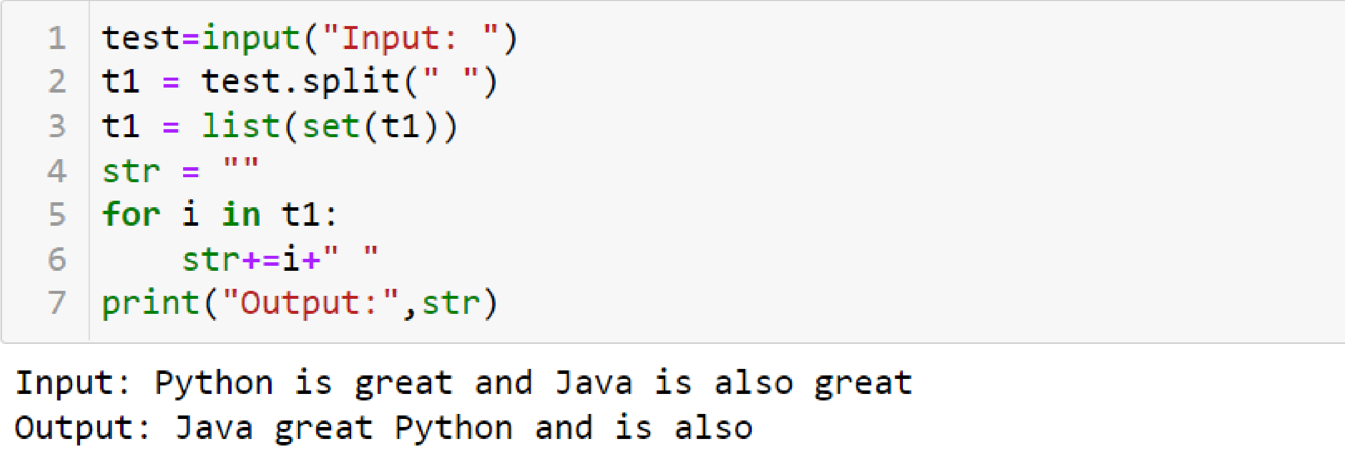


```python
test = {'c':[3], 'b':[12, 10], 'a':[19, 4]}
print("test_dict=",test)
a=list(test.keys())
a.sort()
new={}
for i in a:
    l=list(test[i])
    new.update({i:l})
print("Output=",new)
```

```
test_dict= {'c': [3], 'b': [12, 10], 'a': [19, 4]}
Output= {'a': [19, 4], 'b': [12, 10], 'c': [3]}
```

4. Remove all duplicates words from a given sentence ?

Input : Python is great and Java is also great
Output : is also Java Python and great

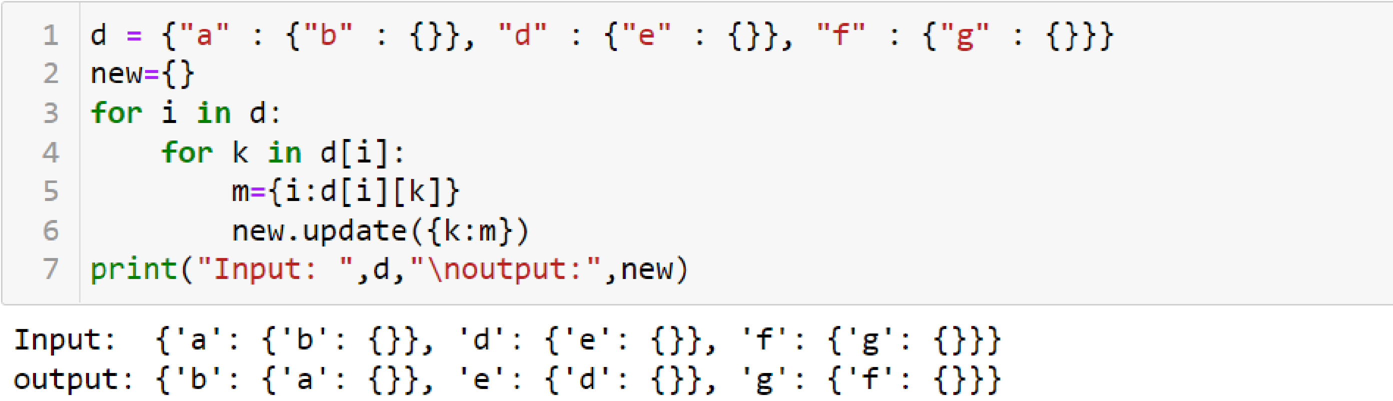


```python
test=input("Input: ")
t1 = test.split(" ")
t1 = list(set(t1))
str = ""
for i in t1:
    str+=i+" "
print("Output:",str)
```

```
Input: Python is great and Java is also great
Output: Java great Python and is also
```

5. Inversion in nested dictionary?

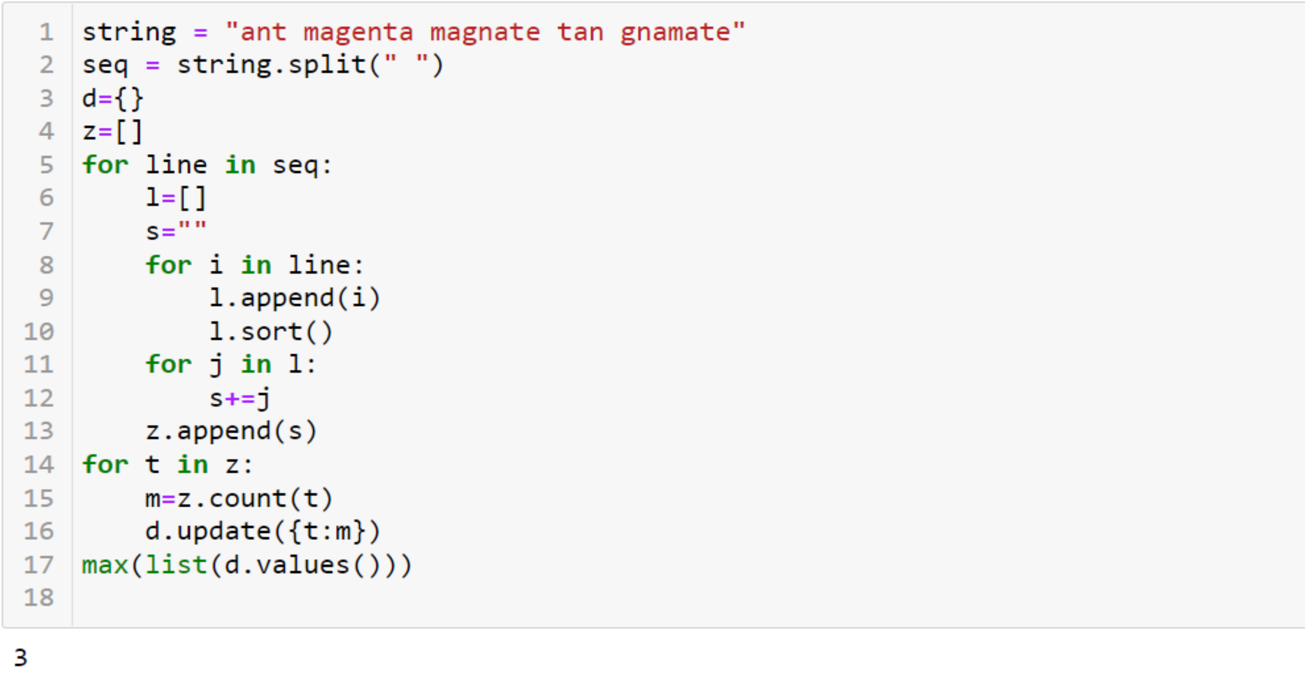
Input : test_dict = {“a” : {“b” : {}},
“d” : {“e” : {}},
“f” : {“g” : {}}
Output : {‘b’: {‘a’: {}}, ‘e’: {‘d’: {}}, ‘g’: {‘f’: {}}
Explanation : Nested dictionaries inverted as outer dictionary keys and viz-a-vis.



```
d = {"a" : {"b" : {}}, "d" : {"e" : {}}, "f" : {"g" : {}}}
new={}
for i in d:
    for k in d[i]:
        m={i:d[i][k]}
        new.update({k:m})
print("Input: ",d,"\noutput:",new)
```

```
Input:  {'a': {'b': {}}, 'd': {'e': {}}, 'f': {'g': {}}}
output: {'b': {'a': {}}, 'e': {'d': {}}, 'g': {'f': {}}}
```

6. Given an array of n string containing lowercase letters. Find the size of largest subset of string which are anagram of each others. An anagram of a string is another string that contains same characters, only the order of characters can be different. For example, “abcd” and “dabc” are anagram of each other.?



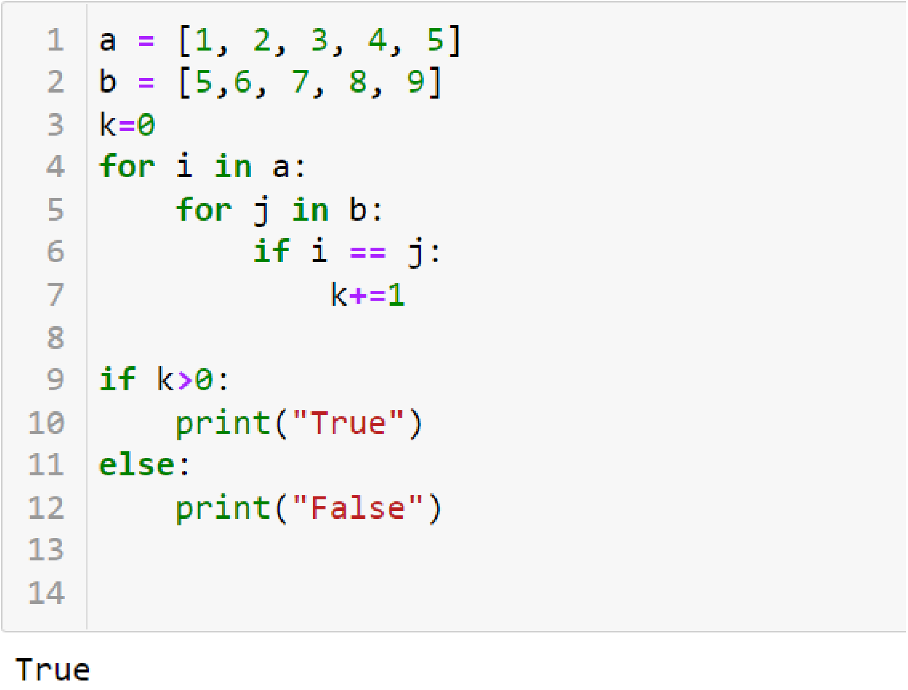
```
string = "ant magenta magnate tan gnamate"
seq = string.split(" ")
d={}
z=[]
for line in seq:
    l=[]
    s=""
    for i in line:
        l.append(i)
        l.sort()
    for j in l:
        s+=j
    z.append(s)
for t in z:
    m=z.count(t)
    d.update({t:m})
max(list(d.values()))
```

```
3
```

Sets

1. Given two lists a, b. Check if two lists have at least one element common in them.

Input : a = [1, 2, 3, 4, 5]
b = [5, 6, 7, 8, 9]
Output : True

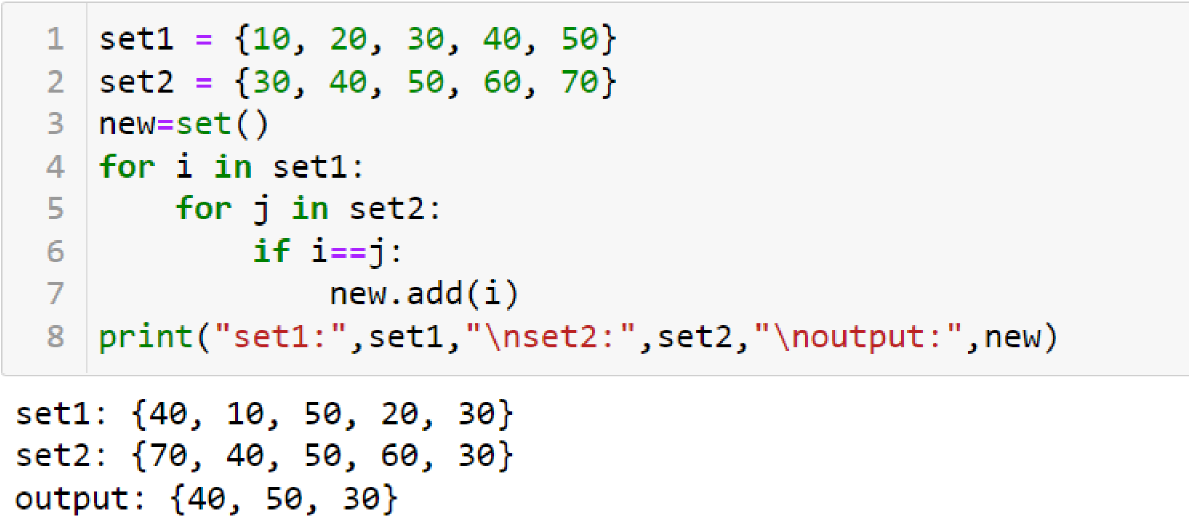


```
a = [1, 2, 3, 4, 5]
b = [5,6, 7, 8, 9]
k=0
for i in a:
    for j in b:
        if i == j:
            k+=1

if k>0:
    print("True")
else:
    print("False")
```

```
True
```

