|  |
| --- |
| Question 1: |
|  |

Define a class with a generator which can iterate the numbers, which are divisible by 7, between a given range 0 and n.

Ans: class iterator(object):  
 *"""docstring for iterator"""* def \_\_init\_\_(self, n):  
 self.n = n  
  
 def divBySeven(self):  
 for i in range(0, self.n):  
 if i % 7 == 0:  
 yield i

for num in iterator(100).divBySeven():  
 print(num)

Question 2:

|  |
| --- |
| Write a program to compute the frequency of the words from the input. The output should output after sorting the key alphanumerically. |
|  |

|  |
| --- |
| Suppose the following input is supplied to the program: |
|  |

|  |
| --- |
| New to Python or choosing between Python 2 and Python 3? Read Python 2 or Python 3. |
|  |

|  |
| --- |
| Then, the output should be: |
|  |

|  |
| --- |
| 2:2 |
|  |

|  |
| --- |
| 3.:1 |
|  |

|  |
| --- |
| 3?:1 |
|  |

|  |
| --- |
| New:1 |
|  |

|  |
| --- |
| Python:5 |
|  |

|  |
| --- |
| Read:1 |
|  |

|  |
| --- |
| and:1 |
|  |

|  |
| --- |
| between:1 |
|  |

|  |
| --- |
| choosing:1 |
|  |

|  |
| --- |
| or:2 |
|  |

to:1

Ans: ##a program to compute the frequency of the words from the input.  
##The output should output after sorting the key alphanumerically  
#input  
##sorting them alphanumerically.  
#inputs  
str=input("enter a string :")  
word=str.split()  
print(word)  
#sortingx  
word\_sorted=word  
for i in range(len(word\_sorted)):  
 for j in range(len(word\_sorted)-1):  
 if word\_sorted[j]>word\_sorted[j+1]:  
 word\_sorted[j],word\_sorted[j+1]=word\_sorted[j+1],word\_sorted[j]  
print(word\_sorted)  
#removing duplicates  
word\_noduplicate=[]  
for i in word\_sorted:  
 if i not in word\_noduplicate:  
 word\_noduplicate.append(i)  
#counting and output  
def count(word\_sorted,i):  
 return word\_sorted.count(i)  
for i in word\_noduplicate:  
 c=count(word\_sorted,i)  
 print(i,":",c)

|  |
| --- |
| Question 3: |
|  |

|  |
| --- |
|  |
|  |

Define a class Person and its two child classes: Male and Female. All classes have a method "getGender" which can print "Male" for Male class and "Female" for Female class.

Ans: #inputs  
class Person(object):  
 def getGender( self ):  
 return "Unknown"  
  
class Male( Person ):  
 def getGender( self ):  
 return "Male"  
  
class Female( Person ):  
 def getGender( self ):  
 return "Female"  
  
aMale = Male()  
aFemale= Female()  
print (aMale.getGender())  
print (aFemale.getGender())

Question 4:

Please write a program to generate all sentences where subject is in ["I", "You"] and verb is in ["Play", "Love"] and the object is in ["Hockey","Football"].

Ans: subjects=["I", "You"]  
verbs=["Play", "Love"]  
objects=["Hockey","Football"]  
  
for sub in subjects:  
 for verb in verbs:  
 for obj in objects:  
 print("{} {} {}".format(sub,verb,obj))

Question 5:

Please write a program to compress and decompress the string "hello world!hello world!hello world!hello world!".

import zlib  
s = 'hello world!hello world!hello world!hello world!'.encode()  
t = zlib.compress(s)  
print(t)  
print(zlib.decompress(t))

Question 6:

Please write a binary search function which searches an item in a sorted list. The function should return the index of element to be searched in the list.

##a program to compute the frequency of the words from the input.  
##The output should output after sorting the key alphanumerically  
##sorting them alphanumerically.  
#inputs  
l=int(input("enter the length of list"))  
lis=[]  
print("enter the elements")  
for i in range(l):  
 x=input()  
 lis.append(x)  
item=input("enter the item to be searched")  
#sorting  
for i in range(l):  
 for j in range(l-1):  
 if lis[j]>lis[j+1]:  
 lis[j],lis[j+1]=lis[j+1],lis[j]  
#binary search  
print("the sorted list :",lis)  
mid=int(l/2)  
if (lis[mid]==item):  
 print("the index of item in the sorted list :", mid )  
elif lis[mid]<item:  
 for i in range(mid,l):  
 if (lis[i]==item):  
 print("the index of item in the sorted list :", i )  
elif lis[mid]>item:  
 for i in range(0,mid):  
 if (lis[i] == item):  
 print("the index of item in the sorted list :", i)  
else:  
 print("the item is not present")