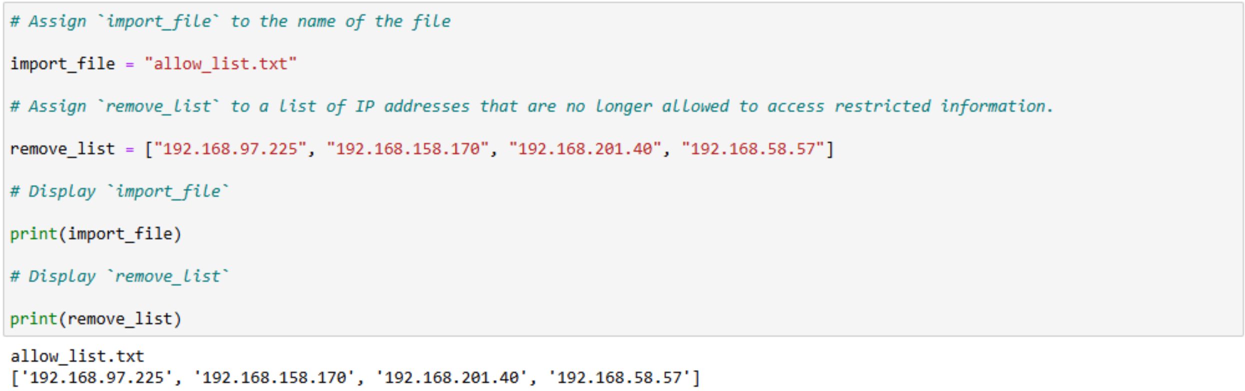
Algorithm for file updates in Python

Project description

In this project, we worked on updating a list of IP addresses stored in a text file. The goal was to remove specific IP addresses from the list that are no longer allowed to access restricted information. We accomplished this by reading the file contents, converting the data from a string to a list, iterating through the list to remove the specified IP addresses, and then writing the updated list back to the file. This process ensures that the file reflects the current allowed IP addresses accurately.

Open the file that contains the allow list



```
# 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"]

# Display `import_file`

print(import_file)

# Display `remove_list`

print(remove_list)
```

```
allow_list.txt
['192.168.97.225', '192.168.158.170', '192.168.201.40', '192.168.58.57']
```

open() function: This function is used to open a file. It requires two main arguments: the file name and the mode in which to open the file. For reading, you use "r" as the mode.

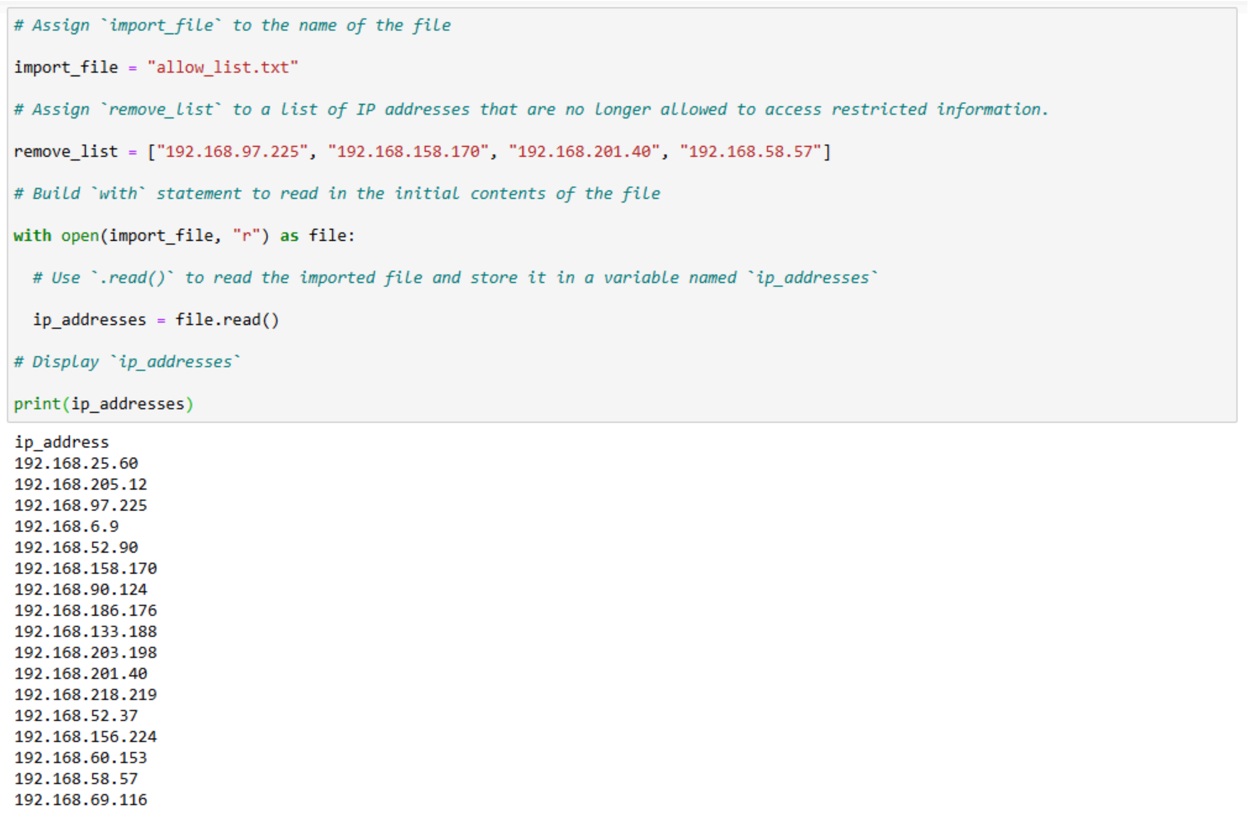
with statement: This is used to ensure that the file is properly closed after its suite finishes, even if an exception is raised. It simplifies exception handling and resource management.

