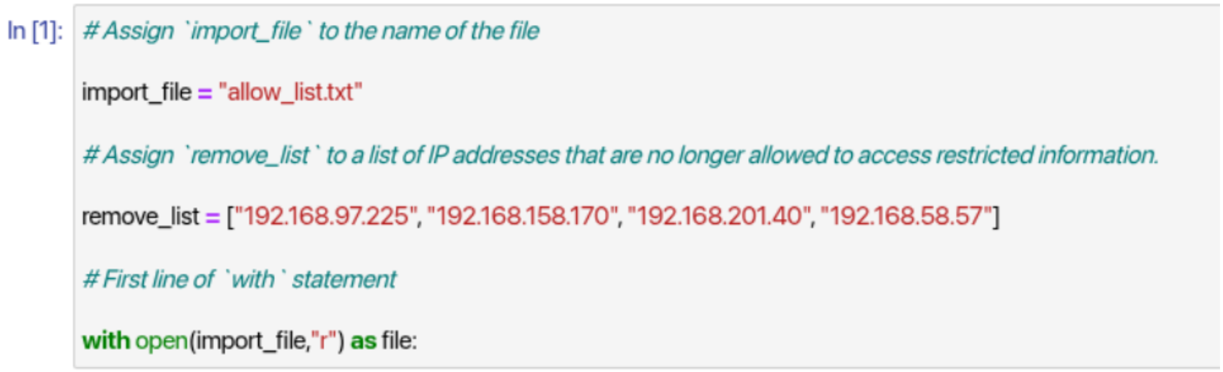
Algorithm for file updates in Python

Project description

At my organisation access to restricted information is controlled with allow lists of ip addresses In this project we will use Python code to check whether the allow list contains any IP addresses identified on the remove list. If so, it should remove those IP addresses from the file containing the allow list.

Open the file that contains the allow list

First we will start by opening the file that contains the allowed ip addresses we will assign a variable import = “allow_list.txt” and we also assign ip addresses that are to be removed as remove_list

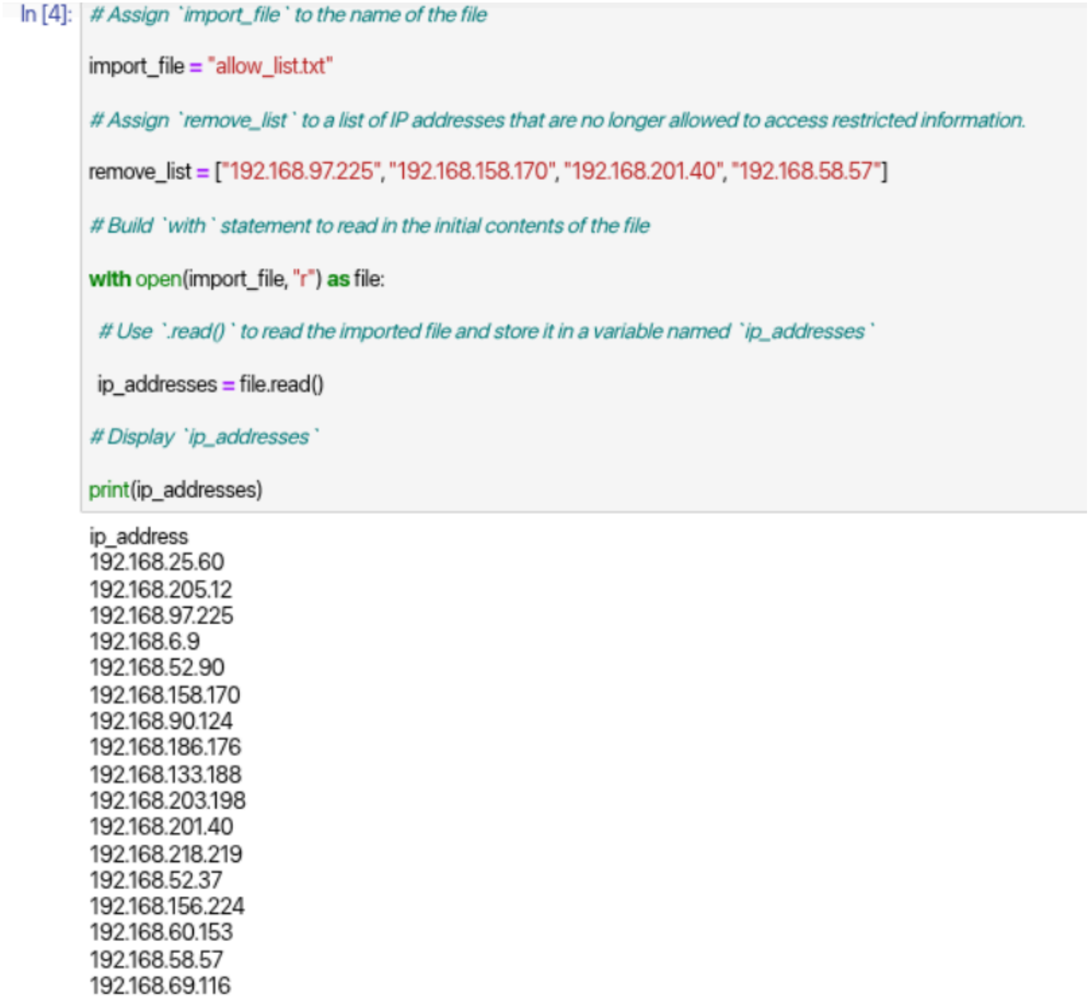


```
In [1]: # Assign `import_file` to the name of the file

        import_file = "allow_list.txt"

        # Assign `remove_list` to a list of IP addresses that are no longer allowed to access restricted information.

        remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]

        # First line of `with` statement

        with open(import_file,"r") as file:
```

