|  |
| --- |
| 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.

# Define a class with a generator which can iterate the numbers, which are divisible by 7,

# between a given range 0 and n.

# Define a class Class1

class Class1:

# Define a generator function, Generator1

def Generator1(n):

for i in range(0,n+1):

num = False

if i % 7 == 0:

num = True

if num == True:

yield i

num = False

# Create an object of a generator, Generator1

g1 = Class1.Generator1(98)

# Print the elements of generator object using list

print(list(g1))

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

# Write a program to compute the frequency of the words from the input.

# The output should output after sorting the key alphanumerically.

from collections import Counter

text = "New to Python or choosing between Python 2 and Python 3? Read Python 2 or Python 3"

# Create a list

l1 = Counter(text.split()).most\_common()

# Sort the list

l1.sort()

# Print the list

print(l1)

|  |
| --- |
| 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.

# 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.

# Define a class Person

class Person:

def getGeneder(self):

print ("this prints gender of a person")

# Define a child class Male

class Male(Person):

def getGeneder(self):

print ("Male")

# Define a child class Female

class Female(Person):

def getGeneder(self):

print ("Female")

# Create an object of each class and use getGender() method

objPerson = Person()

print(objPerson.getGeneder())

objMale = Male()

print(objMale.getGeneder())

objFemale = Female()

print(objFemale.getGeneder())

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"].

# 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"].

# Declare 3 lists variables with the values

l1 = ["I", "You"]

l2 = ["Play", "Love"]

l3 = ["Hockey","Football"]

# Use for loop to generate sentences

for i in range(0, len(l1)):

for j in range(0, len(l2)):

for k in range(0,len(l3)):

print(l1[i]+' '+l2[j]+' '+l3[k])

Question 5:

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

# write a program to compress and decompress the string

# "hello world!hello world!hello world!hello world!".

# import gzip to compress, decompress

import gzip

# declare a string object

s1 = b"hello world!hello world!hello world!hello world!"

# use gzip.compress() to compress the string

s1 = gzip.compress(s1)

print(s1)

# use gzip.decompress() to decompress

s2 = gzip.decompress(s1)

print(s2)

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.

# 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.

# Define a function, function1 which does a binary search

# by passing list and element to be searched as input parameters

def function1(list1, n):

low = 0

high = len(list1) - 1

mid = 0

while low <= high:

# for get integer result

mid = (high + low) // 2

# Check if n is present at mid

if list1[mid] < n:

low = mid + 1

# If n is greater, compare to the right of mid

elif list1[mid] > n:

high = mid - 1

# If n is smaller, compared to the left of mid

else:

return mid

# element was not present in the list, return -1

return -1

# define a sample list

l1 = [2,5,3,9,8,4,6]

n = 9

# call the function, function1 by passing the parameters l1, n

result = function1(l1, n)

if result != -1:

print("Element is present at index", str(result))

else:

print("Element is not present in list1")