Variable assignment: You assign the file name to a variable (import_file) and the list of IP addresses to another variable (remove_list).

Reading the file: Using the .read() method to read the contents of the file and store it in a variable (ip_addresses).

Printing the contents: Using the print() function to display the contents of the variable.

Read the file contents



```
# 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
```

open() function:

This function is used to open a file. It requires two main arguments: the file name and the mode in which to open the file. For reading, you use "r" as the mode.

with statement:

This is used to ensure that the file is properly closed after its suite finishes, even if an exception is raised. It simplifies exception handling and resource management.

Variable assignment:

You assign the file name to a variable (import_file) and the list of IP addresses to another variable (remove_list).

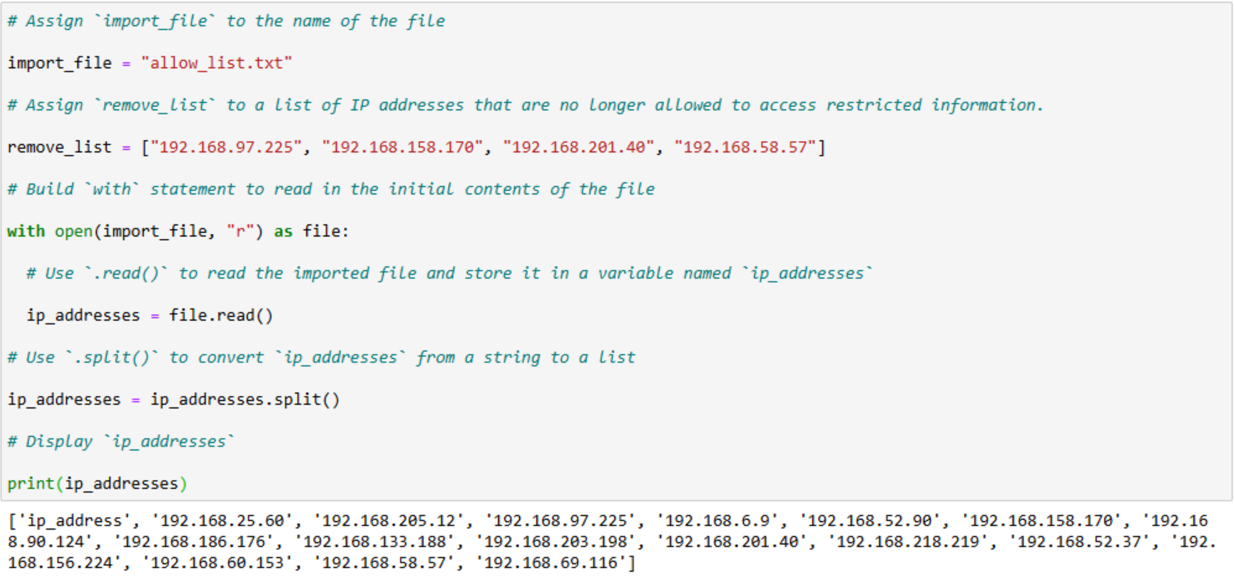
Reading the file:

Using the .read() method to read the contents of the file and store it in a variable (ip_addresses).

Printing the contents:

Using the print() function to display the contents of the variable.