Read the file contents

Now first we will see the contents of the file by using the .read() method, we save the output of the read method in a variable ip_addresses .

After that when we print the ip_addresses variable we can see there are several ip addresses in the allowed list that are to be removed to make sure there are no unauthorized users that are able to access sensitive information.



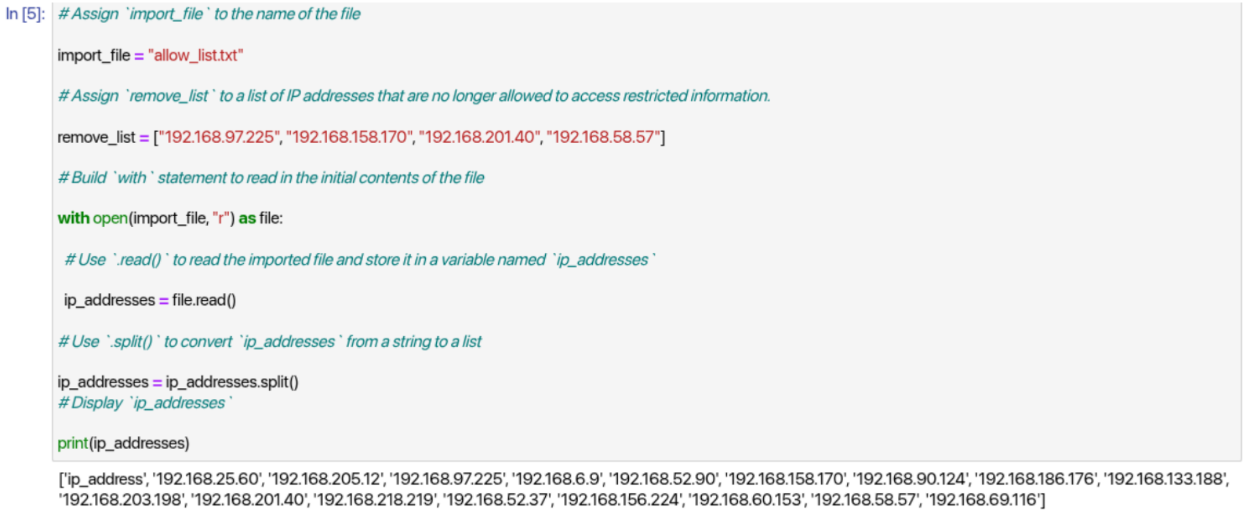
In [4]:

```
# Assign `import_file` to the name of the file

import_file = "allow_list.txt"

# Assign `remove_list` to a list of IP addresses that are no longer allowed to access restricted information.

remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]

# Build `with` statement to read in the initial contents of the file

with open(import_file, "r") as file:

 # Use `.read()` to read the imported file and store it in a variable named `ip_addresses`

 ip_addresses = file.read()

# Display `ip_addresses`

print(ip_addresses)
```

```
ip_address
192.168.25.60
192.168.205.12
192.168.97.225
192.168.6.9
192.168.52.90
192.168.158.170
192.168.90.124
192.168.186.176
192.168.133.188
192.168.203.198
192.168.201.40
192.168.218.219
192.168.52.37
192.168.156.224
192.168.60.153
192.168.58.57
192.168.69.116
```

