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.

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
```

Question 3:

Define a class Person and its two child classes: Male and Female. All classes have a method &quot;getGender&quot; which can print &quot;Male&quot; for Male class and &quot;Female&quot; for Female
class.

Question 4:
Please write a program to generate all sentences where subject is in [&quot;I&quot;, &quot;You&quot;] and
verb is in [&quot;Play&quot;, &quot;Love&quot;] and the object is in [&quot;Hockey&quot;,&quot;Football&quot;].

Question 5:
Please write a program to compress and decompress the string &quot;hello world!hello world!hello world!hello world!&quot;.

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.

Answer 1:

Here's an implementation of a class with a generator method to iterate over the numbers between 0 and n which are divisible by 7:

python

```
class DivisibleBySeven:
    def __init__(self, n):
        self.n = n

    def numbers(self):
        for i in range(self.n + 1):
            if i % 7 == 0:
                yield i
```

Here's how you can use this class to print all numbers between 0 and 100 which are divisible by 7:

python

```
div_by_7 = DivisibleBySeven(100)
for num in div_by_7.numbers():
    print(num)
```

Answer 2:

Here's a Python program to compute the frequency of words in the input string and print the result in sorted order:

python

```
text = input("Enter a string: ")

word_freq = {}
for word in text.split():
    if word in word_freq:
        word_freq[word] += 1
    else:
        word_freq[word] = 1

sorted_word_freq = sorted(word_freq.items(), key=lambda x: x[0])

for word, freq in sorted_word_freq:
    print(f"{word}:{freq}")
```

Example output for the given input:

makefile

```
2:2
```

```
3?:1
3:1
Between:1
Choosing:1
New:1
Or:2
Python:5
Read:1
To:1
and:1
```

Answer 3:

Here's an implementation of the `Person`, `Male`, and `Female` classes with the `getGender` method as specified:

python

```
class Person:
    def getGender(self):
        return "Unknown"

class Male(Person):
    def getGender(self):
        return "Male"

class Female(Person):
    def getGender(self):
        return "Female"
```

You can use these classes like this:

python

```
p = Person()
print(p.getGender())  # Output: Unknown

m = Male()
print(m.getGender())  # Output: Male

f = Female()
print(f.getGender())  # Output: Female
```

Answer 4:

Here's a Python program to generate all sentences where subject is in ["I", "You"] and verb is in ["Play", "Love"] and the object is in ["Hockey","Football"]:

python
```
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}.")
```

Example output:

css
```
I Play Hockey.
I Play Football.
I Love Hockey.
I Love Football.
You Play Hockey.
You Play Football.
You Love Hockey.
You Love Football.
```

Answer 5:

Here's a Python program to compress and decompress the string "hello world!hello world!hello world!hello world!":

python
```
import zlib

text = "hello world!hello world!hello world!hello world!"

# Compress the text
compressed_text = zlib.compress(text.encode("utf-8"))

# Decompress the text
decompressed_text = zlib.decompress(compressed_text).decode("utf-8")
```

```
print(f"Original text: {text}")
print(f"Compressed text: {compressed_text}")
print(f"Decompressed text: {decompressed_text}")
```

Example output:

vbnet

```
Original text: hello world!hello world!hello world!hello world!
Compressed text: b'x\x9c\xcbH\xcd\xc9\xc9\x07\x00\x06\xd
```