2. Return a new set of identical items from two sets

set1 = {10, 20, 30, 40, 50}
set2 = {30, 40, 50, 60, 70}
Expected output:
{40, 50, 30}

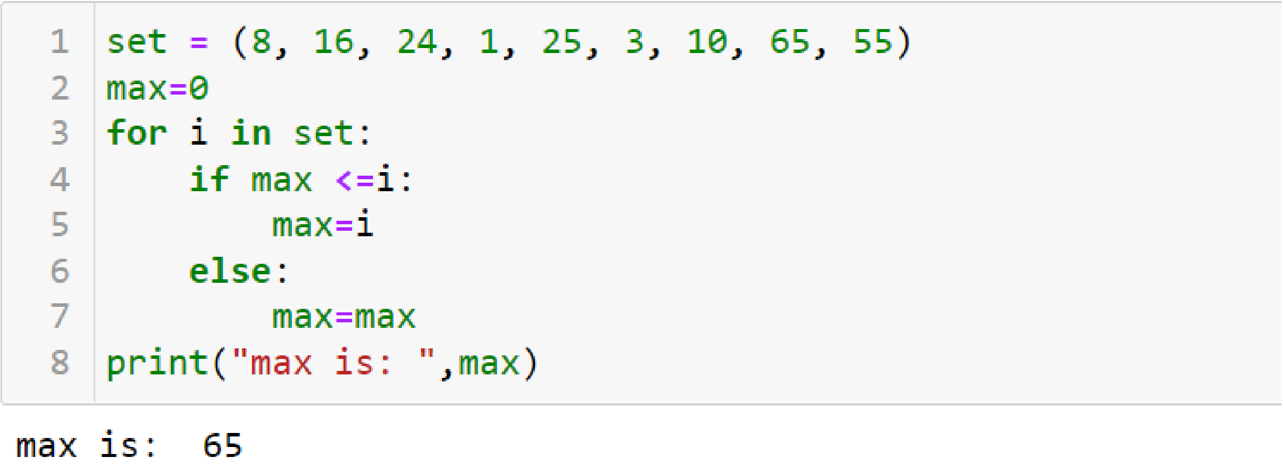


```python
set1 = {10, 20, 30, 40, 50}
set2 = {30, 40, 50, 60, 70}
new=set()
for i in set1:
    for j in set2:
        if i==j:
            new.add(i)
print("set1:",set1,"\nset2:",set2,"\noutput:",new)
```

```
set1: {40, 10, 50, 20, 30}
set2: {70, 40, 50, 60, 30}
output: {40, 50, 30}
```

3. Maximum and Minimum in a Set without use of inbuild max/min functions?

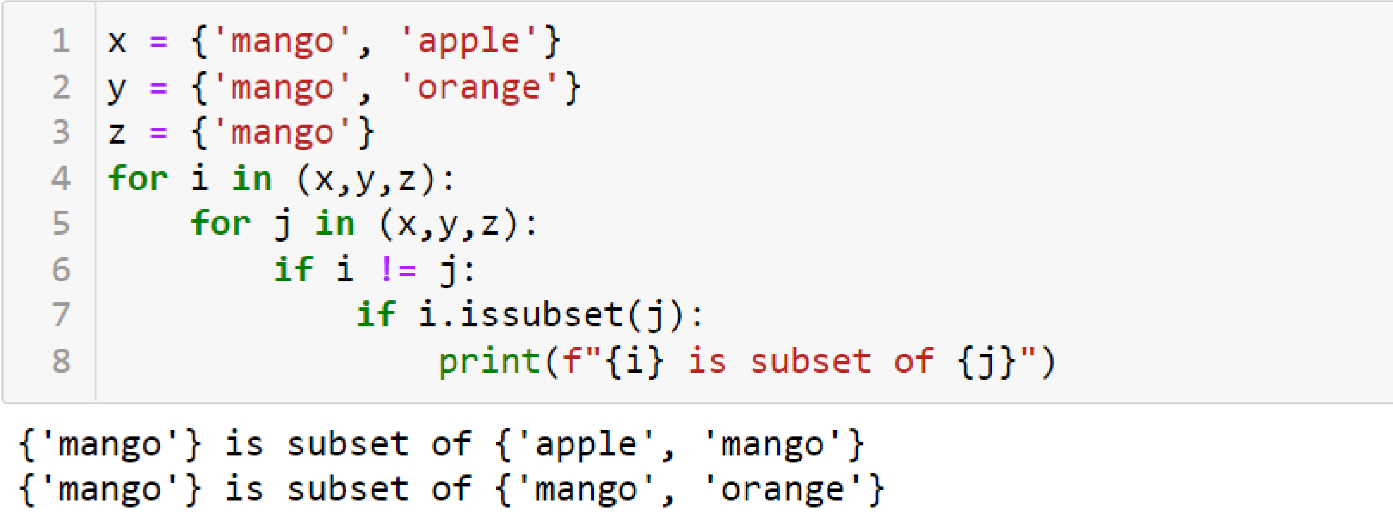
Input : set = ([8, 16, 24, 1, 25, 3, 10, 65, 55])
Output : max is 65
Input : set = ([4, 12, 10, 9, 4, 13])
Output : min is 4



```python
set = (8, 16, 24, 1, 25, 3, 10, 65, 55)
max=0
for i in set:
    if max <=i:
        max=i
    else:
        max=max
print("max is: ",max)
```

```
max is:  65
```

4. Write a Python program to check if a set is a subset of another set

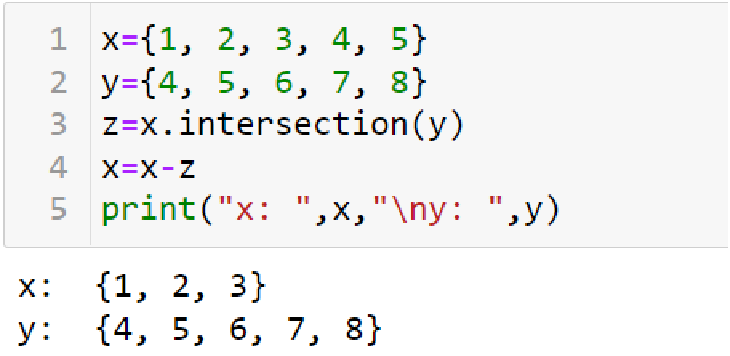


```python
x = {'mango', 'apple'}
y = {'mango', 'orange'}
z = {'mango'}
for i in (x,y,z):
    for j in (x,y,z):
        if i != j:
            if i.issubset(j):
                print(f"{i} is subset of {j}")
```

```
{'mango'} is subset of {'apple', 'mango'}
{'mango'} is subset of {'mango', 'orange'}
```

5. Write a Python program to remove the intersection of a 2nd set from the 1st set

input:-
Original sets:
{1, 2, 3, 4, 5}
{4, 5, 6, 7, 8}
Output:-
sn1: {1, 2, 3}
sn2: {4, 5, 6, 7, 8}



```python
x={1, 2, 3, 4, 5}
y={4, 5, 6, 7, 8}
z=x.intersection(y)
x=x-z
print("x: ",x,"\ny: ",y)
```

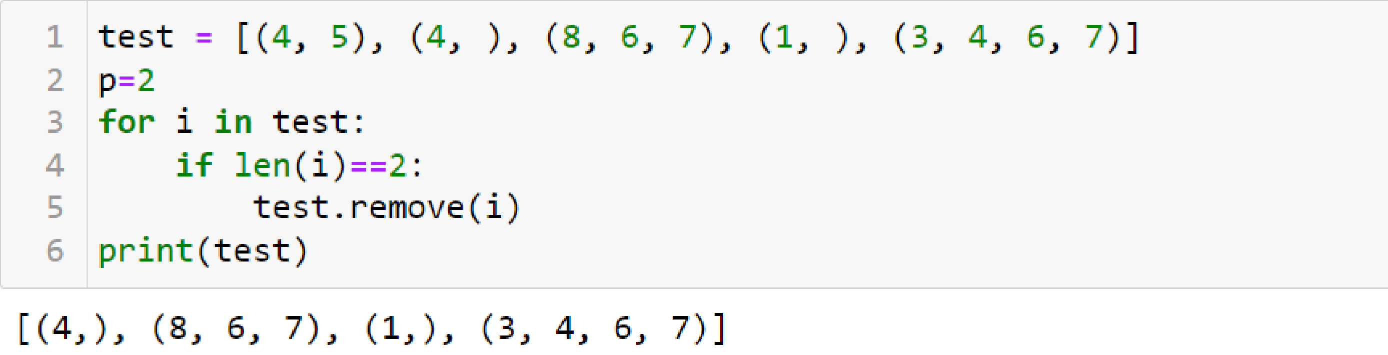
```
x:  {1, 2, 3}
y:  {4, 5, 6, 7, 8}
```

6. What is the result of passing a dictionary to a set constructor?

Tuple

1. Remove Tuples of Length K ?

Input : test_list = [(4, 5), (4,), (8, 6, 7), (1,), (3, 4, 6, 7)], K = 2
Output : [(4,), (8, 6, 7), (1,), (3, 4, 6, 7)]

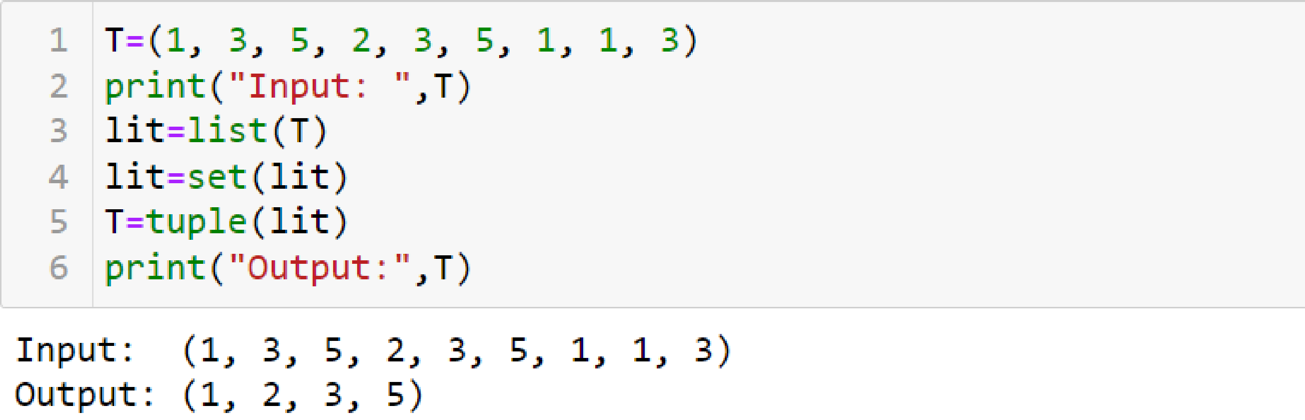


```
test = [(4, 5), (4, ), (8, 6, 7), (1, ), (3, 4, 6, 7)]
p=2
for i in test:
    if len(i)==2:
        test.remove(i)
print(test)
```

```
[(4,), (8, 6, 7), (1,), (3, 4, 6, 7)]
```

2. Removing duplicates from tuple ?

The original tuple is : (1, 3, 5, 2, 3, 5, 1, 1, 3)
The tuple after removing duplicates : (1, 3, 5, 2)