Convert the string into a list

Now we will convert the ip address data currently stored as string in the ip_addresses variable in list by the use of split function so we can access individual elements.

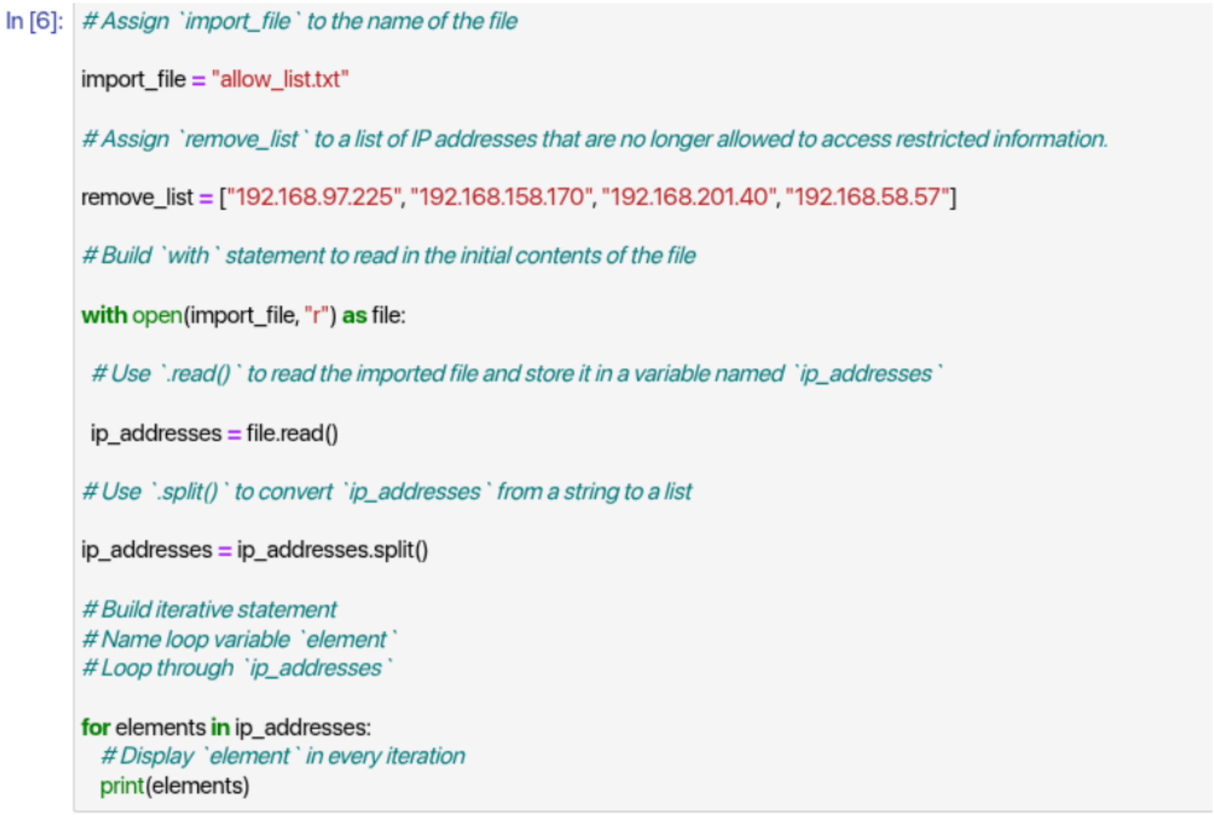


```
In [5]: # Assign `import_file` to the name of the file

        import_file = "allow_list.txt"

        # Assign `remove_list` to a list of IP addresses that are no longer allowed to access restricted information.

        remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]

        # Build `with` statement to read in the initial contents of the file

        with open(import_file, "r") as file:

         # Use `.read()` to read the imported file and store it in a variable named `ip_addresses`

         ip_addresses = file.read()

        # Use `.split()` to convert `ip_addresses` from a string to a list

        ip_addresses = ip_addresses.split()
        # Display `ip_addresses`

        print(ip_addresses)
```

```
['ip_address', '192.168.25.60', '192.168.205.12', '192.168.97.225', '192.168.6.9', '192.168.52.90', '192.168.158.170', '192.168.90.124', '192.168.186.176', '192.168.133.188',
'192.168.203.198', '192.168.201.40', '192.168.218.219', '192.168.52.37', '192.168.156.224', '192.168.60.153', '192.168.58.57', '192.168.69.116']
```

