Python Basic Programming 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 [1]:
```
class Generator:
    def __init__(self, n):
        self.n = n
    def divisible_by_7(self):
        for i in range(self.n+1):
            if i % 7 == 0:
                yield i
# example usage
g = Generator(70)
for num in g.divisible_by_7():
    print(num)
```

```
0
7
14
21
28
35
42
49
56
63
70
```

2. Write a program to compute the frequency of the words from the input. The output should output after sorting the key alphanumerically.

In [2]:
```
from collections import defaultdict
def compute_frequency(string):
    # split the string into a list of words
    words = string.split()
    # create a defaultdict to store the frequencies
    frequency = defaultdict(int)
    # iterate through the list of words and increment the count for each word
    for word in words:
        frequency[word] += 1
        # sort the dictionary by key and return the result
    return sorted(frequency.items())
# example usage
print(compute_frequency("this is a test string with some repeating words"))
```

```
[('a', 1), ('is', 1), ('repeating', 1), ('some', 1), ('string', 1), ('test', 1), ('th
is', 1), ('with', 1), ('words', 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 [3]:
```
class Person:
    def getGender(self):
        raise NotImplementedError
class Male(Person):
    def getGender(self):
        return "Male"
class Female(Person):
    def getGender(self):
        return "Female"
# example usage
person1 = Male()
person2 = Female()
print(person1.getGender())
print(person2.getGender())
```

```
Male
Female
```

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 [4]:
```
subjects = ["I", "You"]
verbs = ['Play', "Love"]
objects = ["Hockey","Football"]
for subject in subjects:
    for verb in verbs:
        for obj in objects:
            print(subject + " " + verb + " " + obj)
```

```
I Play Hockey
I Play Football
I Love Hockey
I Love Football
You Play Hockey
You Play Football
You Love Hockey
You Love Football
```

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

In [5]:
```
def compress(string):
    result = ""
    count = 1
    prev = string[0]
    for i in range(1, len(string)):
        if string[i] == prev:
            count += 1
        else:
            result += prev + str(count)
            prev = string[i]
            count = 1
    result += prev + str(count)
    return result
```

```
def decompress(string):
    result = ""
    count = 0
    prev = ""
    for c in string:
        if c.isdigit():
            count = count * 10 + int(c)
        else:
            result += c * count
            count = 0
    return result
compressed = compress("hello world!hello world!hello world!hello world!")
print(compressed)
decompressed = decompress(compressed)
print(decompressed)
```

```
h1e1l2o1 1w1o1r1l1d1!1h1e1l2o1 1w1o1r1l1d1!1h1e1l2o1 1w1o1r1l1d1!1h1e1l2o1 1w1o1r1l1d
1!1
eloo world!heloo world!heloo world!heloo 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 [6]:
```
def binary_search(arr, elem):
    low = 0
    high = len(arr) - 1
    while low <= high:
        mid = (low + high) // 2
        if arr[mid] == elem:
            return mid
        elif arr[mid] < elem:
            low = mid + 1
        else:
            high = mid - 1
    return -1
# Test the function
arr = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
elem = 5
result = binary_search(arr, elem)
if result != -1:
    print(f"Element {elem} found at index {result}")
else:
    print(f"Element {elem} not found in the list")
```

```
Element 5 found at index 4
```

In [ ]:
```

```