|  |
| --- |
| **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:-**

|  |
| --- |
|  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**# Python program to check whether a number is divisible by 7**

**# Function to check whether a number is divisible by 7**

**def isDivisibleBy7(num) :**

**# If number is negative, make it positive**

**if num < 0 :**

**return isDivisibleBy7( -num )**

**# Base cases**

**if( num == 0 or num == 7 ) :**

**return True**

**if( num < 10 ) :**

**return False**

**# Recur for ( num / 10 - 2 \* num % 10 )**

**return isDivisibleBy7( num // 10 - 2 \* ( num - num // 10 \* 10 ) )**

**# Driver program**

**num = 616**

**if(isDivisibleBy7(num)) :**

**print ("Divisible")**

**else :**

**print ("Not Divisible")**

**Output**

Divisible

**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:-**

**# Find frequency of each word in a string in Python**

**# using dictionary.**

**def count(elements):**

**# check if each word has '.' at its last. If so then ignore '.'**

**if elements[-1] == '.':**

**elements = elements[0:len(elements) - 1]**

**# if there exists a key as "elements" then simply**

**# increase its value.**

**if elements in dictionary:**

**dictionary[elements] += 1**

**# if the dictionary does not have the key as "elements"**

**# then create a key "elements" and assign its value to 1.**

**else:**

**dictionary.update({elements: 1})**

**# driver input to check the program.**

**Sentence = "Apple Mango Orange Mango Guava Guava Mango"**

**# Declare a dictionary**

**dictionary = {}**

**# split all the word of the string.**

**lst = Sentence.split()**

**# take each word from lst and pass it to the method count.**

**for elements in lst:**

**count(elements)**

**# print the keys and its corresponding values.**

**for allKeys in dictionary:**

**print ("Frequency of ", allKeys, end = " ")**

**print (":", end = " ")**

**print (dictionary[allKeys], end = " ")**

**print()**

**Output**

Frequency of Apple : 1

Frequency of Mango : 3

Frequency of Orange : 1

Frequency of Guava : 2

|  |
| --- |
| **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:-**

**class Person(object):**

**def \_\_init\_\_(self):**

**self.gender = "unknown"**

**def getGender(self):**

**print(self.gender)**

**class Male(Person):**

**def \_\_init\_\_(self):**

**self.gender = "Male"**

**class Female(Person):**

**def \_\_init\_\_(self):**

**self.gender = "Female"**

**sharon = Female()**

**doug = Male()**

**sharon.getGender()**

**doug.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:-**

**from itertools import product**

**def question\_79():**

**subject = ["I", "You"]**

**verb = ["Play", "Love"]**

**object = ["Hockey", "Football"]**

**prod = [p for p in product(range(2), repeat=3)]**

**for combination in prod:**

**print(f'{subject[combination[0]]} {verb[combination[1]]} {object[combination[2]]}')**

**Question 5:**

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

**ANS:-**

**s = 'hello world!hello world!hello world!hello world!'**

**# In Python 3 zlib.compress() accepts only DataType <bytes>**

**y = bytes(s, 'utf-8')**

**x = zlib.compress(y)**

**print(x)**

**print(zlib.decompress(x))**

**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.**

**ANS:-**

**# Python code to demonstrate working**

**# of binary search in library**

**from bisect import bisect\_left**

**def BinarySearch(a, x):**

**i = bisect\_left(a, x)**

**if i != len(a) and a[i] == x:**

**return i**

**else:**

**return -1**

**a  = [1, 2, 4, 4, 8]**

**x = int(4)**

**res = BinarySearch(a, x)**

**if res == -1:**

**print(x, "is absent")**

**else:**

**print("First occurrence of", x, "is present at", res)**