1. **Given a list, url = [**[**www.annauniv.edu**](http://www.annauniv.edu)**,** [**www.google.com**](http://www.google.com)**,** [**www.ndtv.com**](http://www.ndtv.com)**,** [**www.website.org**](http://www.website.org)**,** [**www.bis.org.in**](http://www.bis.org.in)**,** [**www.rbi.org.in**](http://www.rbi.org.in)**]; Sort the list based on the top level domain (edu, com, org, in) using custom sorting**

**Solution:**

DomainList=["edu","com","org","in"]

urlList = ["www.annauniv.edu", "www.google.com", "www.ndtv.com", "www.website.org", "www.bis.org.in", "www.rbi.org.in"]

def sortDomainNames(domainList,urlList):

tempDomainList = domainList

tempUrlList = urlList

sortedUrlList = []

for i,v in enumerate(tempDomainList):

for i1,v1 in enumerate(tempUrlList):

if v1.endswith(v):

sortedUrlList.append(v1)

print sortedUrlList

sortDomainNames(DomainList,urlList)

1. **Given a list of strings, return the count of the number of strings where the string length is 2 or more and the first and last chars of the string are the same.**
2. **['axa', 'xyz', 'gg', 'x', 'yyy']**

Solution:

list=['axa', 'xyz', 'gg', 'x', 'yyy']

count=0

for i in list:

if (len(i)>=2 )and (i[0]==i[len(i)-1]):

count=count+1

print ("count is: ",count)

1. **['x', 'cd', 'cnc', 'kk']**

Solution:

list=['x', 'cd', 'cnc', 'kk']

count=0

for i in list:

if (len(i)>=2 )and (i[0]==i[len(i)-1]):

count=count+1

print ("count is: ",count)

1. **['bab', 'ce', 'cba', 'syanora']**

Solution:

list=['bab', 'ce', 'cba', 'syanora']

count=0

for i in list:

if (len(i)>=2 )and (i[0]==i[len(i)-1]):

count=count+1

print ("count is: ",count)

1. **Given a list of strings, return a list with the strings in sorted order, except group all the strings that begin with 'x' first. e.g. ['mix', 'xyz', 'apple', 'xanadu', 'aardvark'] yields**

**['xanadu', 'xyz', 'aardvark', 'apple', 'mix'].**

**Hint: this can be done by making 2 lists and sorting each of them before combining them.**

1. **['bbb', 'ccc', 'axx', 'xzz', 'xaa']**

Solution:

list=['bbb', 'ccc', 'axx', 'xzz', 'xaa']

list1=[]

list2=[]

for i in list:

if(i[0]=='x'):

list1.append(i)

else:

list2.append(i)

list1.sort()

list2.sort()

print (list1+list2)

1. **['mix', 'xyz', 'apple', 'xanadu', 'aardvark']**

Solution:

list=['mix', 'xyz', 'apple', 'xanadu', 'aardvark']

list1=[]

list2=[]

for i in list:

if(i[0]=='x'):

list1.append(i)

else:

list2.append(i)

list1.sort()

list2.sort()

print (list1+list2)

1. **Given a list of non-empty tuples, return a list sorted in increasing order by the last element in each tuple.**

**e.g. [(1, 7), (1, 3), (3, 4, 5), (2, 2)] yields [(2, 2), (1, 3), (3, 4, 5), (1, 7)]**

**Hint: use a custom key= function to extract the last element form each tuple.**

1. **[(1, 3), (3, 2), (2, 1)]**

**Solution:**

list=[(1, 3), (3, 2), (2, 1)]

def lastdig(tuple):

return tuple[-1]

def customSort(lst):

temp=lst

temp.sort(key=lastdig)

return temp

print(customSort(list)

1. **[(1, 7), (1, 3), (3, 4, 5), (2, 2)]**

**Solution:**

list=[(1, 7), (1, 3), (3, 4, 5), (2, 2)]

def lastdig(tuple):

return tuple[-1]

def customSort(lst):

temp=lst

temp.sort(key=lastdig)

return temp

print(customSort(list))

1. **Given a list of numbers, return a list where all adjacent == elements have been reduced to a single element, so [1, 2, 2, 3] returns [1, 2, 3]. You may create a new list or modify the passed in list**.