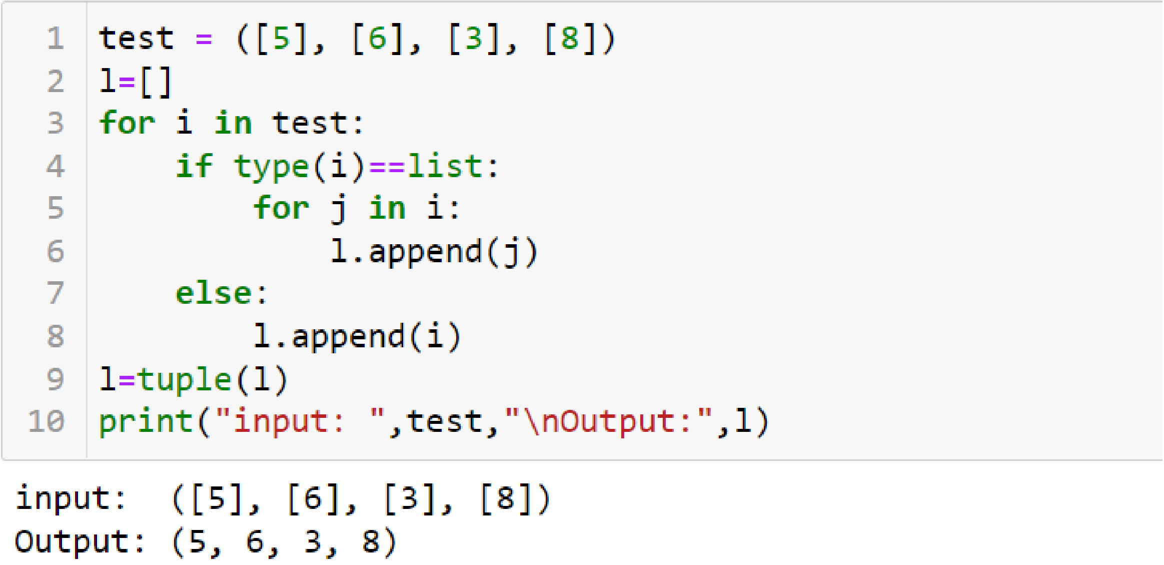


```
T=(1, 3, 5, 2, 3, 5, 1, 1, 3)
print("Input: ",T)
lit=list(T)
lit=set(lit)
T=tuple(lit)
print("Output:",T)
```

```
Input:  (1, 3, 5, 2, 3, 5, 1, 1, 3)
Output: (1, 2, 3, 5)
```

3. Flatten tuple of List to tuple ?

Input : test_tuple = ([5], [6], [3], [8]) Output : (5, 6, 3, 8)
Input : test_tuple = ([5, 7, 8]) Output : (5, 7, 8)

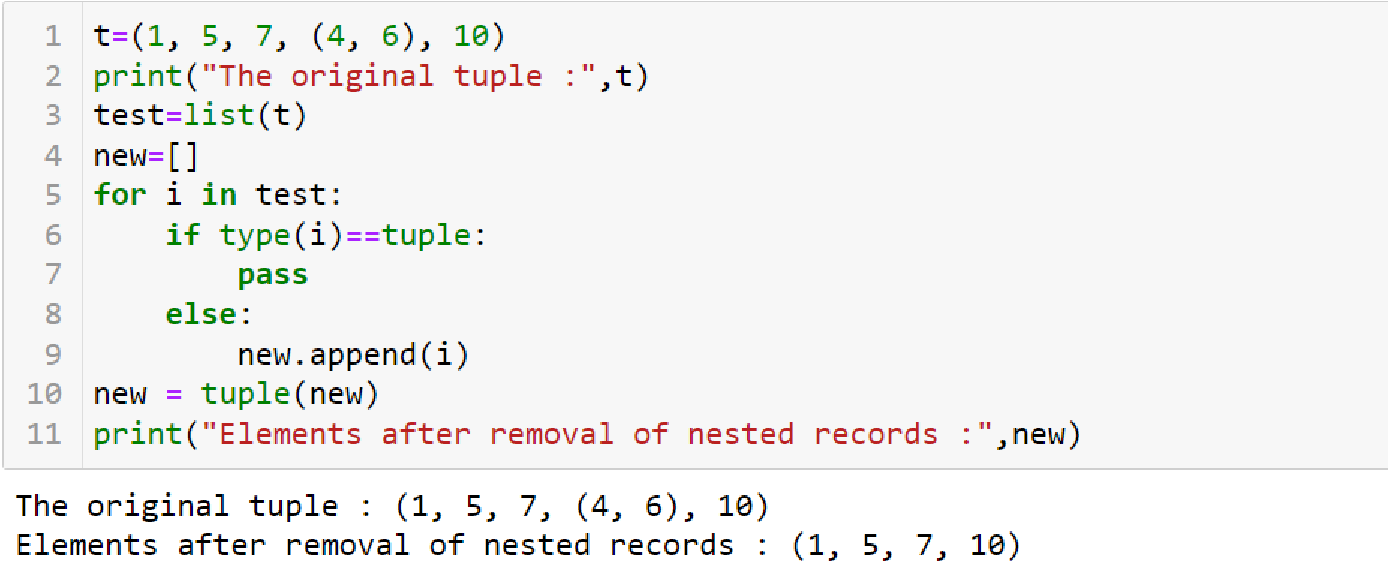


```python
test = ([5], [6], [3], [8])
l=[]
for i in test:
    if type(i)==list:
        for j in i:
            l.append(j)
    else:
        l.append(i)
l=tuple(l)
print("input: ",test,"\nOutput:",l)
```

```
input:  ([5], [6], [3], [8])
Output: (5, 6, 3, 8)
```

4. Remove nested records from tuple?

The original tuple : (1, 5, 7, (4, 6), 10)
Elements after removal of nested records : (1, 5, 7, 10)

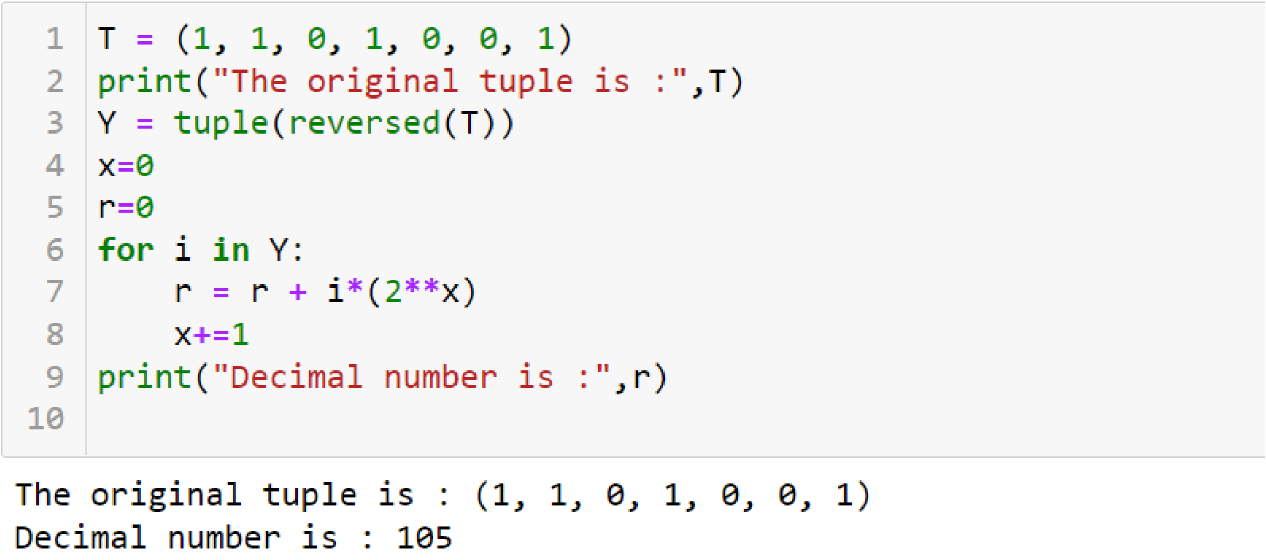


```python
t=(1, 5, 7, (4, 6), 10)
print("The original tuple :",t)
test=list(t)
new=[]
for i in test:
    if type(i)==tuple:
        pass
    else:
        new.append(i)
new = tuple(new)
print("Elements after removal of nested records :",new)
```

```
The original tuple : (1, 5, 7, (4, 6), 10)
Elements after removal of nested records : (1, 5, 7, 10)
```

5. Convert Binary tuple to Integer?

The original tuple is : (1, 1, 0, 1, 0, 0, 1)
Decimal number is : 105

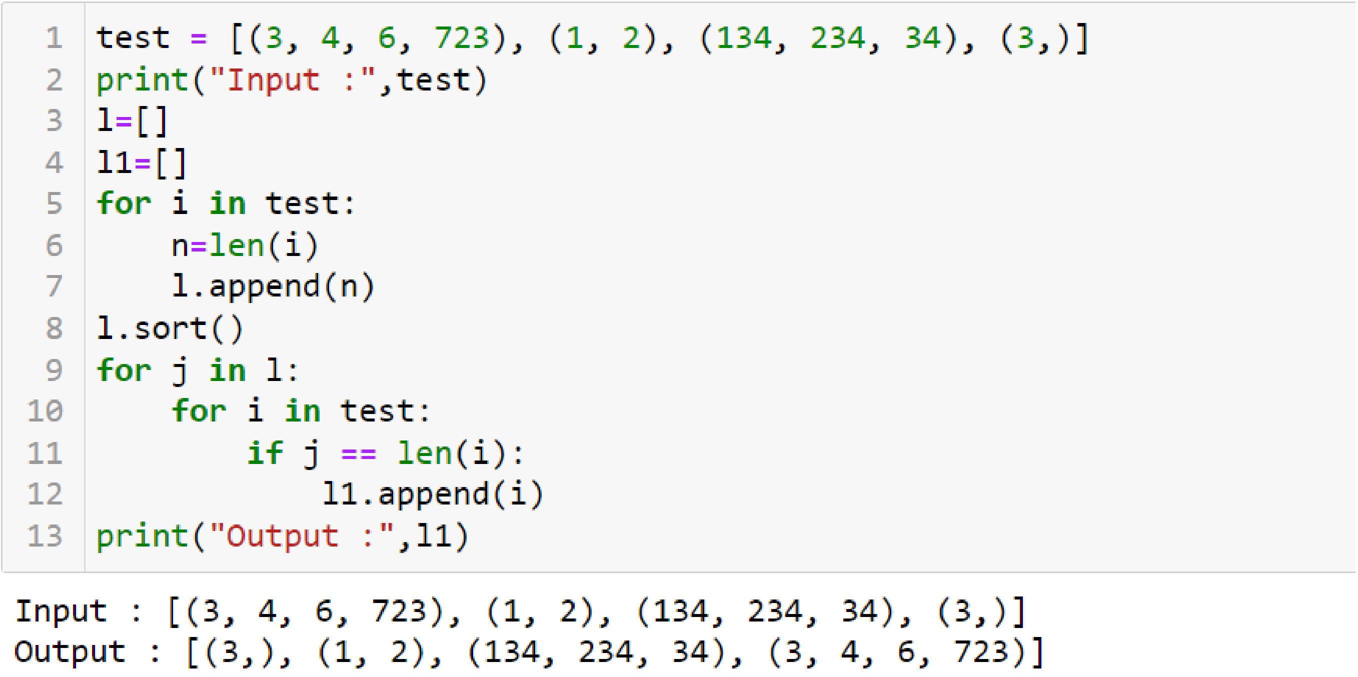


```
T = (1, 1, 0, 1, 0, 0, 1)
print("The original tuple is :",T)
Y = tuple(reversed(T))
x=0
r=0
for i in Y:
    r = r + i*(2**x)
    x+=1
print("Decimal number is :",r)

```

```
The original tuple is : (1, 1, 0, 1, 0, 0, 1)
Decimal number is : 105
```

6. Sort Tuples by Total digits?

Input : test_list = [(3, 4, 6, 723), (1, 2), (134, 234, 34)]
Output : [(1, 2), (3, 4, 6, 723), (134, 234, 34)]
Explanation : 2 < 6 < 8, sorted by increasing total digits.



