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

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

self.n = n

def \_\_iter\_\_(self):

self.num = 0

return self

def \_\_next\_\_(self):

while self.num <= self.n:

num = self.num

self.num += 1

if num % 7 == 0:

return num

raise StopIteration

**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**

db7 = DivisibleBySeven(50)

for num in db7:

print(num)

|  |
| --- |
| **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 \_\_init\_\_(self, name):

self.name = name

def getGender(self):

pass

class Male(Person):

def getGender(self):

print("Male")

class Female(Person):

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"].**

subjects = ["I", "You"]

verbs = ["Play", "Love"]

objects = ["Hockey", "Football"]

for subject in subjects:

for verb in verbs:

for obj in objects:

print(f"{subject} {verb} {obj}.")

**Question 5:**

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

import zlib

# Compress the string

string\_to\_compress = b'hello world!hello world!hello world!hello world!'

compressed\_string = zlib.compress(string\_to\_compress)

# Decompress the string

decompressed\_string = zlib.decompress(compressed\_string)

print("Original string:", string\_to\_compress)

print("Compressed string:", compressed\_string)

print("Decompressed string:", decompressed\_string)

**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.**

def binary\_search(arr, x):

low = 0

high = len(arr) - 1

mid = 0

while low <= high:

mid = (high + low) // 2

if arr[mid] < x:

low = mid + 1

elif arr[mid] > x:

high = mid - 1

else:

return mid

return -1