Iterate through the ip_addresses list

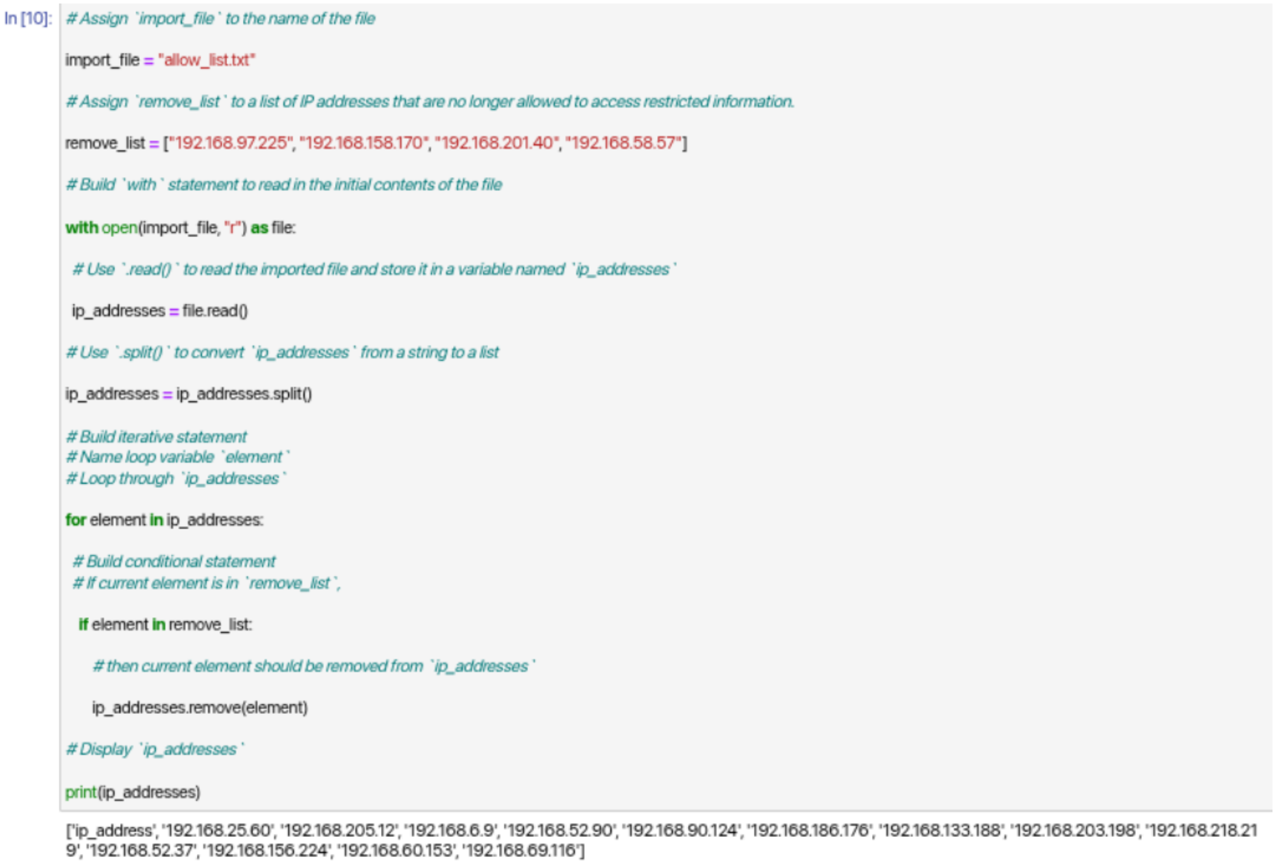
Now we will iterate through every element that are in the list we previously made



```
In [6]: # Assign `import_file` to the name of the file

        import_file = "allow_list.txt"

        # Assign `remove_list` to a list of IP addresses that are no longer allowed to access restricted information.

        remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]

        # Build `with` statement to read in the initial contents of the file

        with open(import_file, "r") as file:

         # Use `.read()` to read the imported file and store it in a variable named `ip_addresses`

         ip_addresses = file.read()

        # Use `.split()` to convert `ip_addresses` from a string to a list

        ip_addresses = ip_addresses.split()

        # Build iterative statement
        # Name loop variable `element`
        # Loop through `ip_addresses`

        for elements in ip_addresses:
          # Display `element` in every iteration
          print(elements)
```