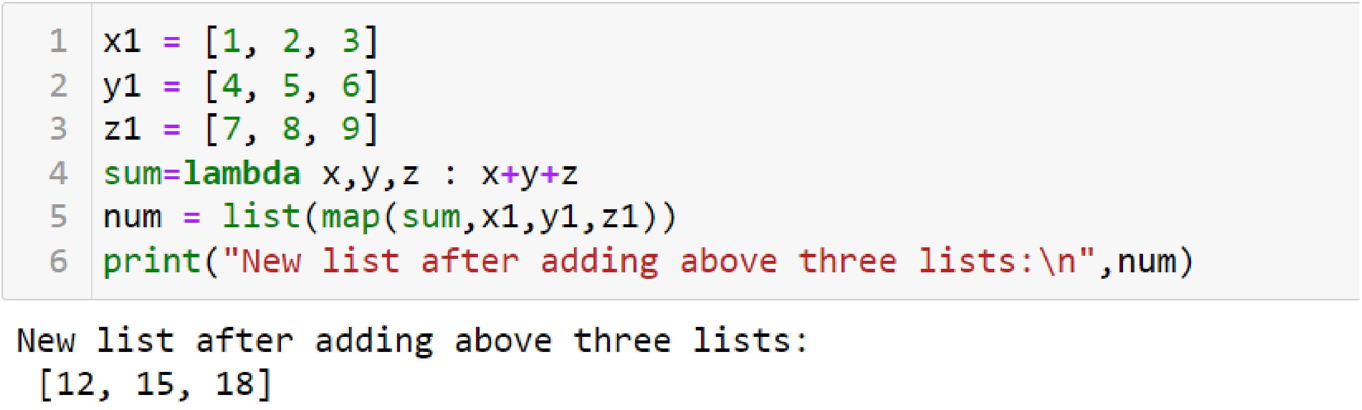
```
test = [(3, 4, 6, 723), (1, 2), (134, 234, 34), (3,)]
print("Input :",test)
l=[]
l1=[]
for i in test:
    n=len(i)
    l.append(n)
l.sort()
for j in l:
    for i in test:
        if j == len(i):
            l1.append(i)
print("Output :",l1)
```

```
Input : [(3, 4, 6, 723), (1, 2), (134, 234, 34), (3,)]
Output : [(3,), (1, 2), (134, 234, 34), (3, 4, 6, 723)]
```

MAP & Lambda Function

1. Write a Python program to add three given lists using Python map and lambda.?

Original list:
[1, 2, 3]
[4, 5, 6]
[7, 8, 9]
New list after adding above three lists:
[12, 15, 18]

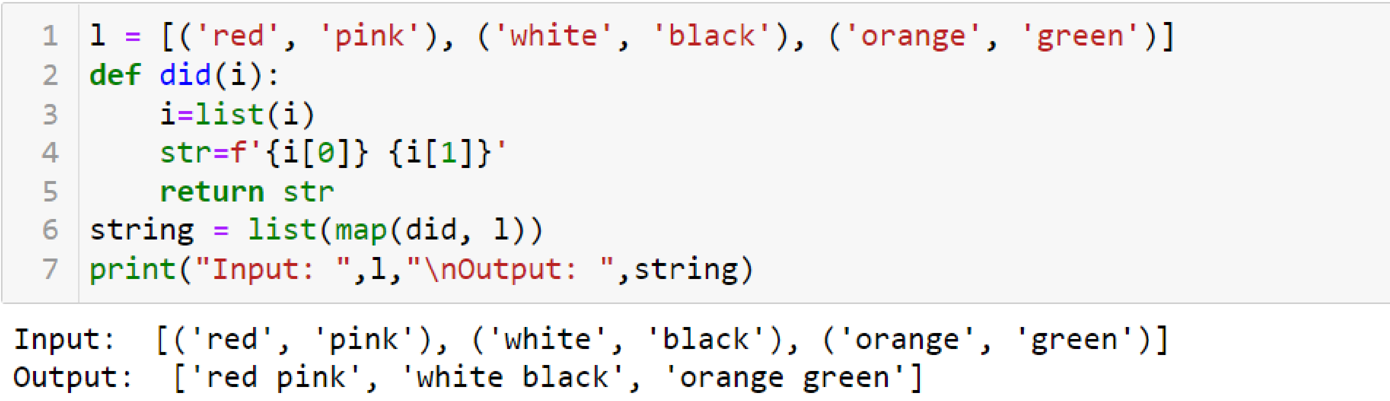


```python
x1 = [1, 2, 3]
y1 = [4, 5, 6]
z1 = [7, 8, 9]
sum=lambda x,y,z : x+y+z
num = list(map(sum,x1,y1,z1))
print("New list after adding above three lists:\n",num)
```

```
New list after adding above three lists:
 [12, 15, 18]
```

2. Write a Python program to convert a given list of tuples to a list of strings using map function?

Original list of tuples:
[('red', 'pink'), ('white', 'black'), ('orange', 'green')]
Convert the said list of tuples to a list of strings:
['red pink', 'white black', 'orange green']

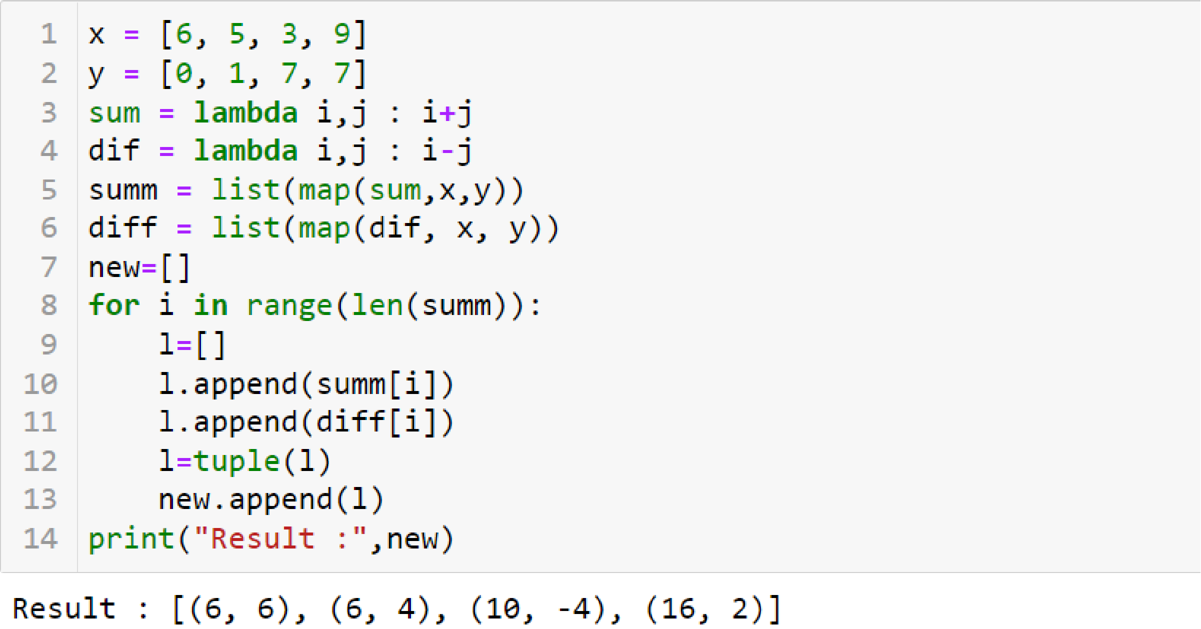


```python
l = [('red', 'pink'), ('white', 'black'), ('orange', 'green')]
def did(i):
    i=list(i)
    str=f'{i[0]} {i[1]}'
    return str
string = list(map(did, l))
print("Input: ",l,"\nOutput: ",string)
```

```
Input:  [('red', 'pink'), ('white', 'black'), ('orange', 'green')]
Output:  ['red pink', 'white black', 'orange green']
```

3. Write a Python program to add two given lists and find the difference between lists. Use map() function?

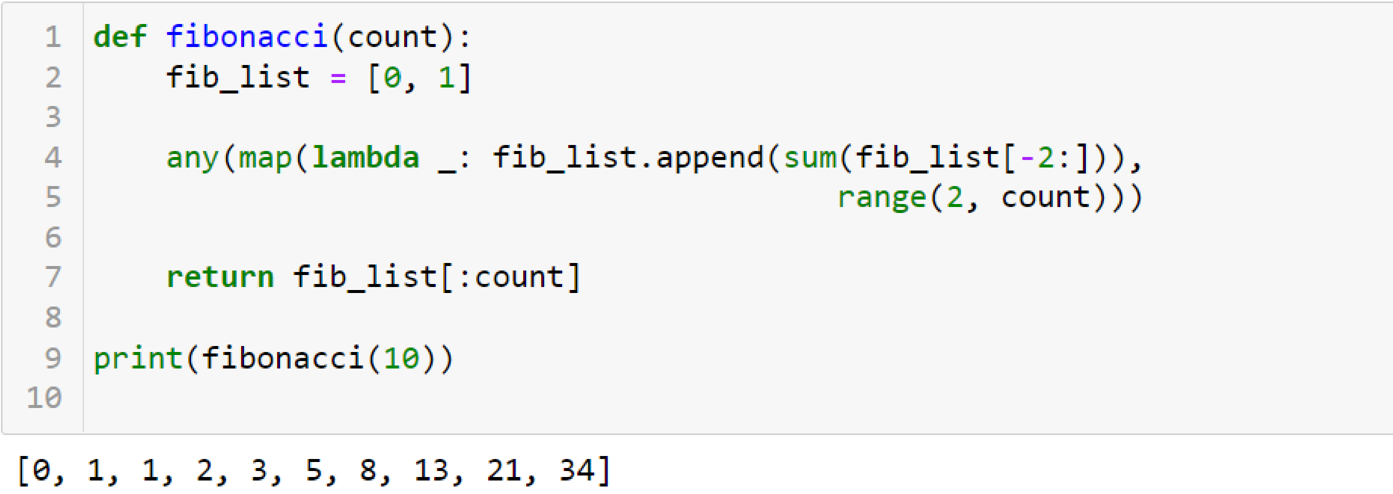
Original lists:
[6, 5, 3, 9]
[0, 1, 7, 7]
Result:
[(6, 6), (6, 4), (10, -4), (16, 2)]



```
x = [6, 5, 3, 9]
y = [0, 1, 7, 7]
sum = lambda i,j : i+j
dif = lambda i,j : i-j
summ = list(map(sum,x,y))
diff = list(map(dif, x, y))
new=[]
for i in range(len(summ)):
    l=[]
    l.append(summ[i])
    l.append(diff[i])
    l=tuple(l)
    new.append(l)
print("Result :",new)
```

```
Result : [(6, 6), (6, 4), (10, -4), (16, 2)]
```

4. Write a Python program to create Fibonacci series upto n using Lambda?

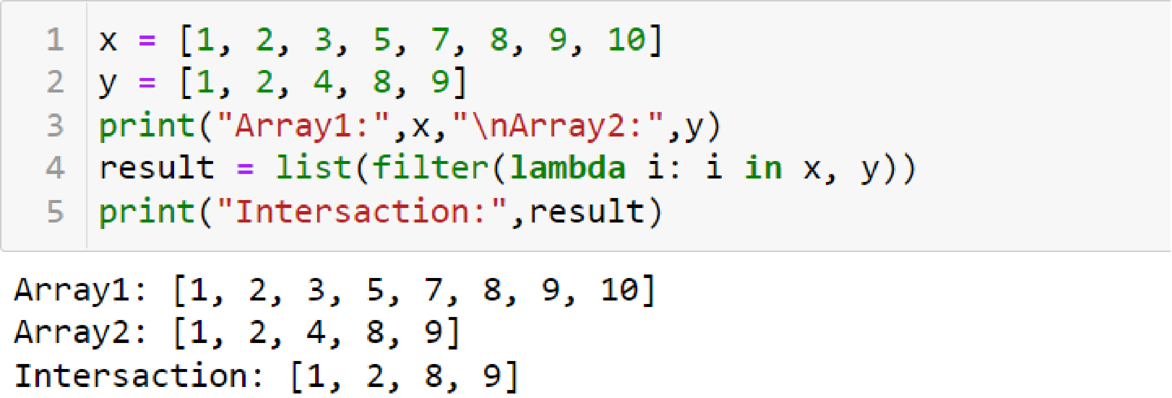


```
def fibonacci(count):
    fib_list = [0, 1]

    any(map(lambda _: fib_list.append(sum(fib_list[-2:])),
                                         range(2, count)))

    return fib_list[:count]

print(fibonacci(10))

```

```
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
```

5. Write a intersection Python program to find of two given arrays using Lambda ?

[1, 2, 3, 5, 7, 8, 9, 10]
[1, 2, 4, 8, 9]
Intersection of the said arrays: [1, 2, 8, 9]

