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 [5]:

```
n = int(input())
divBy7 = [i for i in range(0, n) if (i % 7 == 0)]
print(divBy7)

def divChecker(n):
    for i in range(n):
        if i % 7 == 0:
            value = True
        else:
            value = False
        print(i, value)

divChecker(n)
```

```
12
[0, 7]
0 True
1 False
2 False
3 False
4 False
5 False
6 False
7 True
8 False
9 False
10 False
11 False
```

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

In [9]:

```
import operator

text_line = input("Type in: ")

freq_dict = {}

for i in text_line.split(' '):
    if i.isalpha():
        if i not in freq_dict:
            freq_dict[i] = 1
        elif i in freq_dict:
            freq_dict[i] = freq_dict[i] + 1
    else:
        pass

sorted_freq_dict = sorted(freq_dict.items(), key = operator.itemgetter(0))
print(sorted_freq_dict)

for i in sorted_freq_dict:
    print(i[0], i[1])
```

```
Type in: Anil is at ineuron at bangalore
[('Anil', 1), ('at', 2), ('bangalore', 1), ('ineuron', 1), ('is', 1)]
Anil 1
at 2
bangalore 1
ineuron 1
is 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 [10]:

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

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

In [11]:

```
def solve(s):
   res = ""
   cnt = 1
   for i in range(1, len(s)):
      if s[i - 1] == s[i]:
         cnt += 1
      else:
         res = res + s[i - 1]
         if cnt > 1:
            res += str(cnt)
         cnt = 1
   res = res + s[-1]
   if cnt > 1:
      res += str(cnt)
   return res

s = "hello world!hello world!hello world!hello world!"
print(solve(s))
```

```
hel2o world!hel2o world!hel2o world!hel2o world!
```

5. 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 [12]:

```
def binary_search(arr, low, high, x):

    # Check base case
    if high >= low:

        mid = (high + low) // 2

        # If element is present at the middle itself
        if arr[mid] == x:
            return mid

        # If element is smaller than mid, then it can only
        # be present in left subarray
        elif arr[mid] > x:
            return binary_search(arr, low, mid - 1, x)

        # Else the element can only be present in right subarray
        else:
            return binary_search(arr, mid + 1, high, x)

    else:
        # Element is not present in the array
        return -1

# Test array
arr = [ 2, 3, 4, 10, 40 ]
x = 10

# Function call
result = binary_search(arr, 0, len(arr)-1, x)

if result != -1:
    print("Element is present at index", str(result))
else:
    print("Element is not present in array")
```

```
Element is present at index 3
```

In [ ]:

```

```