`Problem-1:` Write a Python function that takes a list and returns a new list with unique elements of the first list.

**Exercise 1:**

Input:

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

Output:

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

```
In [27]: def uniquelist(*a):
             b=set(*a)
             a1=list(b)

             return a1
```

```
In [29]: num=[1,2,3,3,3,3,4,5]
         uniquelist(num)

Out[29]: [1, 2, 3, 4, 5]
```

```
In [6]: #its giving desired output now lets see same using NumPy
```

```
In [7]: a=[1,2,3,3,3,4,4,5]
        a1=np.array(a)
        b=np.unique(a1)
        b=list(b)
        print(b)

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

`Problem-2:` Write a Python function that accepts a hyphen-separated sequence of words as parameter and returns the words in a hyphen-separated sequence after sorting them alphabetically.

**Example 1:**

Input:

```
green-red-yellow-black-white
```

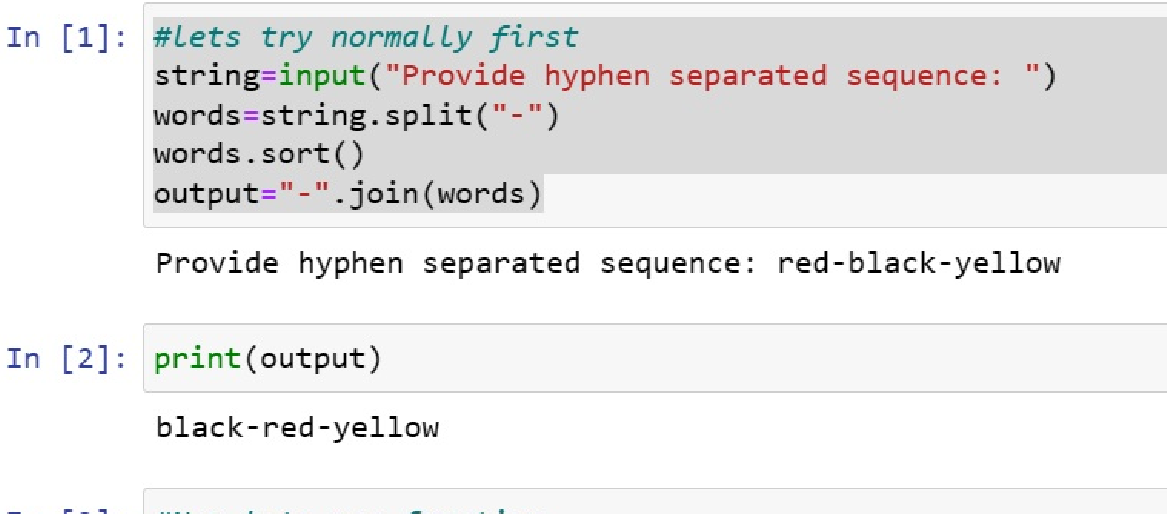
Output:

```
black-green-red-white-yellow
```

```
In [ ]:
```

`Problem 3:` Write a Python function that accepts a string and calculate the number of upper case letters and lower case letters.

```
Sample String : 'CampusX is an Online Mentorship Program fOr EnginEering studentS.'
Expected Output :
No. of Upper case characters :  9
No. of Lower case Characters :  47
```

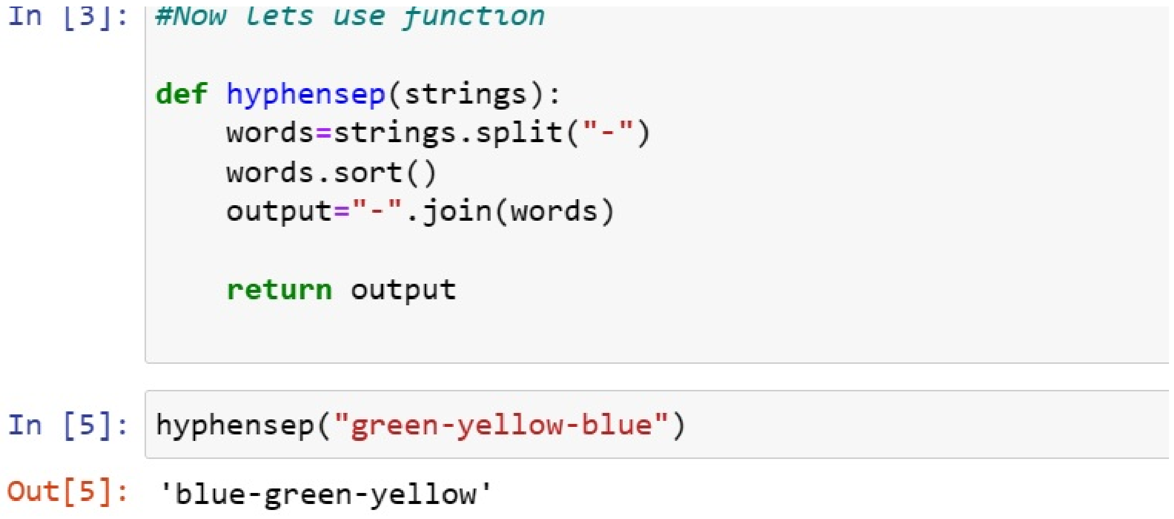


