**Python Programming Basic Assignment-14**

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

In [9]:

**def** gen(n):

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

**if** i**%7**==0:

**yield** i

ob**=**gen(21)

In [10]:

**for** i **in** ob:

print(i)

0

7

14

**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**

In [16]:

**def** freq(string):

l**=**string**.**split()

temp**=**''

**for** i **in** l:

**if** i **not** **in** temp:

print(f'{i}:{l**.**count(i)}')

temp**+=**i

In [17]:

freq('New to Python or choosing between Python 2 and Python 3? Read Python 2 or Python 3.')

New:1

to:1

Python:5

or:2

choosing:1

between:1

2:2

and:1

3?:1

Read:1

3.:1

**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.**

In [30]:

**class** Person:

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

**pass**

**def** getGender(self):

**return** 'This is a parent class'

**class** Male(Person):

**def** \_\_init\_\_(self,male\_cm,**\***args):

super(Male,self)**.**\_\_init\_\_(**\***args)

self**.**male\_child**=**male\_cm

**def** getGender(self):

**return** self**.**male\_child

**class** Female(Person):

**def** \_\_init\_\_(self,female\_cm,**\***args):

super(Female,self)**.**\_\_init\_\_(**\***args)

self**.**female\_child**=**female\_cm

**def** getGender(self):

**return** self**.**female\_child

In [31]:

ob\_p**=**Person()

ob\_male**=**Male('Akshay')

ob\_female**=**Female('Shikha')

In [32]:

print(f'For male child class: {ob\_male**.**getGender()}')

For male child class: Akshay

In [33]:

print(f'For female child class: {ob\_female**.**getGender()}')

For female child class: Shikha

**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"].**

In [16]:

sub**=**[]

verb**=**[]

obj**=**[]

s**=**input('Enter the setence: ')**.**split()

**for** i **in** s:

**try**:

**if** (i**==**'I') **or** (i**==**'You'):

sub**.**append(i)

**except** Exception **as** e:

print(e)

**try**:

**if** (i**==**'Play') **or** (i**==**'Love'):

verb**.**append(i)

**except** Exception **as** e:

print(e)

**try**:

**if** (i**==**'Hockey') **or** (i**==**'Football'):

obj**.**append(i)

**except** Exception **as** e:

print(e)

print(sub)

print(verb)

print(obj)

Enter the setence: I You Love Play Hockey Football

['I', 'You']

['Love', 'Play']

['Hockey', 'Football']

**5) Please write a program to compress and decompress the string "hello world!hello world!hello world!hello world!**

In [52]:

**import** gzip

s **=** b"hello world!hello world!hello world!hello world!"

s **=** gzip**.**compress(s)

*# using gzip.decompress(s) method*

t **=** gzip**.**decompress(s)

print(t)

b'hello world!hello world!hello world!hello world!'

**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.**

In [36]:

**def** sort\_func(item):

l1**=**[12,14,13,10,20]

sorted\_list**=**sorted(l1)

print(sorted\_list)

**if** item **in** sorted\_list:

print(f'Item {item} is present and its index number is {sorted\_list**.**index(item)}')

**else**:

print(f'Item {item} not present in the list')

In [37]:

sort\_func(14)

[10, 12, 13, 14, 20]

Item 14 is present and its index number is 3