| 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.

class numgen:

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

self.n = n

def divby7(self):

for x in range(self.n):

if x%7==0:

yield x

gen1 = numgen(100)

for x in gen1.divby7():

print(x)

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.

**inp = input().split()**

**unq = []**

**for x in inp:**

**if x not in unq:**

**unq.append(x)**

**unq.sort()**

**for x in unq:**

**print(x,':',unq.count(x))**

| 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.

class Person:

def get\_gender(self):

return "Unknown"

class Male(Person):

def get\_gender(self):

return "Male"

class Female(Person):

def get\_gender(self):

return "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"].

sub = ["I", "You"]

verb = ["Play", "Love"]

obj = ["Hockey","Football"]

for x in sub:

for y in verb:

for z in obj:

print(x,y,z,'.')

Question 5:

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

import zlib

inp = input()

compressed\_data = zlib.compress(inp.encode())

decompressed\_data = zlib.decompress(compressed\_data).decode()

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 binary\_search(sorted\_list, target):

left = 0

right = len(sorted\_list) - 1

while left <= right:

mid = left + (right - left) // 2

if sorted\_list[mid] == target:

return mid

elif sorted\_list[mid] < target:

left = mid + 1

else:

right = mid - 1

return -1