Remove IP addresses that are on the remove list

Now we will remove the ip addresses that are on the remove_list from the list

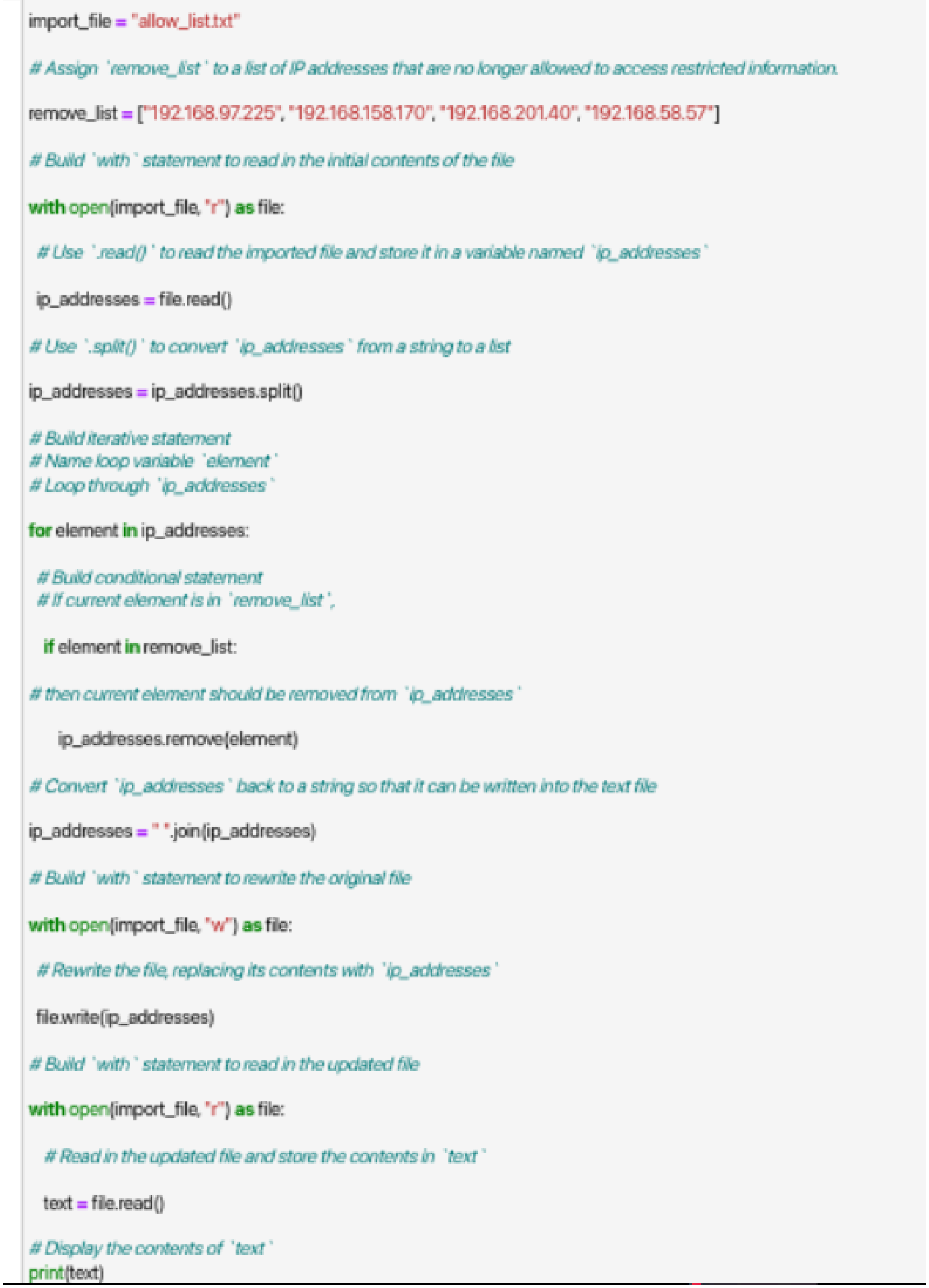


```
In [10]: # Assign `import_file` to the name of the file

import_file = "allow_list.txt"

# Assign `remove_list` to a list of IP addresses that are no longer allowed to access restricted information.

remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]

# Build `with` statement to read in the initial contents of the file

with open(import_file, "r") as file:

 # Use `.read()` to read the imported file and store it in a variable named `ip_addresses`

 ip_addresses = file.read()

# Use `.split()` to convert `ip_addresses` from a string to a list

ip_addresses = ip_addresses.split()

# Build iterative statement
# Name loop variable `element`
# Loop through `ip_addresses`

for element in ip_addresses:

 # Build conditional statement
 # If current element is in `remove_list`,

  if element in remove_list:

    # then current element should be removed from `ip_addresses`

    ip_addresses.remove(element)

# Display `ip_addresses`

print(ip_addresses)
```

```
['ip_address', '192.168.25.60', '192.168.205.12', '192.168.6.9', '192.168.52.90', '192.168.90.124', '192.168.186.176', '192.168.133.188', '192.168.203.198', '192.168.218.21
9', '192.168.52.37', '192.168.156.224', '192.168.60.153', '192.168.69.116']
```