Convert the string into a list



```
# 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', '192.16
8.90.124', '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']
```

Reading the File:

with open(import_file, "r") as file:: This opens the file named import_file in read mode ("r") and assigns it to the variable file.
ip_addresses = file.read(): Reads the entire content of the file and stores it as a string in the variable ip_addresses.

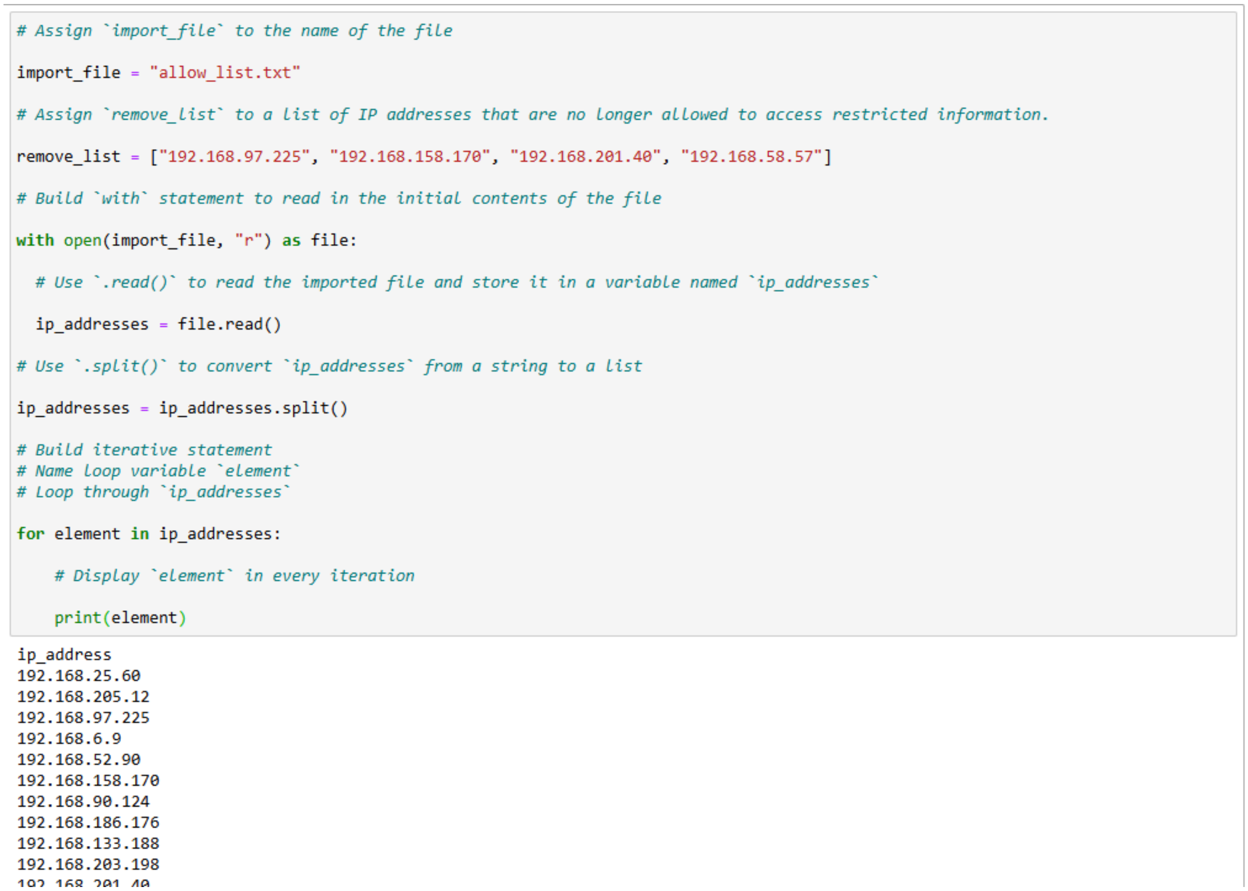
Converting the String to a List:

ip_addresses.split(): The .split() method splits the string into a list of substrings based on whitespace by default. Each IP address in the string, separated by spaces or newlines, becomes an element in the list.
ip_addresses = ip_addresses.split(): Reassigns the ip_addresses variable to the resulting list of IP addresses.

Displaying the List:

print(ip_addresses): Prints the list of IP addresses to verify that the conversion from string to list was successful.