```
In [1]: #lets try normally first
        string=input("Provide hyphen separated sequence: ")
        words=string.split("-")
        words.sort()
        output="-".join(words)

        Provide hyphen separated sequence: red-black-yellow
```

```
In [2]: print(output)

        black-red-yellow
```



```
In [3]: #Now lets use function

        def hyphensep(strings):
            words=strings.split("-")
            words.sort()
            output="-".join(words)

            return output
```

```
In [5]: hyphensep("green-yellow-blue")

Out[5]: 'blue-green-yellow'
```

---

```
In [3]: #sort using numpy
        import numpy as np

        def hyphen(strings):
            string=strings.split("-")
            words="-".join(np.sort(string))

            return words
```

```
In [5]: hyphen("green-blue-red-yellow")

Out[5]: 'blue-green-red-yellow'
```

**`Problem 4:`** Write a Python program to print the even numbers from a given list.

```
Sample List : [1, 2, 3, 4, 5, 6, 7, 8, 9]
Expected Result : [2, 4, 6, 8]
```

```
In [8]: sam=[1, 2, 3, 4, 5, 6, 7, 8, 9]
        even_num=filter(lambda x:x%2==0,sam)
        print(list(even_num))

        [2, 4, 6, 8]
```

```
In [9]: #Can we just sort alphabet using numpy
        a=("ram","hari","shyam")
        b=np.sort(a)
        print(b)

        ['hari' 'ram' 'shyam']
```

**`Problem-6:`** Write a Python function to concatenate any no of dictionaries to create a new one.

```
Sample Dictionary :
dic1={1:10, 2:20}
dic2={3:30, 4:40}
dic3={5:50,6:60}
Expected Result : {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
```

```
In [16]: #firstly, lets try simple and normal way:
         dic1={1:10, 2:20}
         dic2={3:30, 4:40}
         dic3={5:50,6:60}

         def result(*dicts):
             results={}
             for i in dicts:
                 results.update(i)
             return results
```

```
In [17]: result(dic1,dic2,dic3)

Out[17]: {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
```

`Problem-7` Write a python function that accepts a string as input and returns the word with most occurence.

```
Input:
```

```
hello how are you i am fine thank you

Output
you -> 2
```

In [23]:

```python
from collections import Counter

def sentence(string):
    words=string.split()
    counter=Counter(words)
    most_common=counter.most_common(1)
    return most_common

```

In [24]:

```python
sentence("hello how are you i am fine thank you")
```

Out[24]:

```
[('you', 2)]
```

## `Problem 11:` Write a Python program to add three given lists using Python map and lambda.

In [1]:

```python
list1=[1,2,3]
list2=[4,5,6]
list3=[7,8,9]

result=list(map(lambda a,b,c:a+b+c, list1, list2,list3))
print(result)
```

```
[12, 15, 18]
```

## `Problem-13` Using filter() and list() functions and .lower() method filter all the vowels in a given string.

In [25]:

```python
string=["a","E","r","s","t","i","o"]
vowel=["a","e","i","o","u"]

vowel_word=filter(lambda x:x.lower() in vowel,string)
```

In [26]:

```python
print(list(vowel_word))
```

```
['a', 'E', 'i', 'o']
```

`Problem-14` : Use reduce to convert a 2D list to 1D

In [5]:

```python
from functools import reduce
l1=[[1,2,3],[4,5,6]]
flatlist= reduce(lambda x,y:x+y,l1)
```

In [6]:

```python
print(flatlist)
```

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

`Problem 15` - A dictionary contains following information about 5 employees:

- First name
- Last name
- Age
- Grade(Skilled,Semi-skilled,Highly skilled)
  Write a program using map/filter/reduce to a list of employees(first name + last name) who are highly skilled

In [ ]:

```python
# Write code here
employees = [
    {
        'fname':'Nitish',
        'lname':'Singh',
        'age' : 33,
        'grade':'skilled'
    },
    {
        'fname':'Ankit',
        'lname':'Verma',
        'age' : 34,
        'grade':'semi-skilled'
    },
    {
        'fname':'Neha',
        'lname':'Singh',
        'age' : 35,
        'grade':'highly-skilled'
    },
    {
        'fname':'Anurag',
        'lname':'Kumar',
        'age' : 30,
        'grade':'skilled'
    },
    {
        'fname':'Abhinav',
        'lname':'Sharma',
        'age' : 37,
        'grade':'highly-skilled'
    }

]
```

In [ ]:

```
list(map(lambda x:x['fname'] + ' ' + x['lname'],list(filter(lambda x:True if x['grade'] == 'highly-skilled' els
```

Out[ ]:

```
['Neha Singh', 'Abhinav Sharma']
```