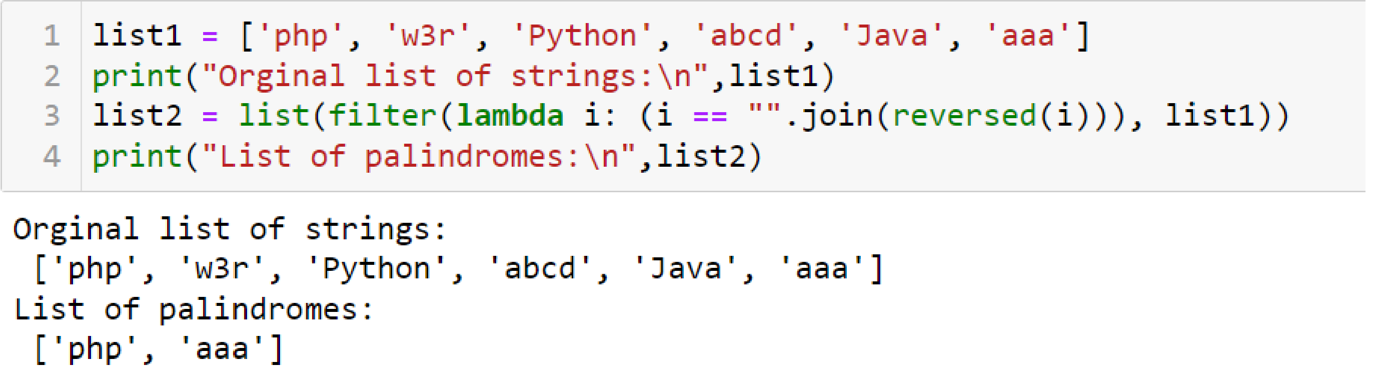


```python
x = [1, 2, 3, 5, 7, 8, 9, 10]
y = [1, 2, 4, 8, 9]
print("Array1:",x,"\nArray2:",y)
result = list(filter(lambda i: i in x, y))
print("Intersaction:",result)
```

```
Array1: [1, 2, 3, 5, 7, 8, 9, 10]
Array2: [1, 2, 4, 8, 9]
Intersaction: [1, 2, 8, 9]
```

6. Write a Python program to find palindromes in a given list of strings using Lambda ?

Orginal list of strings:
['php', 'w3r', 'Python', 'abcd', 'Java', 'aaa']
List of palindromes:
['php', 'aaa']

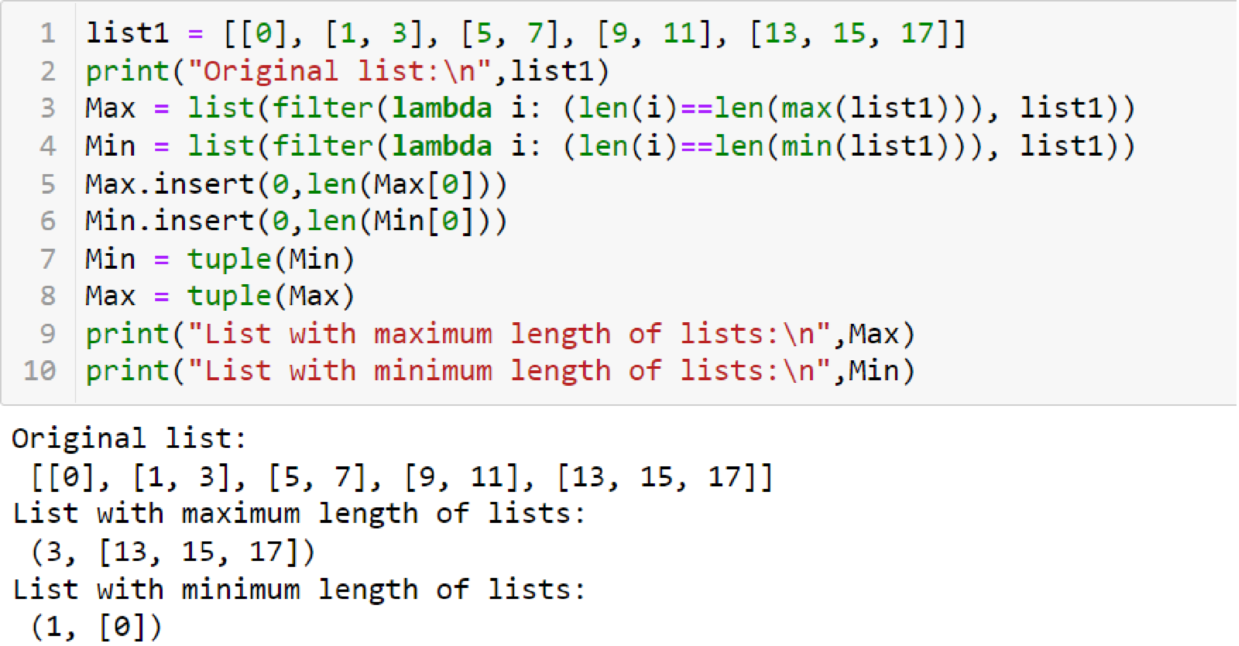


```python
list1 = ['php', 'w3r', 'Python', 'abcd', 'Java', 'aaa']
print("Orginal list of strings:\n",list1)
list2 = list(filter(lambda i: (i == "".join(reversed(i))), list1))
print("List of palindromes:\n",list2)
```

```
Orginal list of strings:
 ['php', 'w3r', 'Python', 'abcd', 'Java', 'aaa']
List of palindromes:
 ['php', 'aaa']
```

7. Write a Python program to find the list with maximum and minimum length using lambda ?

Original list:
[[0], [1, 3], [5, 7], [9, 11], [13, 15, 17]]
List with maximum length of lists:
(3, [13, 15, 17])
List with minimum length of lists:
(1, [0])

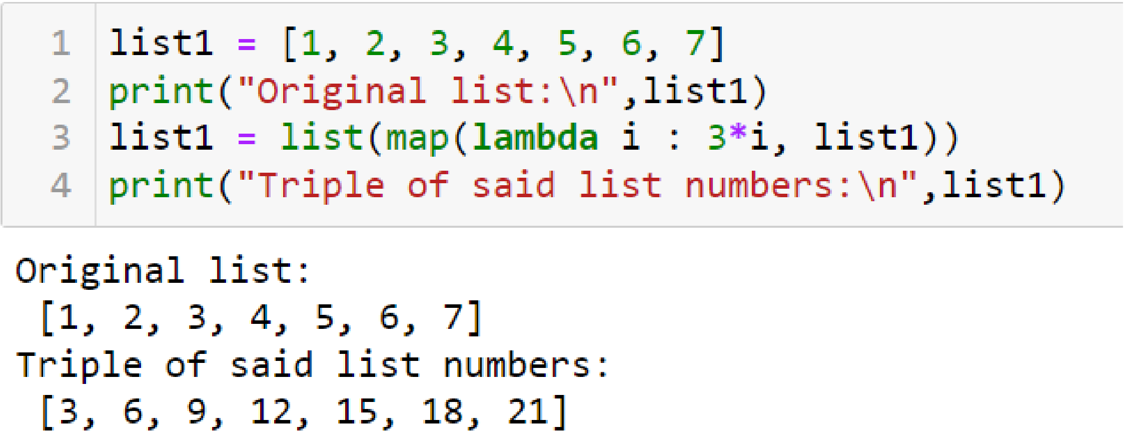


```python
list1 = [[0], [1, 3], [5, 7], [9, 11], [13, 15, 17]]
print("Original list:\n",list1)
Max = list(filter(lambda i: (len(i)==len(max(list1))), list1))
Min = list(filter(lambda i: (len(i)==len(min(list1))), list1))
Max.insert(0,len(Max[0]))
Min.insert(0,len(Min[0]))
Min = tuple(Min)
Max = tuple(Max)
print("List with maximum length of lists:\n",Max)
print("List with minimum length of lists:\n",Min)
```

```
Original list:
 [[0], [1, 3], [5, 7], [9, 11], [13, 15, 17]]
List with maximum length of lists:
 (3, [13, 15, 17])
List with minimum length of lists:
 (1, [0])
```

8. Write a Python program to triple all numbers of a given list of integers.?

Original list: (1, 2, 3, 4, 5, 6, 7)
Triple of said list numbers:
[3, 6, 9, 12, 15, 18, 21]

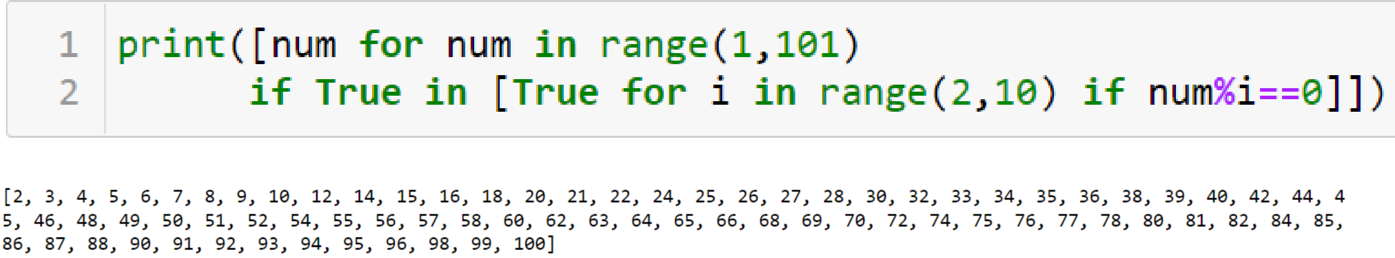


```python
list1 = [1, 2, 3, 4, 5, 6, 7]
print("Original list:\n",list1)
list1 = list(map(lambda i : 3*i, list1))
print("Triple of said list numbers:\n",list1)
```

```
Original list:
 [1, 2, 3, 4, 5, 6, 7]
Triple of said list numbers:
 [3, 6, 9, 12, 15, 18, 21]
```

List Comprehension

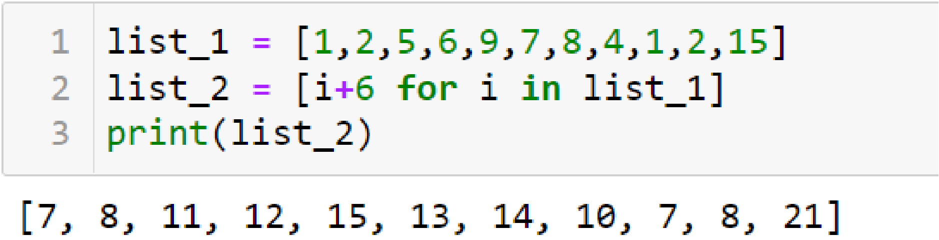
1. Use a nested list comprehension to find all of the numbers from 1–100 that are divisible by any single digit besides 1 (2–9)?



```python
print([num for num in range(1,101)
      if True in [True for i in range(2,10) if num%i==0]])
```

```
[2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 14, 15, 16, 18, 20, 21, 22, 24, 25, 26, 27, 28, 30, 32, 33, 34, 35, 36, 38, 39, 40, 42, 44, 4
5, 46, 48, 49, 50, 51, 52, 54, 55, 56, 57, 58, 60, 62, 63, 64, 65, 66, 68, 69, 70, 72, 74, 75, 76, 77, 78, 80, 81, 82, 84, 85,
86, 87, 88, 90, 91, 92, 93, 94, 95, 96, 98, 99, 100]
```

2. Use list comprehension to construct a new list but add 6 to each item?

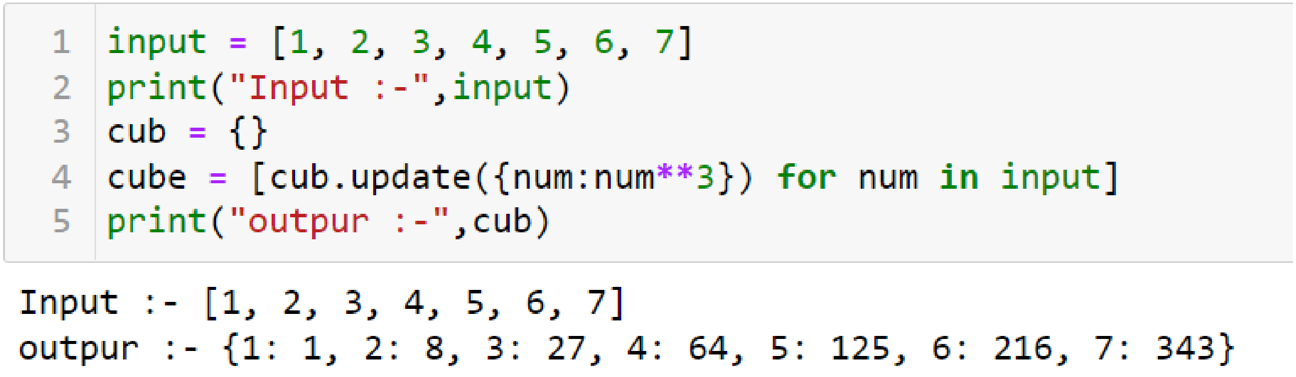


```python
list_1 = [1,2,5,6,9,7,8,4,1,2,15]
list_2 = [i+6 for i in list_1]
print(list_2)
```

```
[7, 8, 11, 12, 15, 13, 14, 10, 7, 8, 21]
```