Iterate through the remove list



```
# 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:

    # Display `element` in every iteration

    print(element)
```

```
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
```

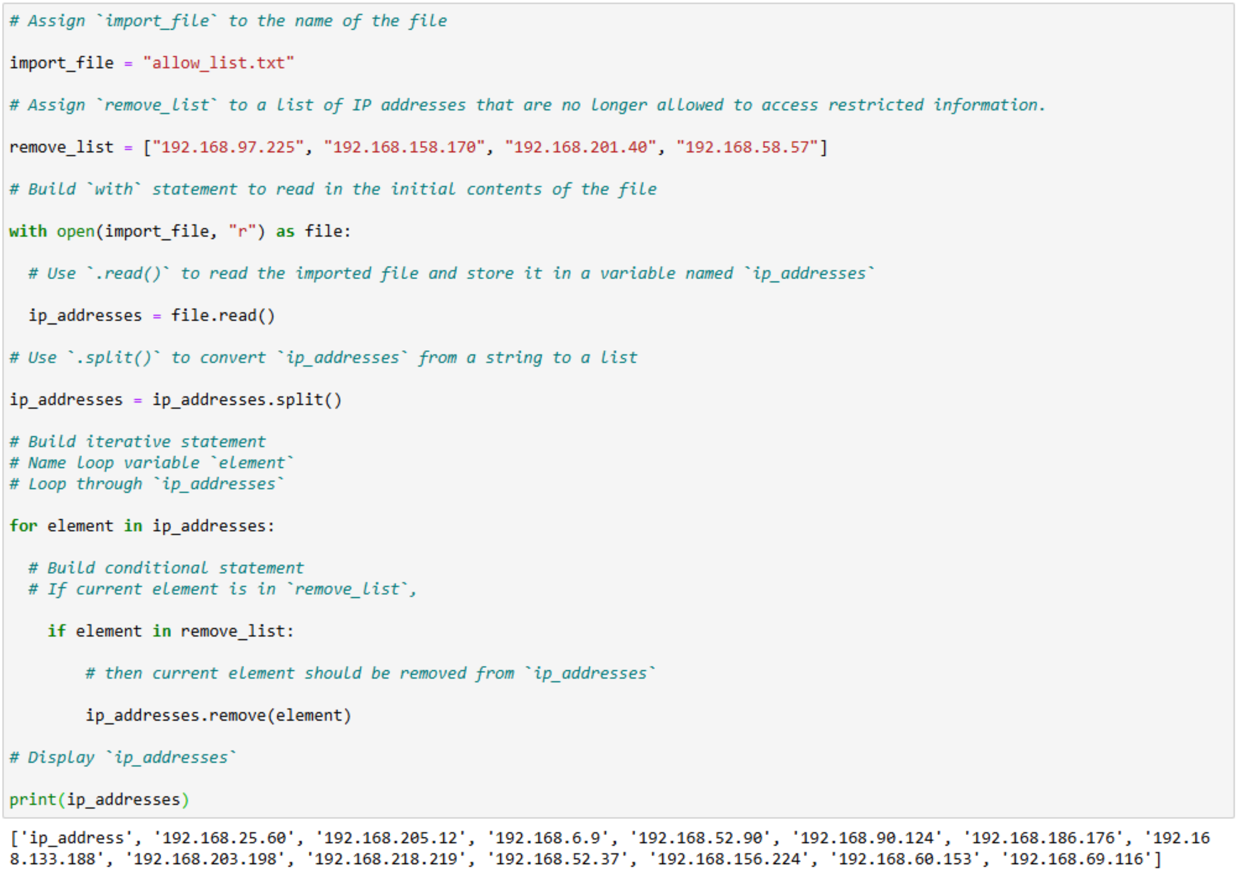
For Loop:

for element in ip_addresses:: This line initializes a for loop. The loop variable element will take on each value in the ip_addresses list, one at a time, in each iteration of the loop.

Print Statement:

print(element): This line prints the current value of element in each iteration of the loop. This allows you to see each IP address in the ip_addresses list.

Remove IP addresses that are on the remove list



```
# 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', '192.16
8.133.188', '192.168.203.198', '192.168.218.219', '192.168.52.37', '192.168.156.224', '192.168.60.153', '192.168.69.116']
```

For Loop:

for element in ip_addresses:: This initializes a for loop. The loop variable element will take on each value in the ip_addresses list, one at a time, in each iteration of the loop.