Update the file with the revised list of IP addresses

Now that we have successfully removed the ip addresses from the ip_addresses list, we will write the updated data to the file from which the data was originally from, with the help of open() to open the file in “w” / write mode.
In the whole process you might notice we have used the with keyword that is because it auto manages the external resources and it auto closes the file that we have open so that we don’t have to close it again and again.



```
import_file = "allow_list.txt"

# Assign `remove_list` to a list of IP addresses that are no longer allowed to access restricted information.

remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]

# Build `with` statement to read in the initial contents of the file

with open(import_file, "r") as file:

 # Use `.read()` to read the imported file and store it in a variable named `ip_addresses`

 ip_addresses = file.read()

# Use `.split()` to convert `ip_addresses` from a string to a list

ip_addresses = ip_addresses.split()

# Build iterative statement
# Name loop variable `element`
# Loop through `ip_addresses`

for element in ip_addresses:

 # Build conditional statement
 # If current element is in `remove_list`,

  if element in remove_list:

# then current element should be removed from `ip_addresses`

    ip_addresses.remove(element)

# Convert `ip_addresses` back to a string so that it can be written into the text file

ip_addresses = " ".join(ip_addresses)

# Build `with` statement to rewrite the original file

with open(import_file, "w") as file:

 # Rewrite the file, replacing its contents with `ip_addresses`

 file.write(ip_addresses)

# Build `with` statement to read in the updated file

with open(import_file, "r") as file:

  # Read in the updated file and store the contents in `text`

  text = file.read()

# Display the contents of `text`
print(text)
```

Summary

In summary, in this project we wrote an algorithm to access data from a file then we converted the data in list and then we iterated through individual elements (ip addresses) to match for any ip address that is not allowed and then remove them and write that updated data to the file.
In this project we explored many concepts of python like loops and conditional statements, we also explored how files are parsed to access and work on their data with the help of different modes like “r” and “w” and we also used method like split to changes the form of data to make the data suitable for our purpose.