3. Suppose we want to create an output dictionary which contains only the odd numbers that are present in the input list as keys and their cubes as values. Let’s see how to do this using for loops and dictionary comprehension.?

input = [1, 2, 3, 4, 5, 6, 7]
outpur :- {1: 1, 3: 27, 5: 125, 7: 343}

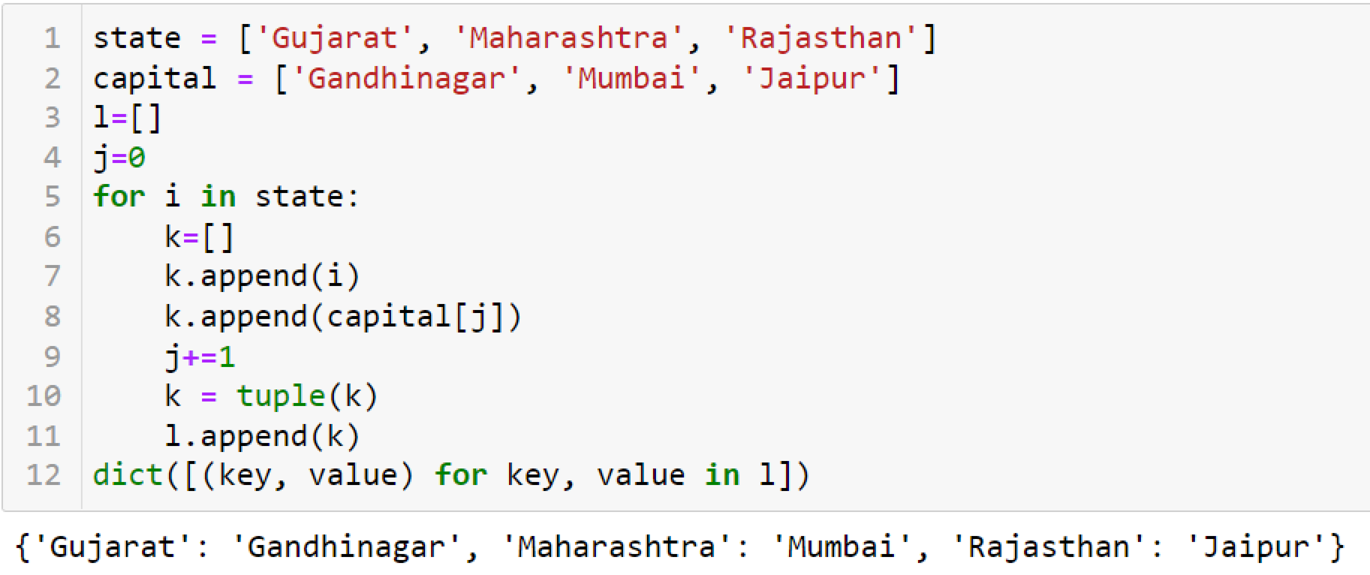


```
input = [1, 2, 3, 4, 5, 6, 7]
print("Input :-",input)
cub = {}
cube = [cub.update({num:num**3}) for num in input]
print("outpur :-",cub)
```

```
Input :- [1, 2, 3, 4, 5, 6, 7]
outpur :- {1: 1, 2: 8, 3: 27, 4: 64, 5: 125, 6: 216, 7: 343}
```

4. Given two lists containing the names of states and their corresponding capitals, construct a dictionary which maps the states with their respective capitals. Let’s see how to do this using for loops and dictionary comprehension.?

state = ['Gujarat', 'Maharashtra', 'Rajasthan']
capital = ['Gandhinagar', 'Mumbai', 'Jaipur']
output:- {'Gujarat': 'Gandhinagar','Maharashtra': 'Mumbai', 'Rajasthan': 'Jaipur'}

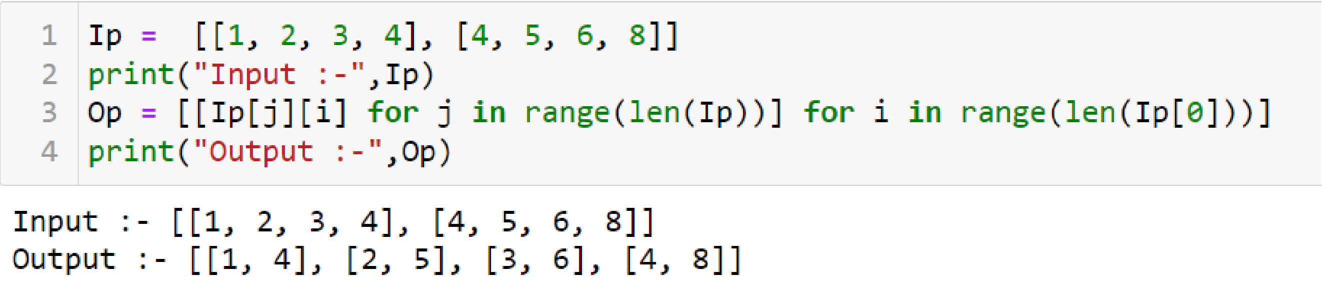


```
state = ['Gujarat', 'Maharashtra', 'Rajasthan']
capital = ['Gandhinagar', 'Mumbai', 'Jaipur']
l=[]
j=0
for i in state:
    k=[]
    k.append(i)
    k.append(capital[j])
    j+=1
    k = tuple(k)
    l.append(k)
dict([(key, value) for key, value in l])
```

```
{'Gujarat': 'Gandhinagar', 'Maharashtra': 'Mumbai', 'Rajasthan': 'Jaipur'}
```

5. Transpose of Matrix using Comprehension?

Input :- [[1, 2, 3, 4], [4, 5, 6, 8]]
Output :- [[1, 4], [2, 5], [3, 6], [4, 8]]

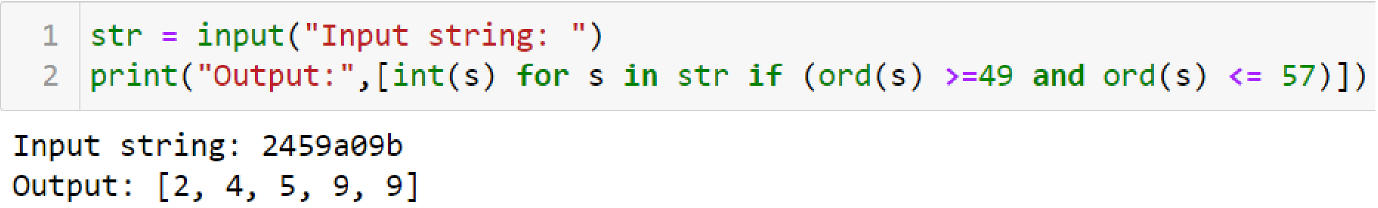


```
Ip =  [[1, 2, 3, 4], [4, 5, 6, 8]]
print("Input :-",Ip)
Op = [[Ip[j][i] for j in range(len(Ip))] for i in range(len(Ip[0]))]
print("Output :-",Op)
```

```
Input :- [[1, 2, 3, 4], [4, 5, 6, 8]]
Output :- [[1, 4], [2, 5], [3, 6], [4, 8]]
```

6. We have a string ‘2459a09b’ and we want to extract all integer literals, and use int() to cast them into integers.?

Input- 2459a09b'
output :- [2, 4, 5, 9, 0, 9]

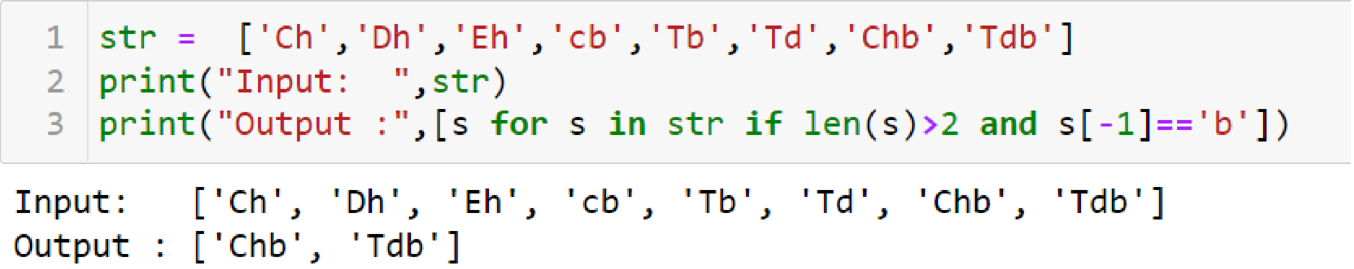


```
str = input("Input string: ")
print("Output:",[int(s) for s in str if (ord(s) >=49 and ord(s) <= 57)])
```

```
Input string: 2459a09b
Output: [2, 4, 5, 9, 9]
```

7. Finding the elements in a list in which elements are ended with the letter ‘b’ and the length of that element is greater than 2?

input :- names = ['Ch','Dh','Eh','cb','Tb','Td','Chb','Tdb']
output :- ['Chb', 'Tdb']

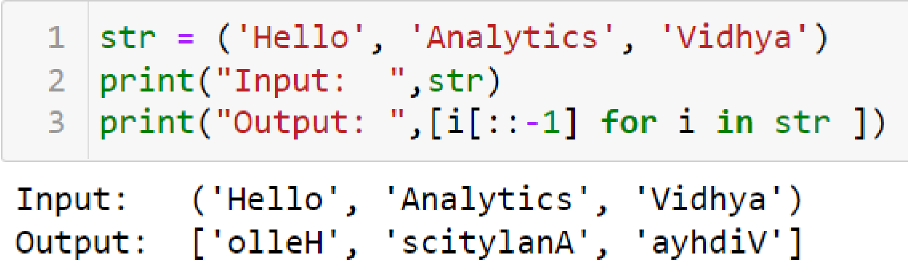


```
str =  ['Ch','Dh','Eh','cb','Tb','Td','Chb','Tdb']
print("Input:  ",str)
print("Output :",[s for s in str if len(s)>2 and s[-1]=='b'])
```

```
Input:   ['Ch', 'Dh', 'Eh', 'cb', 'Tb', 'Td', 'Chb', 'Tdb']
Output : ['Chb', 'Tdb']
```

8. Reverse each String in a Tuple using list comprehension ?

Input :- 'Hello', 'Analytics', 'Vidhya'
output :- ['olleH', 'scitylanA', 'ayhdiV']



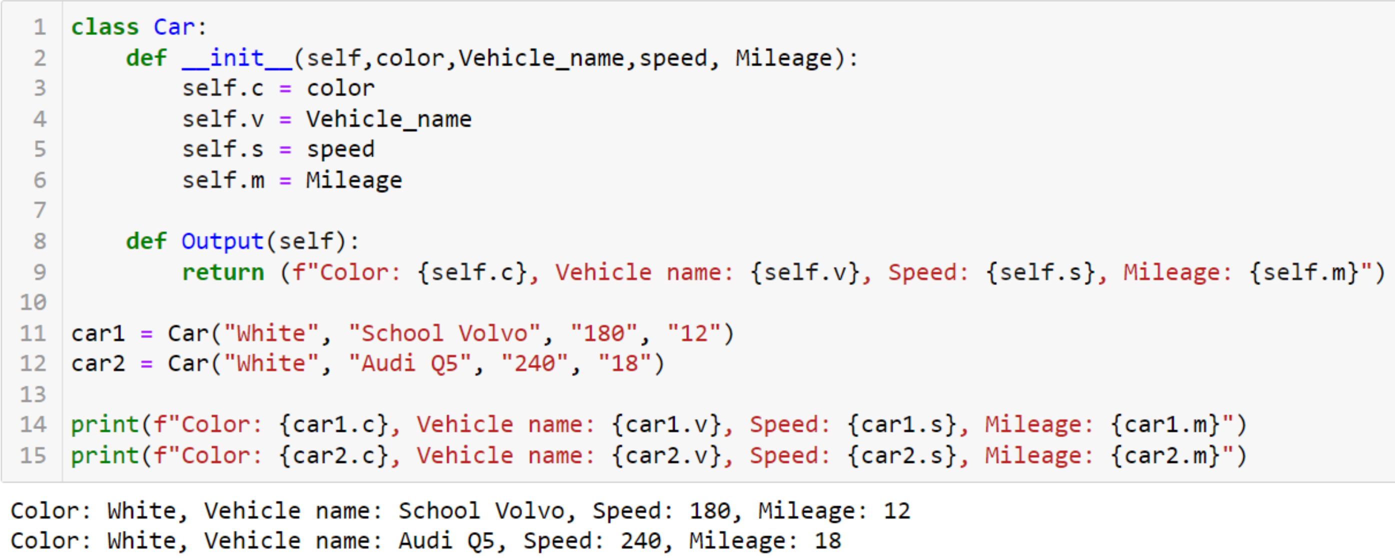
```python
str = ('Hello', 'Analytics', 'Vidhya')
print("Input:  ",str)
print("Output: ",[i[::-1] for i in str ])
```

```
Input:   ('Hello', 'Analytics', 'Vidhya')
Output:  ['olleH', 'scitylanA', 'ayhdiV']
```