**i. [1, 2, 2, 3],**

Solution:

list=[2,2,3,3,3]

new\_list=[]

def remove\_duplicate(x):

for i in x:

if(i not in new\_list):

new\_list.append(i)

return new\_list

final=remove\_duplicate(list)

print(final)

1. **[2, 2, 3, 3, 3]**

Solution:

list=[2,2,3,3,3]

new\_list=[]

def remove\_duplicate(x):

for i in x:

if(i not in new\_list):

new\_list.append(i)

return new\_list

final=remove\_duplicate(list)

print(final)

1. **Write a function to print the information in the dictionary(bookstore) in the given format**

**bookstore={"New Arrivals":{"COOKING":["Everyday Italian","Giada De Laurentiis","2005","30.00"],"CHILDREN":["Harry Potter”, J K. Rowling","2005","29.99"],"WEB":["Learning XML","Erik T. Ray","2003","39.95"]}}**

**BOOKSTORE**

**'Learning XML', 'Erik T. Ray', '2003', '39.95'**

**'Everyday Italian', 'Giada De Laurent is', '2005', '30.00']**

**'Harry Potter', 'J K. Rowling', '2005', '29.99']**

**Solution:**

bookstore={"New Arrivals":{"COOKING":["Everyday Italian","Giada DeLaurentiis","2005","30.00"],"CHILDREN":["Harry Potter”, J K.Rowling","2005","29.99"],"WEB":["Learning XML","Erik T.Ray","2003","39.95"]}}

for i,j in bookstore.items():

#print(bookstore[i].keys())

print("BOOKSTORE\n")

s=[]

s=dict(bookstore[i].items())

#print(s)

for key, value in reversed(sorted(s.items())):

print(value)

1. **Given a string, str1=""”Python is a widely used high-level programming language for general-purpose programming, created by Guido van Rossum and first released in 1991. An interpreted language, Python has a design philosophy which emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly braces or keywords), and a syntax which allows programmers to express concepts in fewer lines of code than possible in languages such as C++ or Java. The language provides constructs intended to enable writing clear programs on both a small and large scale .Python features a dynamic type system and automatic memory management and supports multiple programming paradigms, including object-oriented, imperative, functional programming, and procedural styles. It has a large and comprehensive standard library. Python interpreters are available for many operating systems, allowing Python code to run on a wide variety of systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit Python Software Foundation."""**
   1. **Build a dictionary, with "words as key" --> Frequency of occurrence as value**

**E.g. Python 🡪7, is🡪3**

**Solution:**

text ="""Python is a widely used high-level programming language for

general-purpose programming, created by Guido van Rossum and first

released in 1991. An interpreted language, Python has a design

philosophy which emphasizes code readability (notably using whitespace

indentation to delimit code blocks rather than curly braces or

keywords), and a syntax which allows programmers to express concepts in

fewer lines of code than possible in languages such as C++ or Java. The

language provides constructs intended to enable writing clear programs

on both a small and large scale .Python features a dynamic type system

and automatic memory management and supports multiple programming

paradigms, including object-oriented, imperative, functional

programming, and procedural styles. It has a large and comprehensive

standard library. Python interpreters are available for many operating

systems, allowing Python code to run on a wide variety of systems.

CPython, the reference implementation of Python, is open source software

and has a community-based development model, as do nearly all of its

variant implementations. CPython is managed by the non-profit Python

Software Foundation."""

dic = dict()

for w in text.split():

if w in dic.keys():

dic[w] = dic[w]+1

else:

dic[w] = 1

print(dic)

* 1. **Print the top 5 words with their frequency of occurrence**

**Solution:**

text ="""Python is a widely used high-level programming language for

general-purpose programming, created by Guido van Rossum and first

released in 1991. An interpreted language, Python has a design

philosophy which emphasizes code readability (notably using whitespace

indentation to delimit code blocks rather than curly braces or

keywords), and a syntax which allows programmers to express concepts in

fewer lines of code than possible in languages such as C++ or Java. The

language provides constructs intended to enable writing clear programs

on both a small and large scale .Python features a dynamic type system

and automatic memory management and supports multiple programming

paradigms, including object-oriented, imperative, functional

programming, and procedural styles. It has a large and comprehensive

standard library. Python interpreters are available for many operating

systems, allowing Python code to run on a wide variety of systems.

