ASSIGNMENT – 2 Dhruv Anurag, 18103310, B9

1)
```
for x in range(1, 4):

    print ("Hello world")
```

2)
```
print ("Hello world")   ; Output: Hello world
```

3)
```
duplicate(list)
l1=[]
for i in list:
    If i not in l1:
       l1.append(i)
    else:

       print(i,end=' ')
```

4)
```
def group(l, size):
        return [l[i:i+size] for i in range(0, len(l), size)]
```

5)
```
def lensort(a):
    n = len(a)
   for i in range(n):
      for j in range(i+1,n):
          if len(a[i]) > len(a[j]):
              temp = a[i]
              a[i] = a[j]
              a[j] = temp
    return a
```

6)
```
def extsort(x):
    i=0
    while(i<len(x)):
        x[i]=x[i].split('.')
       i=i+1
    x.sort(key=lambda x:x[1])
    i=0
    while(i<len(x)):
        x[i]=".".join(x[i])
        i=i+1
    return x
```

7)
```
file = open("sample.txt", "r")

number_of_lines = 0
number_of_words = 0
number_of_characters = 0
for line in file:
  line = line.strip("\n")

  words = line.split()
  number_of_lines += 1
  number_of_characters += len(line)

file.close()
```

8)
```
f1 = open("output1.txt", "w")

with open("file.txt", "r") as myfile:
  data = myfile.read()

data_1 = data[::-1]

f1.write(data_1)

f1.close()
```

9)
```
def revline(x):
                        i=0
                          z=len(open(x).readlines())
                          rev=[None]*z
                          f=open(x)
                          while(i<z):
                              rev[i]=f.readline()
                              rev[i]=rev[i].strip()
                              print rev[i][::-1]
                              i=i+1
```

10)
```
def wrap(filename):
                         k = int(sys.argv[2])

                        f=open(filename).readlines()
                         for i in f:
                          new=i
                          while len(new)>k:
                            print new[:k]
                            new=new[k:]
                          print new
```

11)
```
def num (n) :
          return n * 2

    lst = [2, 44, 5.5, 6, -7]
    # creates a map object
    x = map(num, lst)
    print(x)

    print(list(x))
```

12)
```
 numbers = [1,2,3,4,5]
def func(element):
      return element<5
filter(func,numbers)
```

13)
```
def triplets(n):
      return [ (a,c-a,c) for c in range(2,n) for a in range(1,c//2+1) ]
```

14) def parse_csv(file):

```
f=open(file,'r')
txt=f.readlines()
for l in txt:
  s=re.findall('\w+',l)
  res.append(s)
print res
```

15) def mutate(d):

```
    ret=[d]
    i=0
    l=len(d)
    alp=map(chr,range(97,123))

    while i<l:
        cop=d
        ret.append(cop[:i]+cop[i+1:])
        if i<l-2:
          ret.append(cop[:i]+cop[i+1]+cop[i]+cop[i+2:])
        elif i<l-1:
          ret.append(cop[:i]+cop[i+1]+cop[i])
        for x in alp:
            ret.append(cop[:i]+x+cop[i+1:])
    for x in alp:
            ret.append(d+x)
            ret.append(x+d)
            ret.append(cop[:i]+x+cop[i:])
      i=i+1
    return ret
```

16) def nearly_equal(str1,str2):

```
  count=0
  i=j=0
  while(i<len(str1) and j<len(str2)):
  if(str1[i]!=str2[j]):
     count=count+1
     if(len(str1)>len(str2)):
        i=i+1
     elif(len(str1)==len(str2)):
        pass
     else:
        i=i-1
  if(count>1):
     return False
  i=i+1
  j=j+1
  if(count<2):
        return True
```

17) file = open (f, "r")

```
a=[]
b={}
for i in file:
    for j in range(0,len(i)):
        a.append(i[j])
for i in a:
    if i in b:
        b[i]+=1
    else:
        b[i]=1
print(b)
c=f.split(".")
if c[1]=="txt":
    print("\n\nit is a text file")
elif c[1]=="cpp":
    print("\n\nit is a c++ file")
else:
    print("\n\nit is a c file")
```

18)
```
def anagrams(x):
    from itertools import permutations
    s={}
    while len(x)>0:
        x1=x.pop()
        s[x1]=s.get(x1,[])
        s[x1].append(x1)
        i=0
        while i<len(x):
            z1=x[i]
            perm=[".join(p) for p in permutations(x1)]
            if z1 in perm:
                x.remove(z1)
                s[x1].append(z1)
            else:i=i+1
    return s.values()
```