Object Oriented Programming

1. Define a property that must have the same value for every class instance (object)?

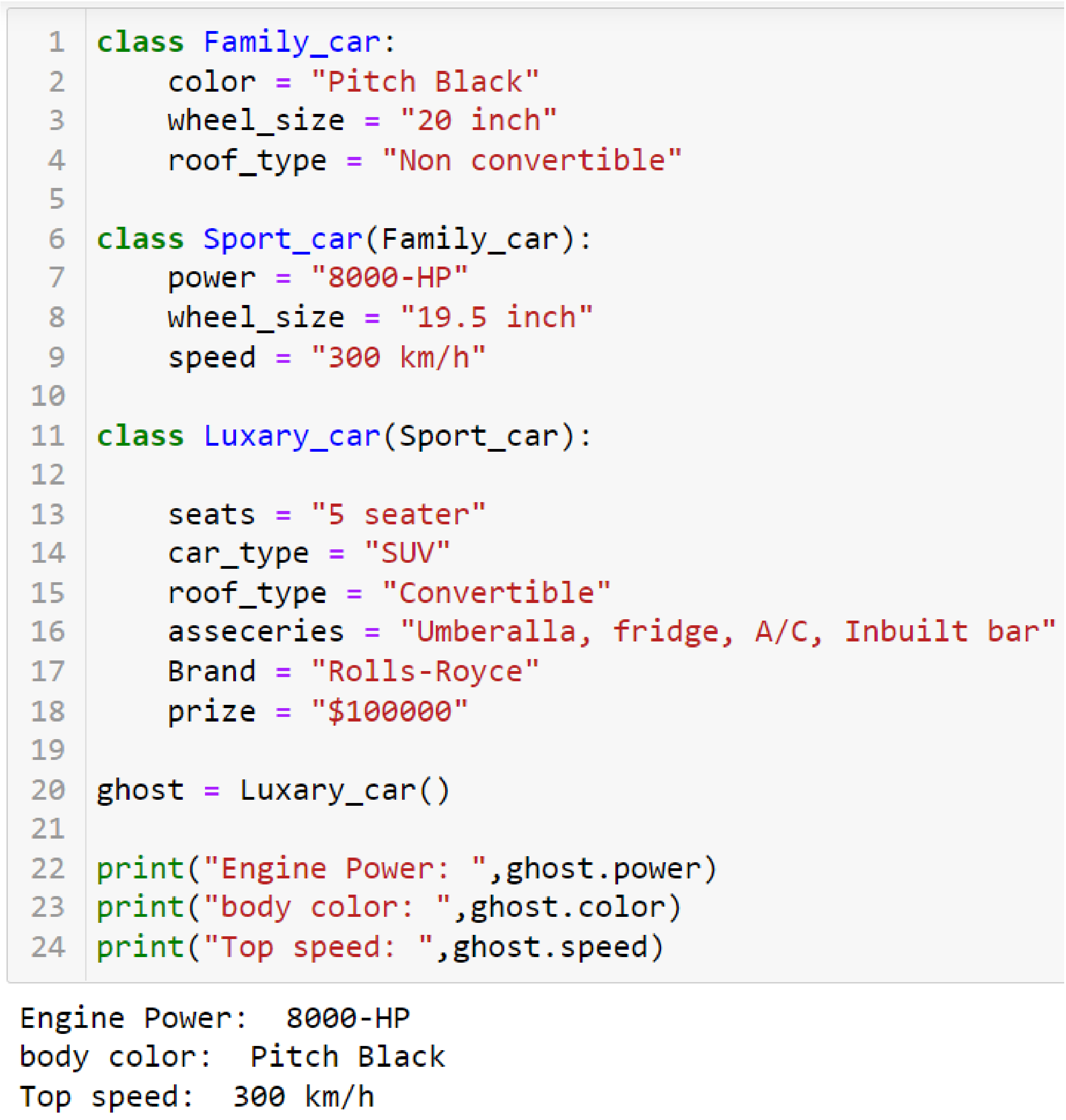
Output :-
Color: White, Vehicle name: School Volvo, Speed: 180, Mileage: 12
Color: White, Vehicle name: Audi Q5, Speed: 240, Mileage: 18



```python
class Car:
    def __init__(self,color,Vehicle_name,speed, Mileage):
        self.c = color
        self.v = Vehicle_name
        self.s = speed
        self.m = Mileage

    def Output(self):
        return (f"Color: {self.c}, Vehicle name: {self.v}, Speed: {self.s}, Mileage: {self.m}")

car1 = Car("White", "School Volvo", "180", "12")
car2 = Car("White", "Audi Q5", "240", "18")

print(f"Color: {car1.c}, Vehicle name: {car1.v}, Speed: {car1.s}, Mileage: {car1.m}")
print(f"Color: {car2.c}, Vehicle name: {car2.v}, Speed: {car2.s}, Mileage: {car2.m}")
```

```
Color: White, Vehicle name: School Volvo, Speed: 180, Mileage: 12
Color: White, Vehicle name: Audi Q5, Speed: 240, Mileage: 18
```

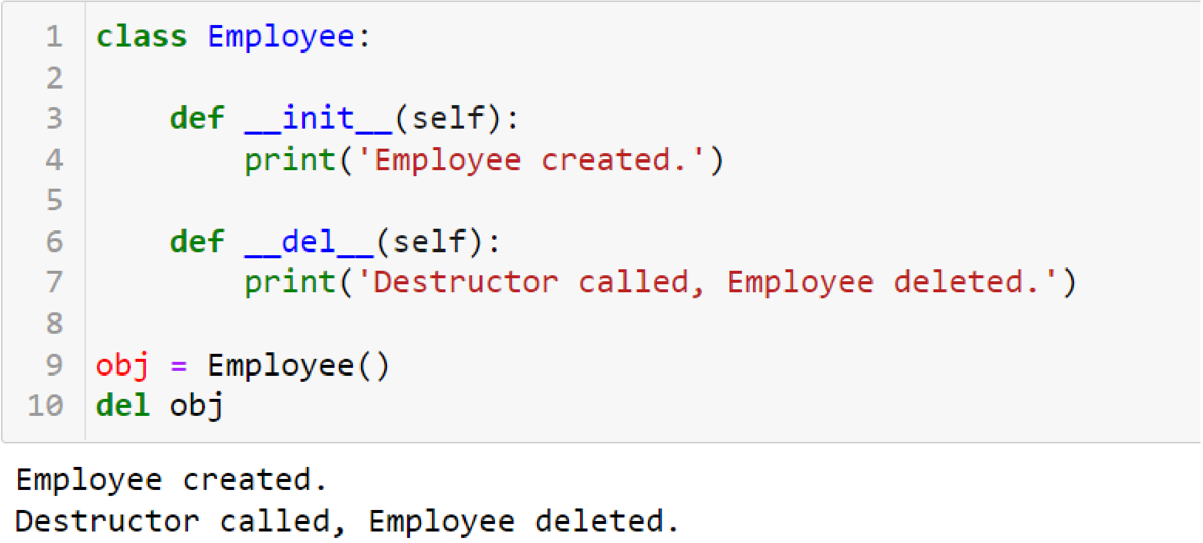
3. Create a class with Multi-level Inheritance ?



```python
class Family_car:
    color = "Pitch Black"
    wheel_size = "20 inch"
    roof_type = "Non convertible"

class Sport_car(Family_car):
    power = "8000-HP"
    wheel_size = "19.5 inch"
    speed = "300 km/h"

class Luxary_car(Sport_car):

    seats = "5 seater"
    car_type = "SUV"
    roof_type = "Convertible"
    asseceries = "Umberalla, fridge, A/C, Inbuilt bar"
    Brand = "Rolls-Royce"
    prize = "$100000"

ghost = Luxary_car()

print("Engine Power: ",ghost.power)
print("body color: ",ghost.color)
print("Top speed: ",ghost.speed)
```

```
Engine Power:  8000-HP
body color:  Pitch Black
Top speed:  300 km/h
```

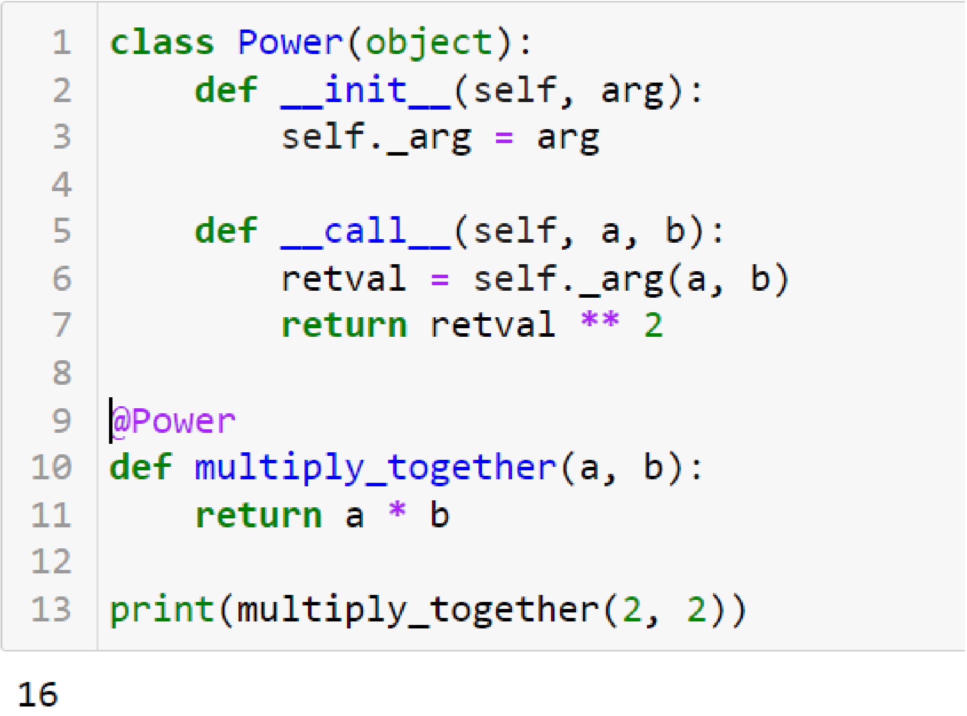
4. Create a class a overwrite a class Destructor ?



```python
class Employee:

    def __init__(self):
        print('Employee created.')

    def __del__(self):
        print('Destructor called, Employee deleted.')

obj = Employee()
del obj
```

```
Employee created.
Destructor called, Employee deleted.
```