CPython, the reference implementation of Python, is open source software

and has a community-based development model, as do nearly all of its

variant implementations. CPython is managed by the non-profit Python

Software Foundation."""

dic = dict()

for w in text.split():

if w in dic.keys():

dic[w] = dic[w]+1

else:

dic[w] = 1

for i in range (0,5):

max\_key = max(dic, key=dic.get)

print(max\_key, "-", dic[max\_key])

del dic[max\_key]

**output:**

a - 8

and - 8

Python - 5

code - 4

to – 4

1. **Given a string, str1=""”Python is a widely used high-level programming language for general-purpose programming, created by Guido van Rossum and first released in 1991. An interpreted language, Python has a design philosophy which emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly braces or keywords), and a syntax which allows programmers to express concepts in fewer lines of code than possible in languages such as C++ or Java. The language provides constructs intended to enable writing clear programs on both a small and large scale .Python features a dynamic type system and automatic memory management and supports multiple programming paradigms, including object-oriented, imperative, functional programming, and procedural styles. It has a large and comprehensive standard library. Python interpreters are available for many operating systems, allowing Python code to run on a wide variety of systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit Python Software Foundation."""**

**Hint: Assume that the first word is preceded by " "**

* 1. **Build a dictionary where the key is a word and the value is the list of words that are likely to follow.**
     1. **E.g. Python 🡪 [is, has, features, interpreters, code, Software]. In this example the words listed are likely to follow “Python”**

**Solution:**

str1="""Python is a widely used high-level programming language for general-purpose programming, created by Guido van Rossum and first released in 1991. An interpreted language, Python has a design philosophy which emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly braces or keywords), and a syntax which allows programmers to express concepts in fewer lines of code than possible in languages such as C++ or Java. The language provides constructs intended to enable writing clear programs on both a small and large scale .Python features a dynamic type system and automatic memory management and supports multiple programming paradigms, including object-oriented, imperative, functional programming, and procedural styles. It has a large and comprehensive standard library. Python interpreters are available for many operating systems, allowing Python code to run on a wide variety of systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit Python Software Foundation."""

list=[]

list=str1.split()

i=0

dict={}

for i in range(0,len(list)-1,2):

print(i)

dict[list[i]]=(list[i+1])

i=i+2

print(i)

print(dict)

1. **Below is the output of # show ip interface brief command on a router**

**Interface IP-Address OK? Method Status Protocol**

**FastEthernet0/0 192.168.1.242 YES manual up up**

**FastEthernet1/0 unassigned YES unset down**

**Serial2/0 192.168.1.250 YES manual up up**

**Serial3/0 192.168.1.233 YES manual up up**

**FastEthernet4/0 unassigned YES unset down**

**FastEthernet5/0 unassigned YES unset down**

* 1. **Use regular expressions to extract and display Interface and method status for all the interfaces.**
     1. **E.g. FastEthernet0/0, manual up**

**Solution:**

import re

str1="""Interface IP-Address OK? Method Status Protocol

FastEthernet0/0 192.168.1.242 YES manual up up

FastEthernet1/0 unassigned YES unset down

Serial2/0 192.168.1.250 YES manual up up

Serial3/0 192.168.1.233 YES manual up up

FastEthernet4/0 unassigned YES unset down

FastEthernet5/0 unassigned YES unset down"""

print("\n\n\n")

for line in str1.splitlines():

matchObj = re.match( r'(\w+\d\/\d)\s+[.0-9a-z]+\s+\w+\s+(\w+\s?\w+?)\s+\w+', line, re.M|re.I)

if matchObj:

print matchObj.group(1),",",matchObj.group(2),"\n"

print ("\n\n\n")