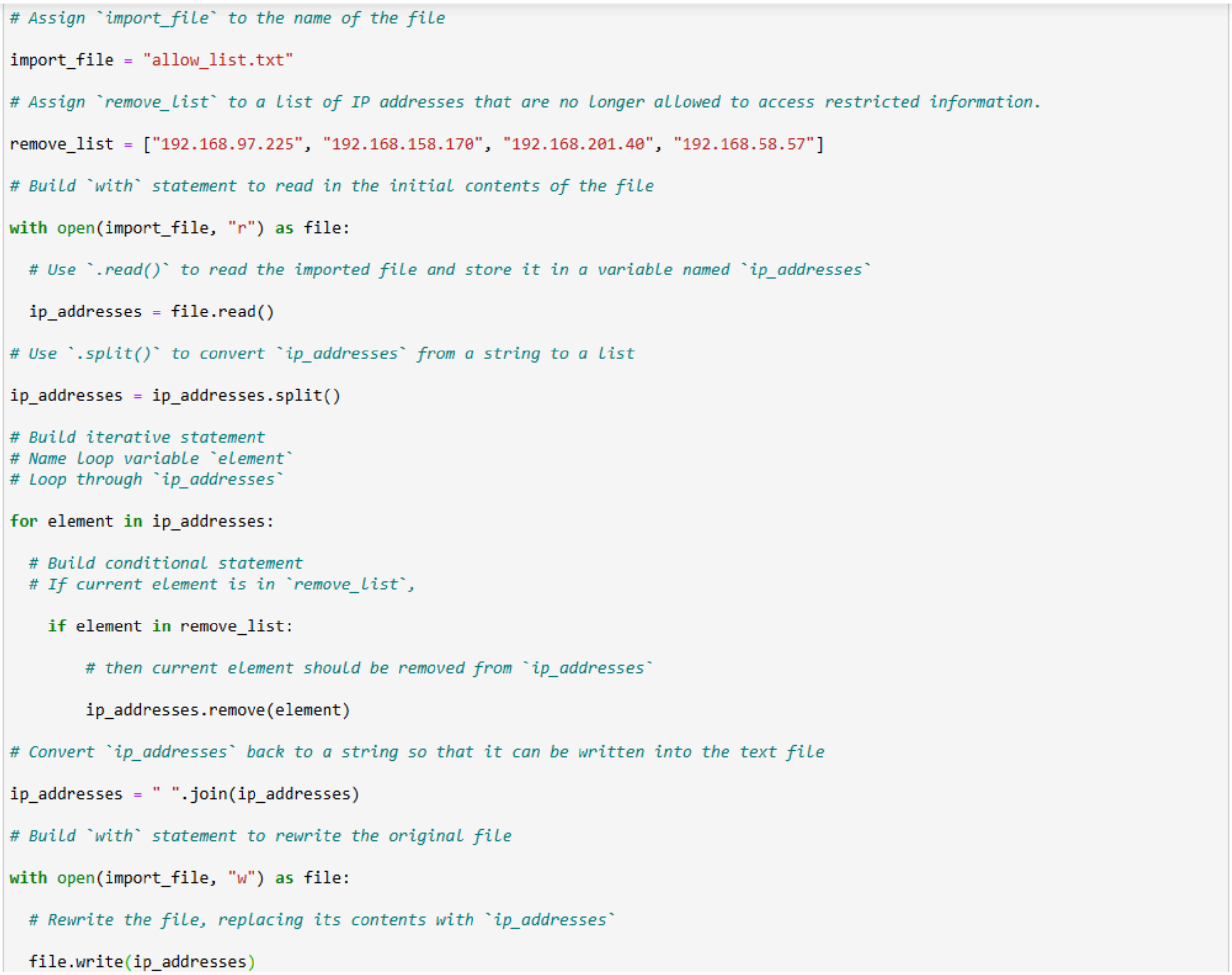
If Statement:

if element in remove_list:: This conditional statement checks if the current element (IP address) is present in the remove_list.

List Method remove():

ip_addresses.remove(element): This method removes the first occurrence of the specified value (element) from the ip_addresses list.

Update the file with the revised list of IP addresses



```python
# 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)

# 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)
```

Convert List to String:

The .join() method is used to convert the list of IP addresses back into a single string, with each IP address separated by a space.
Writing to the File:

The with open(import_file, "w") as file:
statement opens the file in write mode, and file.write(ip_addresses) writes the updated string of IP addresses to the file, replacing its previous contents.

This code will update the allow_list.txt file with the revised list of IP addresses, ensuring that the IP addresses in remove_list are no longer included.

Summary

The algorithm begins by reading the contents of the file and storing it as a string. This string is then converted into a list of IP addresses using the .split() method. We use a for loop combined with an if statement to iterate through the list and remove any IP addresses that are found in the remove_list. After updating the list, we convert it back into a string using the .join() method. Finally, the updated string is written back to the file, replacing its previous contents. This ensures that the file is updated with the revised list of allowed IP addresses.