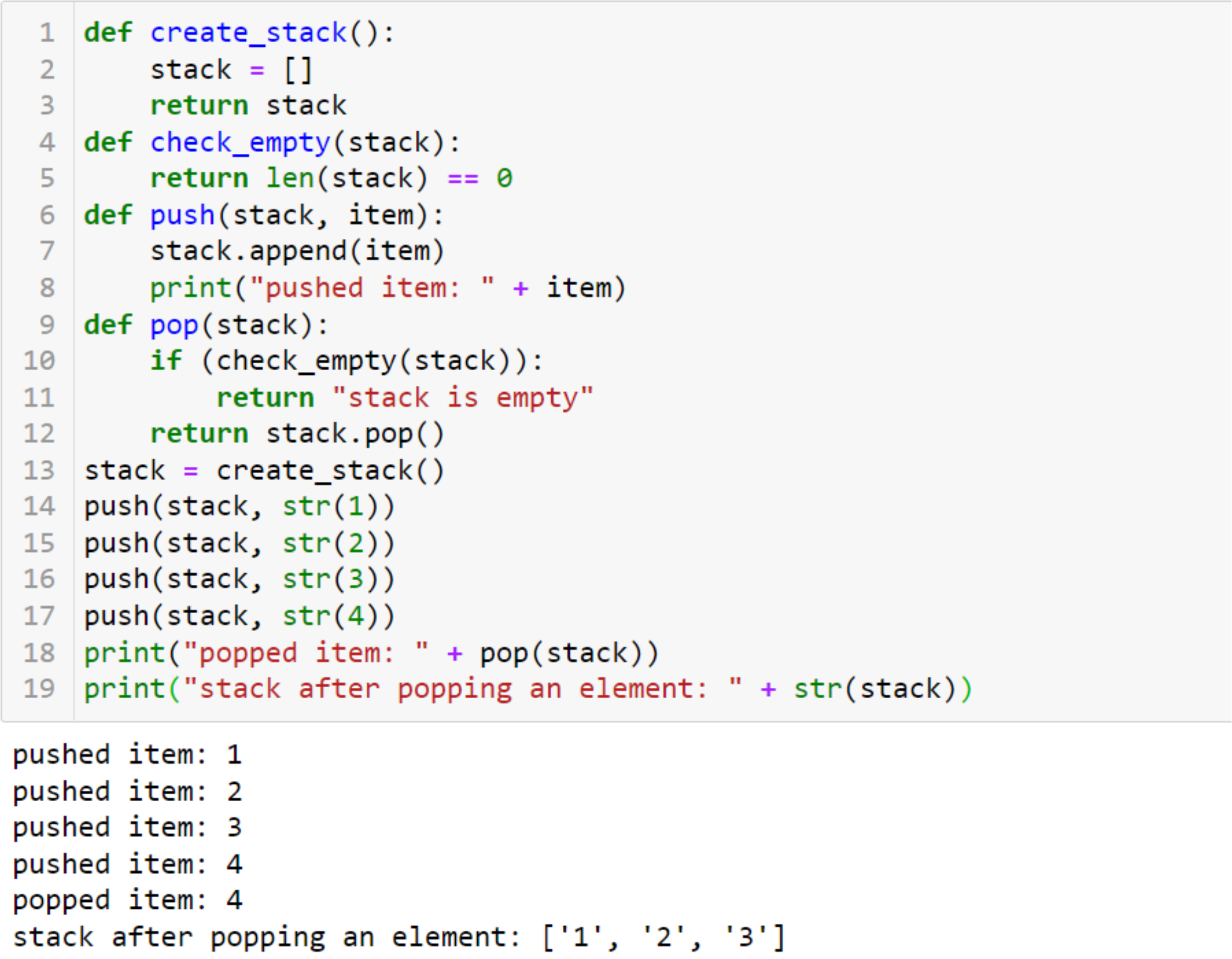
5. for write a decorator a class?



```python
class Power(object):
    def __init__(self, arg):
        self._arg = arg

    def __call__(self, a, b):
        retval = self._arg(a, b)
        return retval ** 2

@Power
def multiply_together(a, b):
    return a * b

print(multiply_together(2, 2))
```

```
16
```

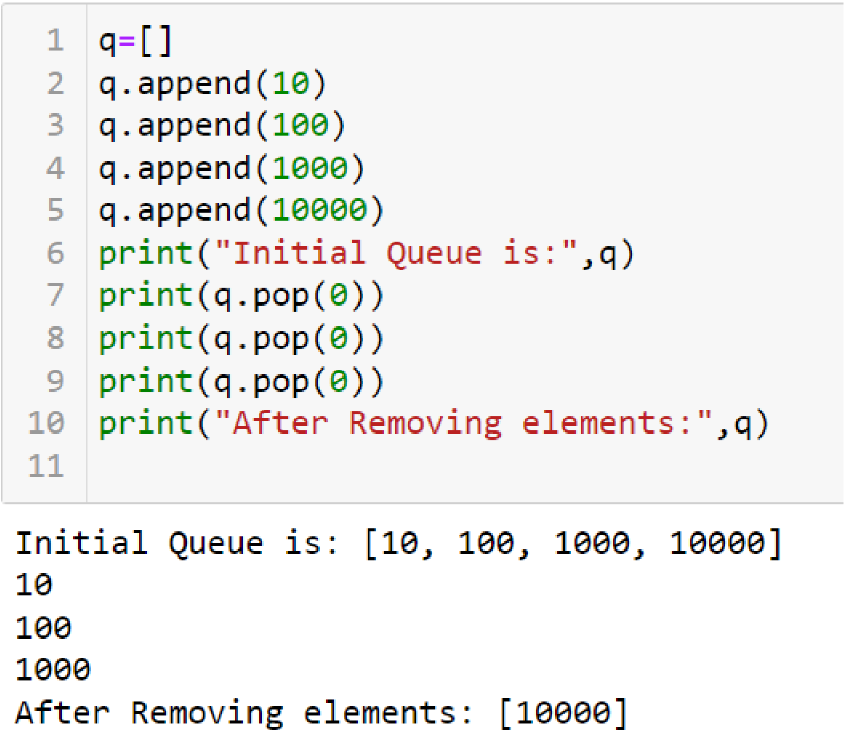
6. Implement a stack ?



```python
def create_stack():
    stack = []
    return stack
def check_empty(stack):
    return len(stack) == 0
def push(stack, item):
    stack.append(item)
    print("pushed item: " + item)
def pop(stack):
    if (check_empty(stack)):
        return "stack is empty"
    return stack.pop()
stack = create_stack()
push(stack, str(1))
push(stack, str(2))
push(stack, str(3))
push(stack, str(4))
print("popped item: " + pop(stack))
print("stack after popping an element: " + str(stack))
```

```
pushed item: 1
pushed item: 2
pushed item: 3
pushed item: 4
popped item: 4
stack after popping an element: ['1', '2', '3']
```

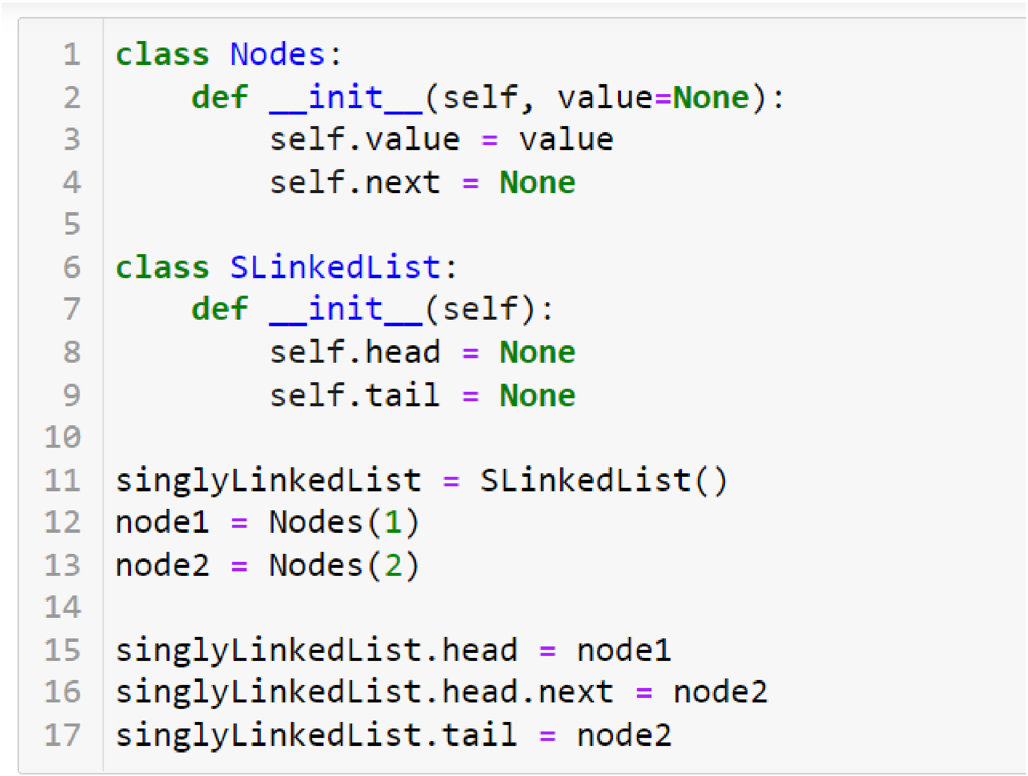
7. Implement a Queue ?



```python
q=[]
q.append(10)
q.append(100)
q.append(1000)
q.append(10000)
print("Initial Queue is:",q)
print(q.pop(0))
print(q.pop(0))
print(q.pop(0))
print("After Removing elements:",q)

```

```
Initial Queue is: [10, 100, 1000, 10000]
10
100
1000
After Removing elements: [10000]
```

8. Implement a Linked List ?



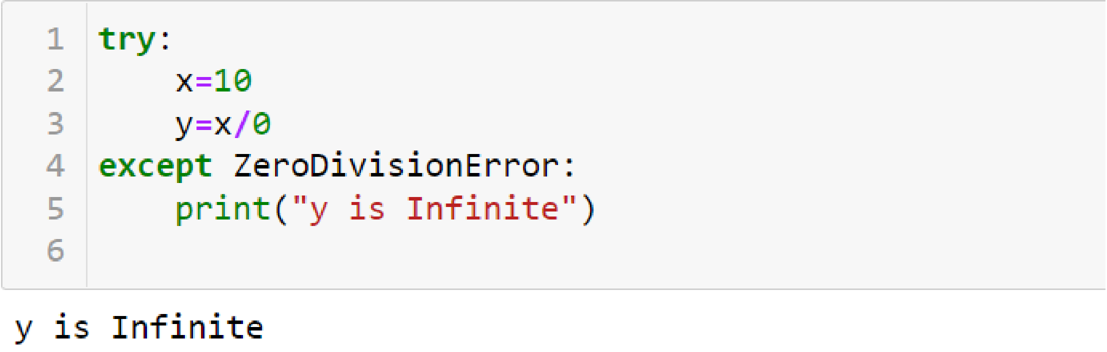
```python
class Nodes:
    def __init__(self, value=None):
        self.value = value
        self.next = None

class SLinkedList:
    def __init__(self):
        self.head = None
        self.tail = None

singlyLinkedList = SLinkedList()
node1 = Nodes(1)
node2 = Nodes(2)

singlyLinkedList.head = node1
singlyLinkedList.head.next = node2
singlyLinkedList.tail = node2
```

9. Explain What happens you create a object of a class (Inside python)?

Each time when you create an object of class the copy of each data variables defined in that class is created. In simple language we can state that each object of a class has its own copy of data members defined in that class.

Exception Handling

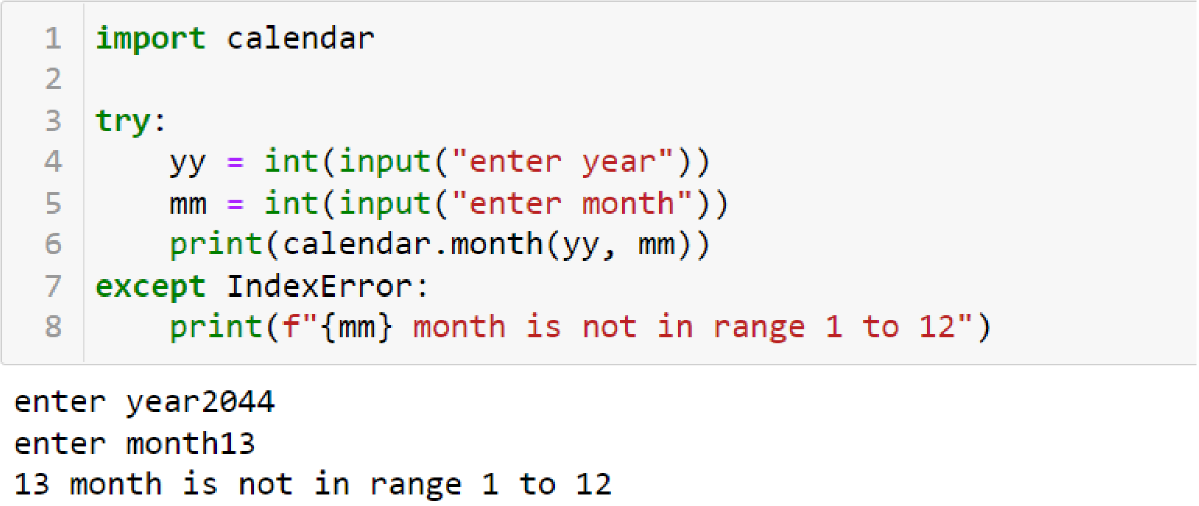
1. Write Custom Exception to handle Zero division Error ?



```
try:
    x=10
    y=x/0
except ZeroDivisionError:
    print("y is Infinite")

```

```
y is Infinite
```

2. Catching specific exception in Python.?



```
import calendar

try:
    yy = int(input("enter year"))
    mm = int(input("enter month"))
    print(calendar.month(yy, mm))
except IndexError:
    print(f"{mm} month is not in range 1 to 12")
```

```
enter year2044
enter month13
13 month is not in range 1 to 12
```