```
1  from collections import Counter
2  from collections import defaultdict
3  # File Objects
4
5  # r ---> Read Only, w ---> Write Only, a ---> Append, r+ ---> Read and Write
6
7  # Reading file without using Context manager
8  f = open('read.txt', 'r')
9  f_contents = f.read()
10 print(f_contents)
11 f.close()
12
13 # Reading file Using Context Manager (No need to close the file explicitly wh
14 with open('read.txt') as f:
15     f_contents = f.readlines()   # Returns a List
16     for line in f_contents:
17         print(line, end='')
18     print('Number of lines in the file ', len(f_contents))  # Prints total nu
19
20 with open('read.txt') as f:
21     f_contents = f.read()    # Reads the entire contents of the file into a v
22     print(f_contents, end='')
23
24 with open('read.txt') as f:
25     print(f.readline(), end='')  # Reads one line at a time
26     print(f.readline(), end='')
27     print(f.readline(), end='')
```

```
28
29  with open('read.txt') as f:
30      line = f.readline()
31      while line:
32          print(line, end='')
33          line = f.readline()
34
35  with open('read.txt') as f:
36      for line in f:           # Loads only one line at a time and prints the l
37          print(line, end='')
38
39  # Reading 10 characters at a time
40  size_to_read = 10
41  with open('read.txt') as f:
42      f_contents = f.read(size_to_read)
43      print(f_contents, end='')
44      f_contents = f.read(size_to_read)
45      print(f_contents, end='')
46
47
48  # Printing the line's with line numbers
49  with open('sample.txt') as f:
50      for linenumber, line in enumerate(f, start=1):
51          print(linenumber, line, end='')
52
53  # Reading the file in reversed order
54  with open('read.txt') as f:
55      for line in reversed(list(f)):
56          print(line, end='')
57
58  # Finding the length of each line in the text file
59  with open('read.txt') as f:
60      for line in f:
61          print(len(line))
62
63  # Exercises:
64  # 1. Extracting IP addresses from log file
65  with open('access-log.txt') as f:
66      ip = []
67      for line in f:
68          line = line.strip()
69          if line:
70              parts = line.split()
71              ip.append(parts[0])
72
```

```python
# Getting unique ip's from the list
unique_ip = set(ip)

# Using set
with open('access-log.txt') as f:
    ip = set()
    for line in f:
        if line.strip():
            parts = line.split()
            ip.add(parts[0])

# Using List Comprehension
ip = [line.split()[0] for line in open('access-log.txt') if line.strip()]

# Using Set Comprehension
unique_ip = {line.split()[0] for line in open('access-log.txt', 'r') if line.

out_file = open('ip_list.txt', 'w')
for item in unique_ip:
    out_file.write(item)
out_file.close()

# Counting number of occurances of ip adresses in the log file.
# Normal Dict
d = {}
for item in ip:
    if item in d:
        d[item] += 1
    else:
        d[item] = 1

# Using defaultdict
d = defaultdict(int)

for item in ip:
    d[item] += 1

# Using Counter Object
d = Counter(ip)

# Sotring dictionary based on occurances of ip addresses.
sorted_ip = sorted(d.items(), key=lambda item: item[-1])

# Extracting Messages from sample.log
with open('sample.log') as log:
```

```
    for line in log:
        line = line.strip()
        if line:
            parts = line.split()
            print(parts[2])

# Getting Unique Messages (Set comprehension)
unique_messages = {line.split()[2] for line in open("sample.log") if line.str

# Counting Number of INFO, WARN, TRACE Messages.
messages = defaultdict(int)
with open('sample.log') as log:
    for line in log:
        line = line.strip()
        if line:
            parts = line.split()
            messages[parts[2]] += 1

# Using Counter object
message_list = [line.strip().split()[2] for line in open('sample.log') if lin
c = Counter(message_list)
print(c)

# Reading Countries from football.txt
with open('football.txt') as log:
    countries = []
    headers = next(log)     # Skipping Header
    for line in log:
        if line.strip():
            parts = line.split("\t")
            countries.append(parts[1])

# Using List Comprehension
countries = [line.strip().split()[1] for line in open("football.txt") if line
print(len(countries))

# Using set
with open('football.txt') as f:
    unique_countries = set()
    headers = next(log)  # Skipping Header
    for line in f:
        if line.strip("\t"):
            parts = line.split()
            unique_countries.add(parts[1])

```

```python
# Getting Unique Countries using Set Comprehension
set_countries = {line.strip().split()[1] for line in open("football.txt") if

# Using defaultdict
word_count = defaultdict(int)
f = open('sample.txt')
for line in f:
    if line.strip():
        words = line.split()
        for word in words:
            word_count[word] += 1
print(word_count)

# Least and Most occurances of the word
least, *rest, maximum = sorted(d.items(), key=lambda name: name[-1])
print(least)    # Prints the word with least occurance
print(maximum)  # Prints the word with maximum occurance
print(rest)     # Prints all elements between 1st and last element

# Counting total number of words present in a file
words_count = 0
with open('sample.txt') as f:
    for line in f:
        if line.strip():
            words = line.split()
            print(words)
            words_count += len(words)

# Finding the line no of a perticular word in a file.
with open('sample.txt') as f:
    for lineno, line in enumerate(f, start=1):
        if line.strip():
            if "Ruby" in line:
                print(lineno, line)

# list of Dicts from data
def make_dict(line):
    data = line.strip().split('\t')
    return {"brand": data[0], "color": data[1], "size": data[2]}

s = [make_dict(line) for line in open('data.txt')]

# Writing to file
with open('write.txt', 'a') as f:
    f.write('Hello world')
```

```
208     f.write('\n')
209 
210 # # Preventing files from being overwritten
211 # with open('sample.txt', 'x') as f:
212 #     f.write('hello world')
213 #     f.write('hello world') # Throws an exception. Cannot overwrite a file
214 
215 # Opening two files in parallel
216 # Comparing the two files line by line
217 with open('file1.txt') as f1, open('file2.txt') as f2:
218     for f1_line, f2_line in zip(f1, f2):
219         print(f1_line.strip(), f2_line.strip())
220 
221 # Copying Contents of one file to another file
222 with open('sample.txt') as rf, open('sample2.txt', 'w') as wf:
223     for line in rf:     # Iterate through each line in sample.txt
224         wf.write(line)  # Write each line to sample2.txt
```