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

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

**self.init\_num = 0**

**self.final\_num = n + 1**

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

**for i in range(self.init\_num, self.final\_num , 7):**

**yield i**

**ge = gen( 100 )**

**result = ge.gen\_()**

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:**

**def char\_freq(string):**

**lst = string.split(' ')**

**lst1 = []**

**while (lst):**

**tmp = lst[0]**

**flag = 1**

**lst.remove(tmp)**

**for j in lst:**

**if( j == tmp):**

**flag = flag + 1**

**lst.remove(j)**

**lst1.append( (tmp , flag) )**

**lst1.sort()**

**return lst1**

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

**for i in char\_freq( string):**

**print(str(i[0]) + ":" + str(i[1]))**

|  |
| --- |
| 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:**

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

**self.gender = gender**

**def getGender():**

**print(self.gender)**

**class Male(Person):**

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

**super().\_\_init\_\_( "Male")**

**def getGender(self):**

**print( "Male")**

**class Female(Person):**

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

**super().\_\_init\_\_( "Female")**

**def getGender(self):**

**print("Female")**

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:**

**lst1 = ["I", "You"]**

**lst2 = ["Play", "Love"]**

**lst3 = ["Hockey","Football"]**

**for i in range(2):**

**for j in range(2):**

**for k in range(2):**

**print( lst1[i] + " " + lst2[j] + " " + lst3[k])**

Question 5:

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

hello world!hello world!hello world!hello world!

**Ans:**

**def compress(String):**

**tmp = ""**

**flag = 0**

**for i in String:**

**if( i != "!"):**

**tmp = tmp + i**

**else:**

**break**

**for i in String:**

**if( i == "!"):**

**flag = flag + 1**

**return tmp + "!" + str(flag)**

**def decompress(String):**

**n = String[-1]**

**strn = ""**

**String = String[:-1]**

**for i in range(int(n)):**

**strn = strn + String**

**return strn**

**print( compress("hello world!hello world!hello world!hello world!"))**

**print( decompress("hello world!4"))**

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:**

**def search\_sorted( lst , item):**

**length = len(lst)**

**high = length**

**low = 0**

**while( low != high):**

**if( item > lst[ int(( high + low)/2) ] ):**

**low = int(( high + low)/2) + 1**

**elif( item == lst[ int(( high + low)/2) ] ):**

**return int(( high + low)/2)**

**else:**

**high = int(( high + low)/2) -1**